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

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

+//ISBN 978-1-894049::CSL::Courses::PFUX//DOCUMENT Practical Formatting Using XSL-FO 2008-01-27 17:30UTC//EN
Seventh Edition - 2008-01-27
ISBN 978-1-894049-19-1
http://www.CraneSoftwrights.com
Copyright © Crane Softwrights Ltd.

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.

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

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

 Copyright © Crane Softwrights Ltd.

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

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:

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

 Copyright © Crane Softwrights Ltd.

Paginating structured information (cont.)

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations

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

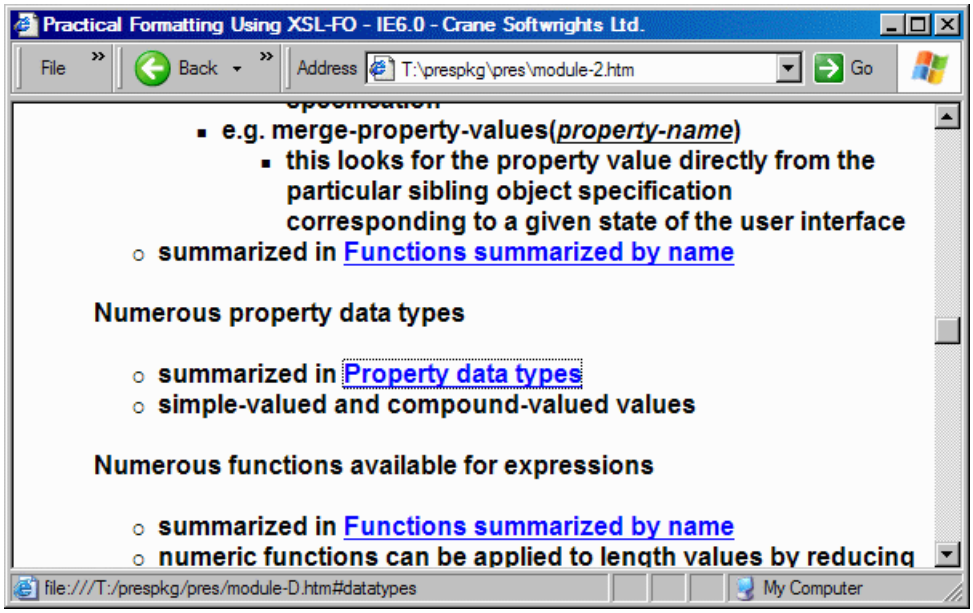
- perhaps the HTML presentation is authored separately from the printed presentation
    - since the navigation tools are different, each authoring tool gives the author a different user interface and authoring options
- the lack of a common source of content introduces maintenance challenges
    - problems 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

Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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



Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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

Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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

```
<course>
    <title>Practical Formatting Using XSL-FO</title>
...
<module id="basic">
  <title>Basic concepts of XSL-FO</title>
  <lesson id="vocab">
    <title>Formatting object XML vocabulary</title>
    <frame id="propexp">
      <title>Property value expressions</title>
...
  <point>summarized in <ref idref="funcname"/></point>
</points>
<para>Numerous property data types</para>
<points>
  <point>summarized in <ref idref="datatypes"/></point>
  <point>simple-valued and compound-valued values</point>
</points>
<para>Numerous functions available for expressions</para>
<points>
  <point>summarized in <ref idref="funcname"/></point>
...
    </frame>
  </lesson>
</module>
...
  <frame id="funcname">
    <title>Functions summarized by name</title>
...
  </frame>
...
  <frame id="datatypes">
    <title>Property data types</title>
...
  </frame>
...
</course>
```


Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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

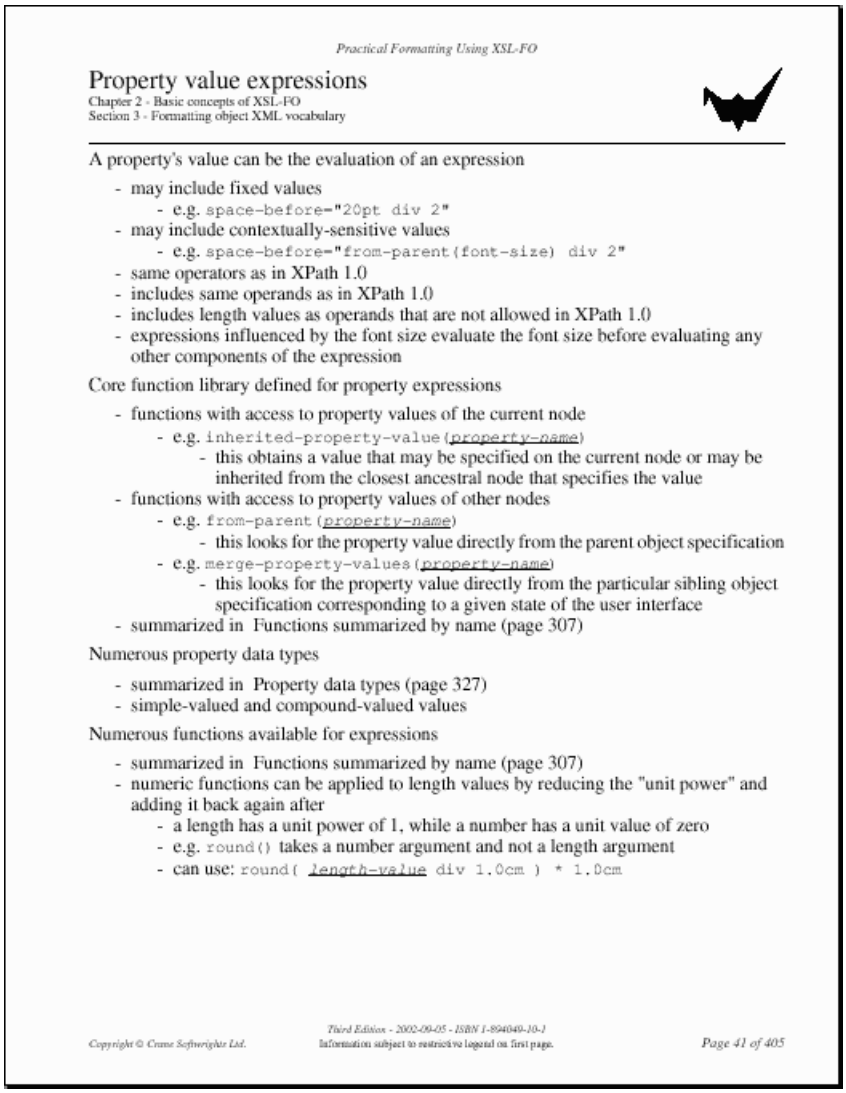
- rearranges instances of XML information into instances of other vocabularies
- e.g. an XSLT tool reads our XML instances and produces an instance of the HTML vocabulary
    - the stylesheet dictates how the HTML is created with the hyperlink reference for traversal and the clickable content for the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

Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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.

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Contrasting browsed vs. paginated presentations (cont.)

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

Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

- `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

 Copyright © Crane Softwrights Ltd.

Extensible Markup Language (XML) (cont.)

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

Copyright © Crane Softwrights Ltd.

XML information links

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

 Copyright © Crane Softwrights Ltd.

- 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

Copyright © Crane Softwrights Ltd.

- `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

 Copyright © Crane Softwrights Ltd.

Styling structured information

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

Copyright © Crane Softwrights Ltd.

- 1.0 `http://www.w3.org/TR/xslt`
- 2.0 `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
- 2.0 XSLT 2.0 has many more programming features and function calls than XSLT 1.0

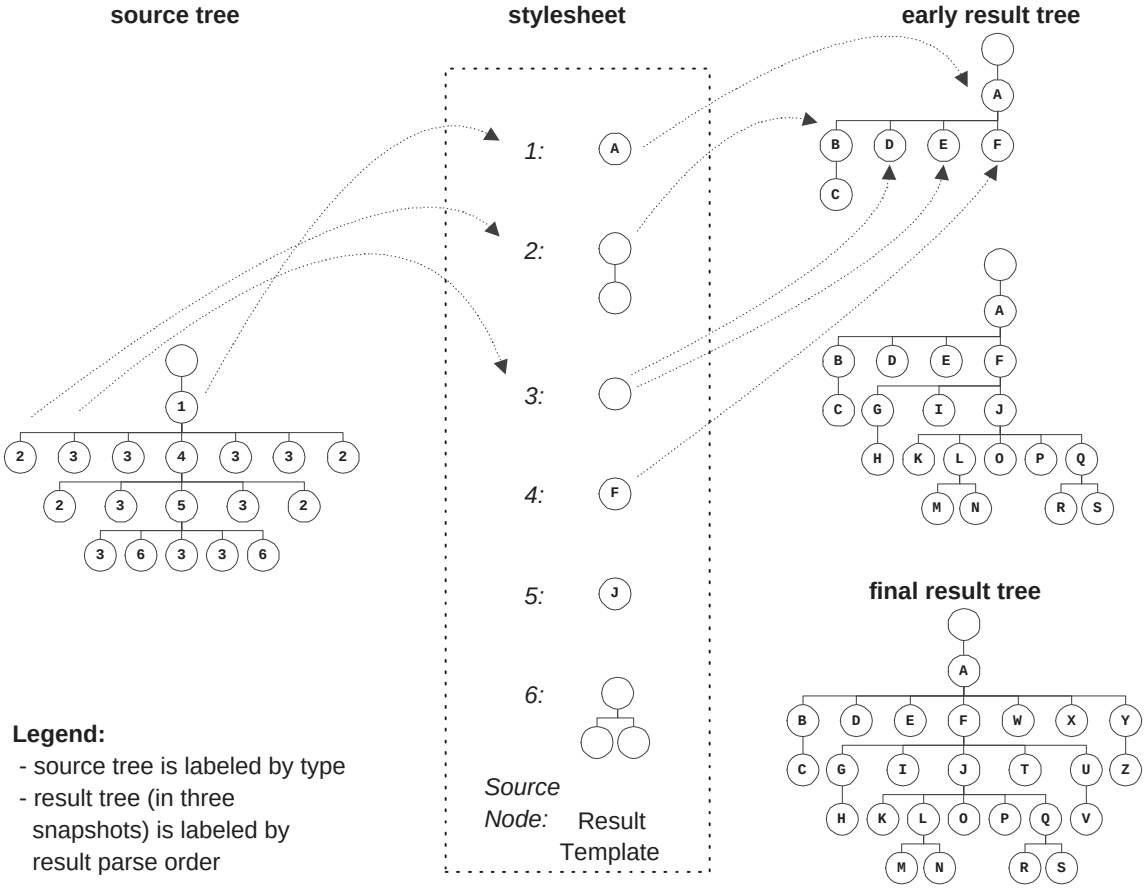
Not intended for syntactic general purpose XML transformations

- 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
- 2.0 XSLT 2.0 has many more syntax serialization features than XSLT 1.0

 Copyright © Crane Softwrights Ltd.

Extensible Stylesheet Language Transformations (XSLT) (cont.)

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

Copyright © Crane Softwrights Ltd.

- `http://www.w3.org/TR/2001/REC-xsl-20011015/`
- `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

 Copyright © Crane Softwrights Ltd.

Styling semantics and vocabularies

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

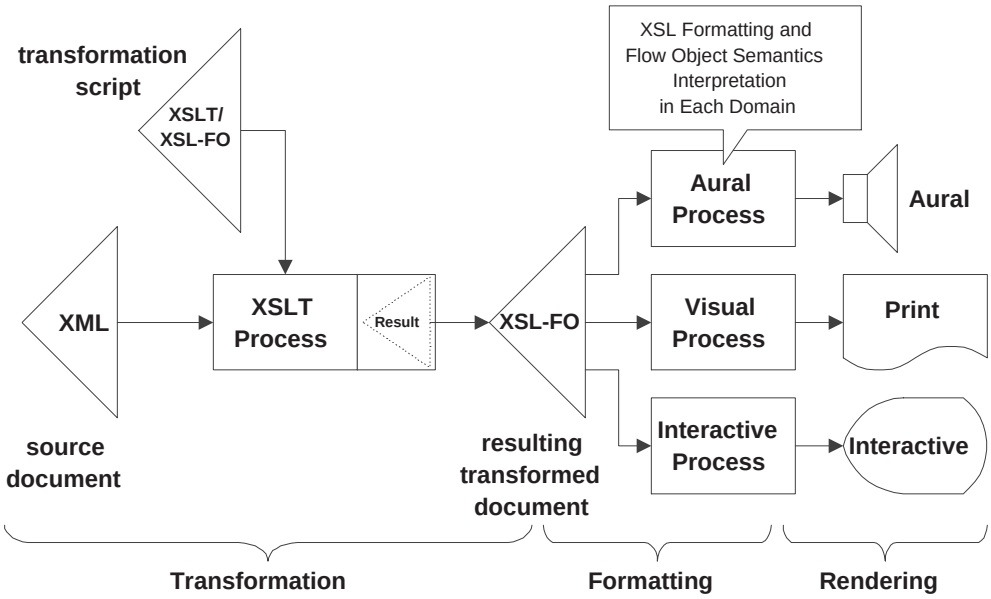
- there are no standardized XML 1.0 DTD representations of the grammar of the vocabularies
    - DTD semantics and syntax unable to fully express all of the grammatical constraints
- XSLT 1.0 and XSL 1.0 Recommendations describe the document type definitions
- processors do all aspects of validation 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

Copyright © Crane Softwrights Ltd.

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:



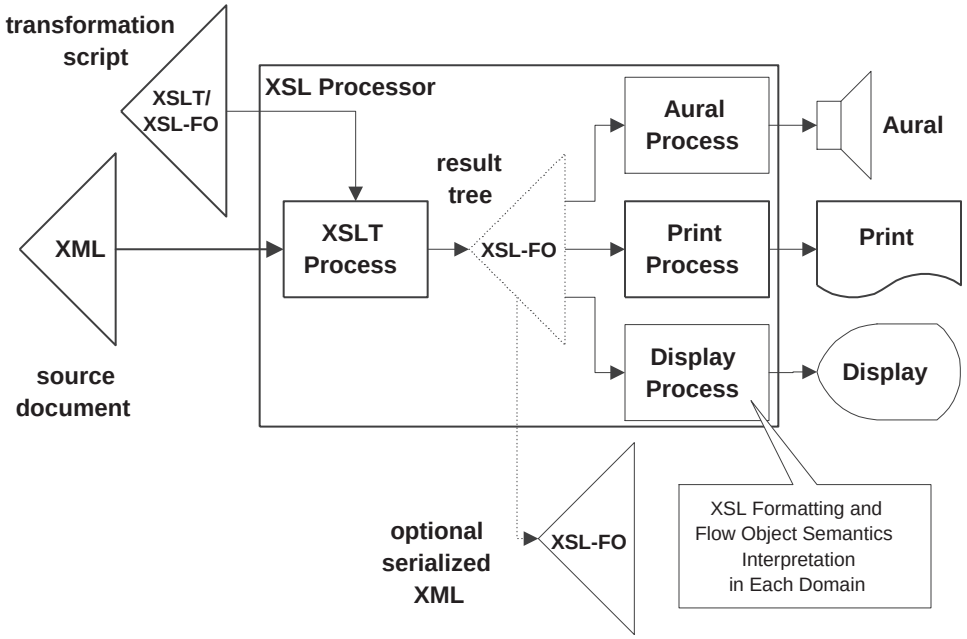
Note the same three distinct phases as when XSLT and XSL-FO processors are combined in a single application:

- 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

Copyright © Crane Softwrights Ltd.

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:

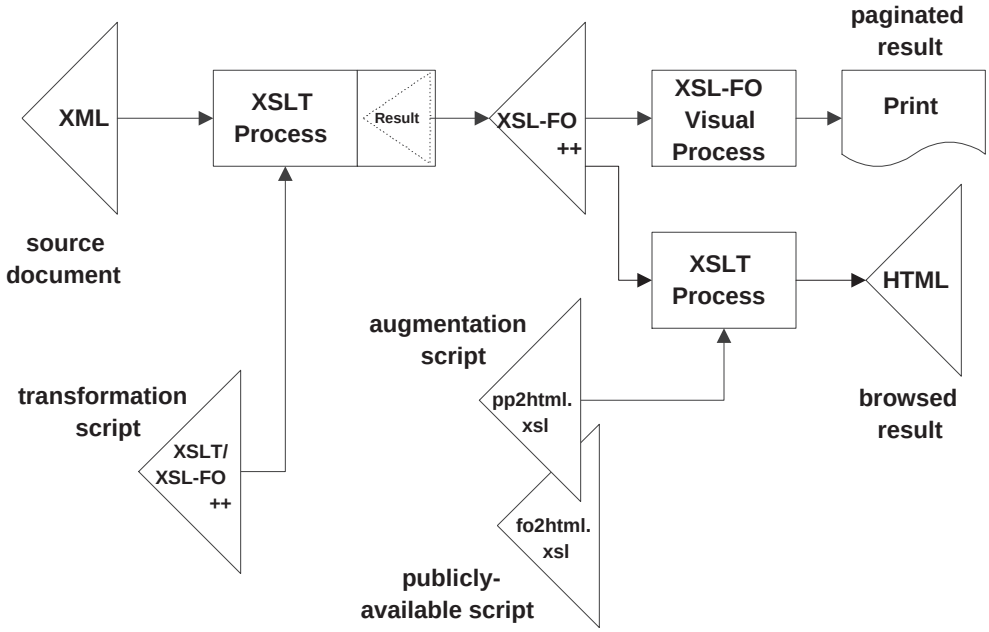
- the stylesheet contains only the XSLT transformation vocabulary, the XSL formatting vocabulary, and extension transformation or foreign object vocabularies
- the source XML contains the user's vocabularies
- the result of transformation contains exclusively the XSL formatting vocabulary and any extension 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

Copyright © Crane Softwrights Ltd.

Using XSL-FO as an intermediate form

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:

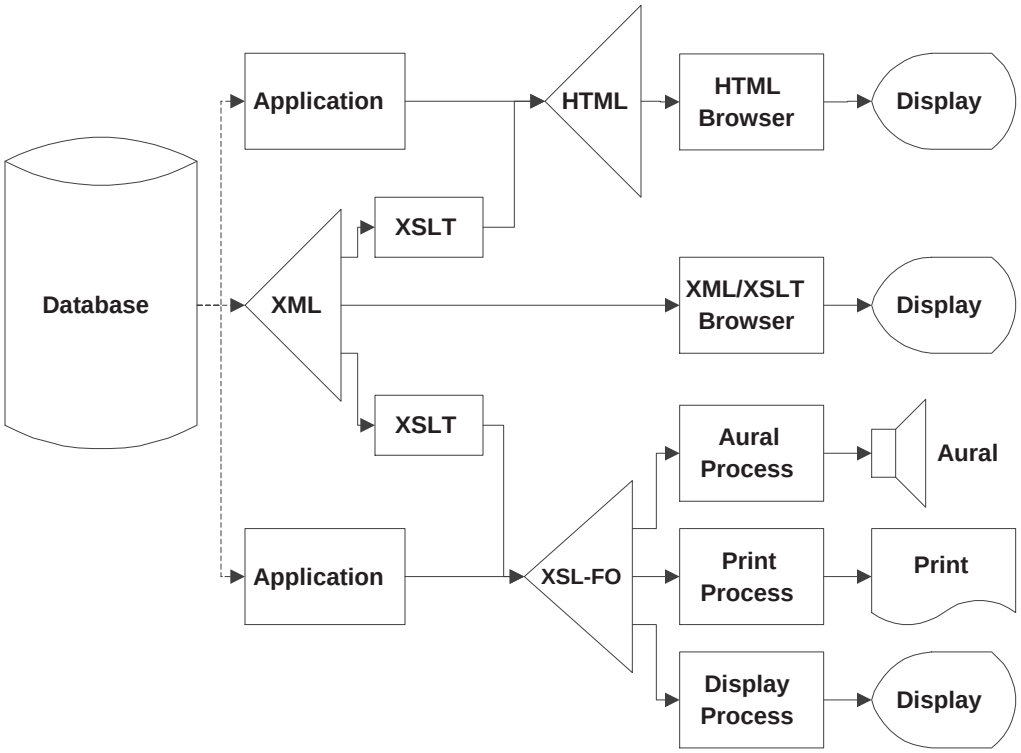
- find the latest version of the `fo2html.xsl` stylesheet using a Google search with the following criteria:
    - `fo2html site:renderx.com`

 Copyright © Crane Softwrights Ltd.

Generating XSL-FO instances

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

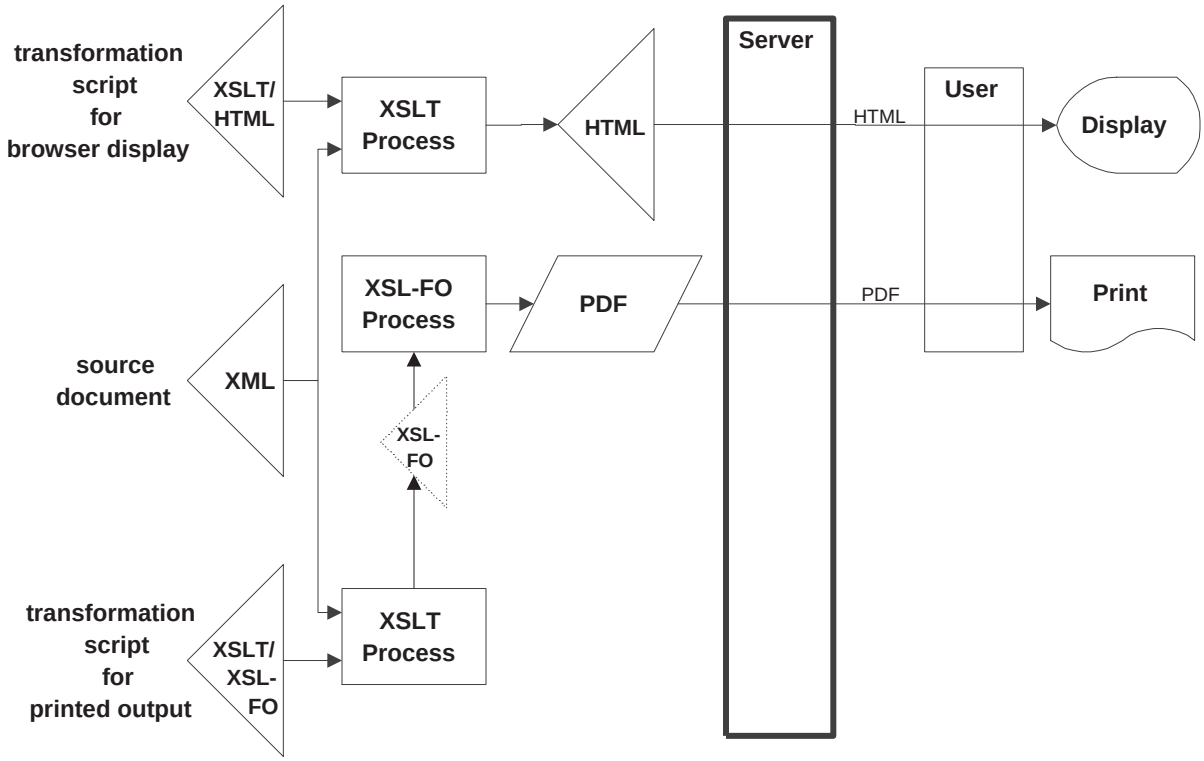
Copyright © Crane Softwrights Ltd.

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

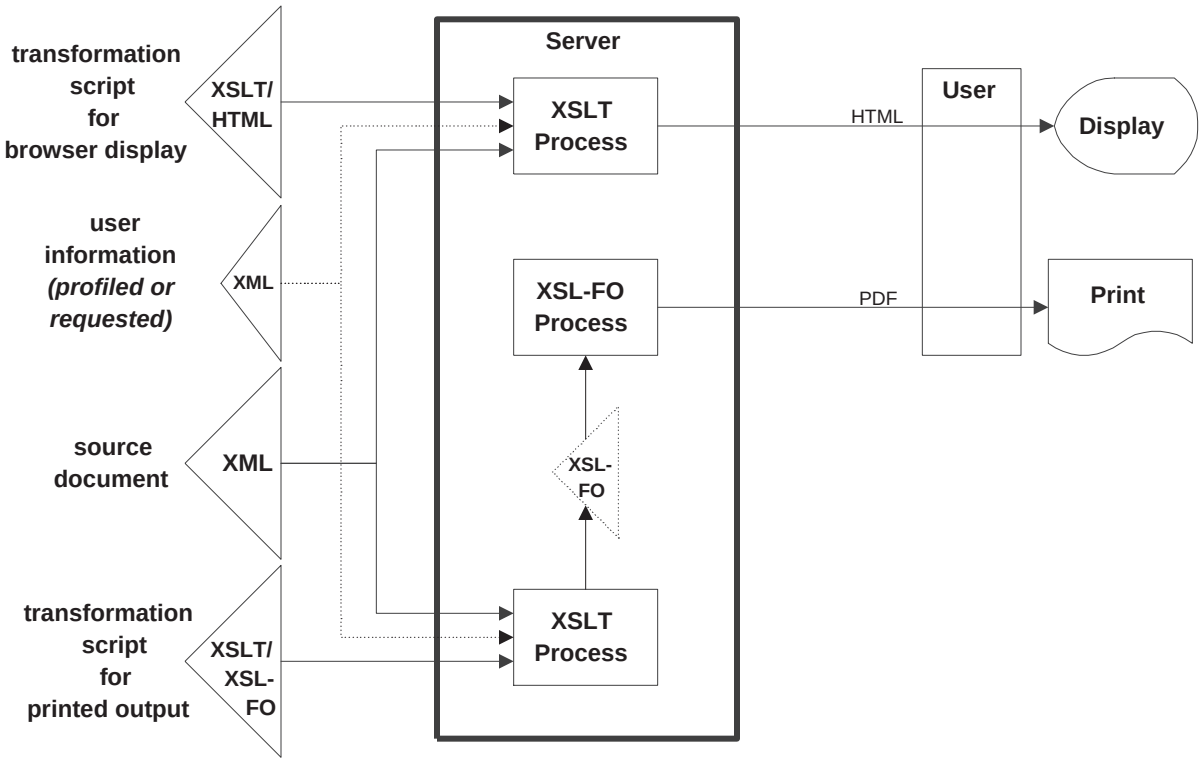


Copyright © Crane Softwrights Ltd.

Using XSL-FO on a server (cont.)

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

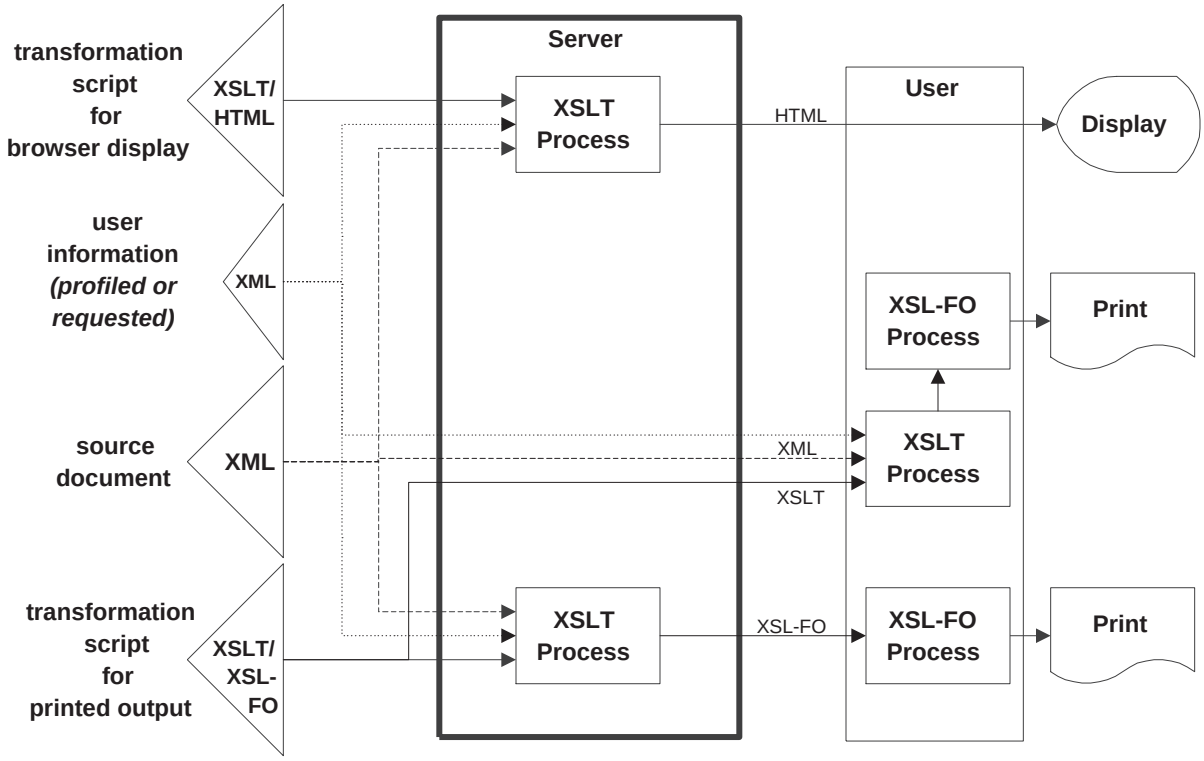


Copyright © Crane Softwrights Ltd.

Using XSL-FO on a server (cont.)

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



 Copyright © Crane Softwrights Ltd.

Historical development of the XSL and XSLT 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`

Copyright © Crane Softwrights Ltd.

XSL information links

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)

 Copyright © Crane Softwrights Ltd.

XSL information links (cont.)

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.

Copyright © Crane Softwrights Ltd.

XSL information links (cont.)

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.

 Copyright © Crane Softwrights Ltd.

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:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xmlns="http://www.w3.org/1999/XSL/Format"
      font-size="16pt">
  <layout-master-set>
    <simple-page-master
            margin-right="15mm" margin-left="15mm"
            margin-bottom="15mm" margin-top="15mm"
            page-width="210mm" page-height="297mm"
            master-name="bookpage">
      <region-body region-name="bookpage-body"
            margin-bottom="5mm" margin-top="5mm" />
    </simple-page-master>
  </layout-master-set>
  <page-sequence master-reference="bookpage">
    <title>Hello world example</title>
    <flow flow-name="bookpage-body">
      <block>Hello XSL-FO!</block>
    </flow>
  </page-sequence>
</root>
```

All examples illustrate instances of the XSL-FO vocabulary

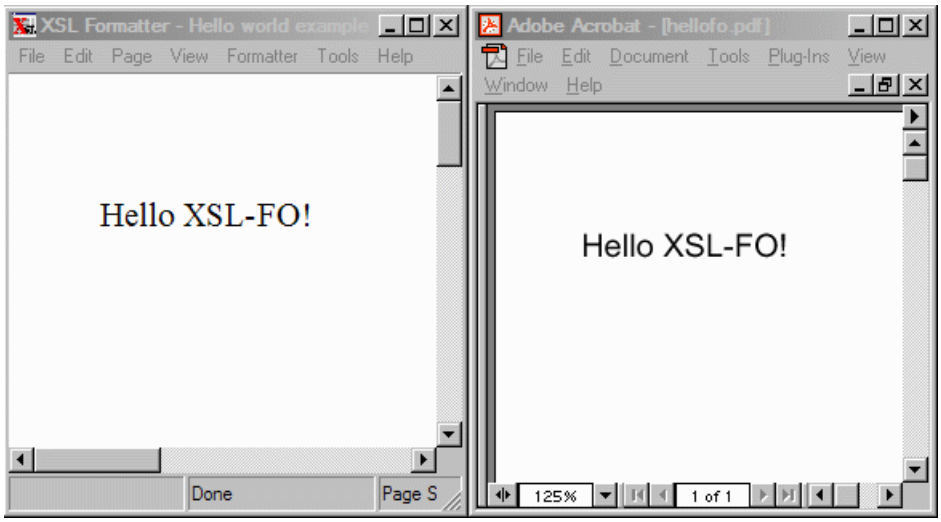
- how the instance is created is not material to the semantics of the vocabulary
  - could be hand-authored in a simple text or XML editor
  - could be the result of an XSLT transformation from another XML vocabulary
  - could be output from any application
- the default namespace is used in the examples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

Copyright © Crane Softwrights Ltd.

Hello world example (cont.)

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

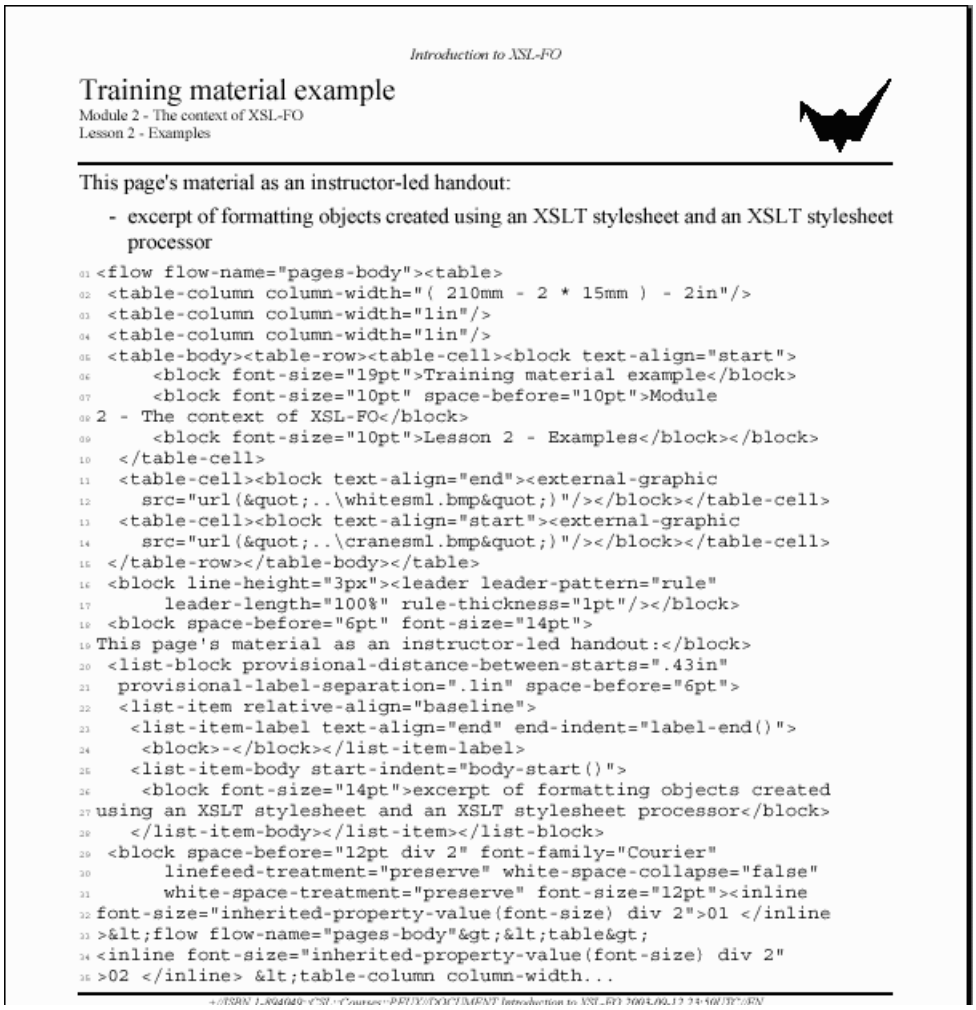
- a simple web page not using CSS properties relies on the browser settings
- the HTML constructs represent the intent of what is to be formatted
- absent formatting properties are satisfied using the tool option defaults

Copyright © Crane Softwrights Ltd.

A detailed example of flowed content

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

Consider a page of content from some instructor-led training material that contains a mixture of a table, a list, a proportionally-spaced paragraph, and mono-spaced paragraphs:



Introduction to XSL-FO

Training material example

Module 2 - The context of XSL-FO
Lesson 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

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

+//ISBN 1-894049-<CR>:<Courses>:PFUY:DOCUMENT Introduction to XSL-FO 2003-09-12 23:30:03:<EN

Copyright © Crane Softwrights Ltd.

Training material example

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

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

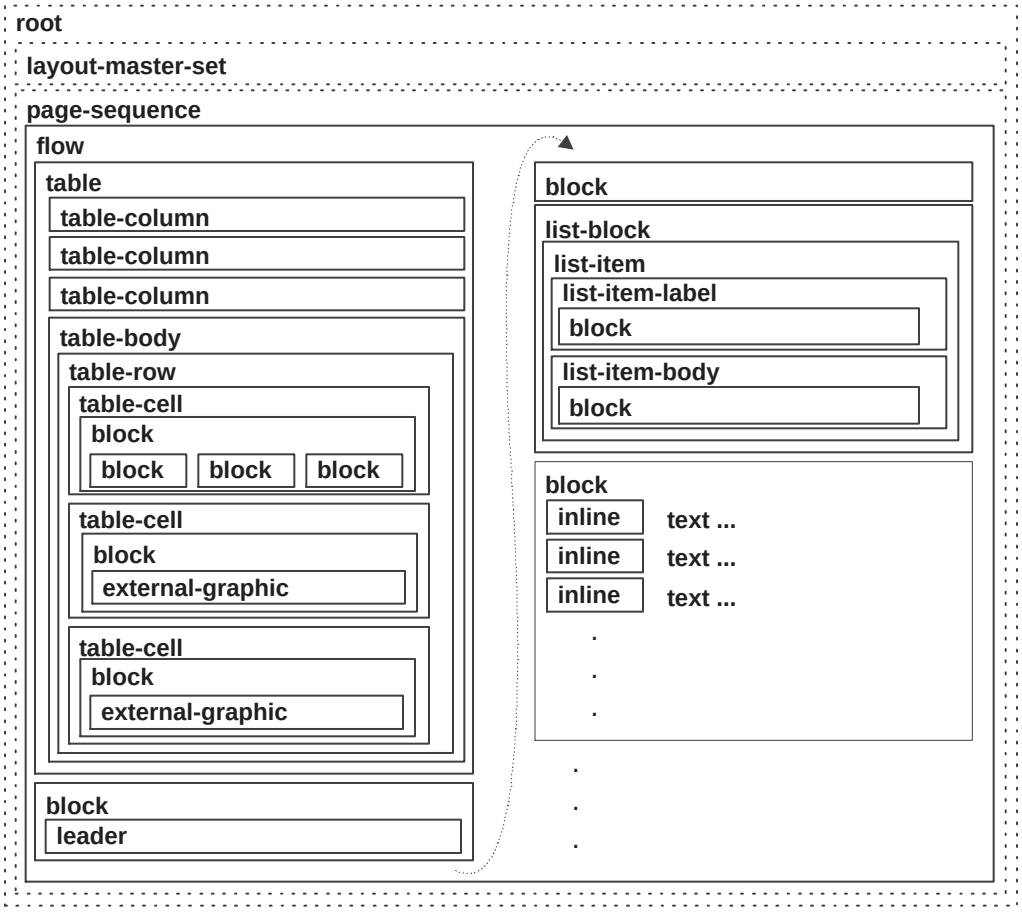
- excerpt of formatting objects created using an XSLT stylesheet and an XSLT stylesheet processor

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


Copyright © Crane Softwrights Ltd.

Training material example (cont.)

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



Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Essential concepts and terminology (cont.)

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

 Copyright © Crane Softwrights Ltd.

Layout-based vs. content-based formatting

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

Copyright © Crane Softwrights Ltd.

Formatting vs. Rendering

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 407).

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

 Copyright © Crane Softwrights Ltd.

Formatting vs. Rendering (cont.)

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

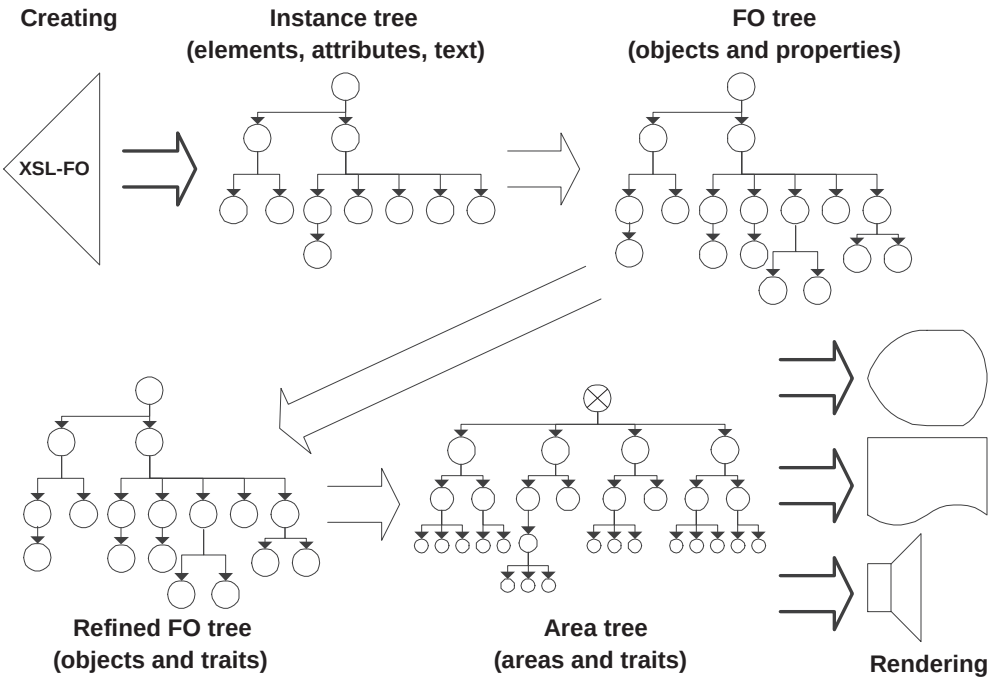
- production of another intermediate formatting language
  - e.g. TeX
- production of an electronic representation of the final form page description
  - e.g. Portable Document Format (PDF) by Adobe
  - e.g. Standard Page Description Language (SPDL) - ISO/IEC 10180
- interpretation and rendering to the 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

Copyright © Crane Softwrights Ltd.

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:



Copyright © Crane Softwrights Ltd.

Processing model of formatting (cont.)

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

Copyright © Crane Softwrights Ltd.

Processing model of formatting (cont.)

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

 Copyright © Crane Softwrights Ltd.

Vocabulary structure

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 409) for a list of all objects
- see Prototypical hierarchy (page 414) 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

Copyright © Crane Softwrights Ltd.

Vocabulary structure (cont.)

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 447) 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

 Copyright © Crane Softwrights Ltd.

Direct vs. constraint property specification

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 423).

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

Copyright © Crane Softwrights Ltd.

Property value expressions

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 404)

Numerous property data types

- summarized in Property data types (page 435)
- simple-valued and compound-valued values

Numerous functions available for expressions

- summarized in Functions summarized by name (page 404)
- 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.

Inherited properties

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 Inheritable properties (page 443) 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)

Copyright © Crane Softwrights Ltd.

Shorthand properties

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 446) for a summary
- portability problem when developing a stylesheet using one processor and delivering the stylesheet to someone else using a different processor

 Copyright © Crane Softwrights Ltd.

Conformance

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

Copyright © Crane Softwrights Ltd.

Top-level formatting objects

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
- `<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

 Copyright © Crane Softwrights Ltd.

<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:

- 0..1 (6.4.2) (layout-master-set,declarations?,page-sequence+)
- 1..1 (6.4.2) (layout-master-set,declarations?,bookmark-tree?,(page-sequence|page-sequence-wrapper)+)
- Child objects (alphabetical):
    - 1..1 <bookmark-tree>(6.11.1;298)
    - <declarations>(6.4.3;348)
    - <layout-master-set>(6.4.7;63)
    - <page-sequence>(6.4.5;65)
    - 1..1 <page-sequence-wrapper>(6.4.6;67)

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

id=(7.30.8;475)
1..1 index-class=(7.24.1;475)
1..1 index-key=(7.24.2;475)
media-usage=(7.27.11;482)

Property of interest:

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

Copyright © Crane Softwrights Ltd.

<root> Object (cont.)

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>
```

 Copyright © Crane Softwrights Ltd.

<layout-master-set> Object

Purpose:

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

Content:

- 1 (6.4.6) (simple-page-master|page-sequence-master)+
- 1 (6.4.7) (simple-page-master|page-sequence-master|flow-map)+
- Child objects (alphabetical):
  - 1 <flow-map>(6.4.22;246)
  - <page-sequence-master>(6.4.8;283)
  - <simple-page-master>(6.4.13;119)
- Referring object:
  - <root>(6.4.2;61)

No properties are defined for this formatting object.

Copyright © Crane Softwrights Ltd.

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

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>
```

 Copyright © Crane Softwrights Ltd.

<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;256)
    - <folio-suffix>(6.6.14;258)
    - <static-content>(6.4.20;253)
    - <title>(6.4.21;124)
- Referring objects:
    - <root>(6.4.2;61)
    - <page-sequence-wrapper>(6.4.6;67)

Required property:
master-reference=(7.27.9;482)

Optional properties:
country=(7.10.1;467)
flow-map-reference=(7.27.19;470)
force-page-count=(7.27.6;473)
format=(7.26.1;473)
grouping-separator=(7.26.2;473)
grouping-size=(7.26.3;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
initial-page-number=(7.27.7;476)
language=(7.10.2;478)
letter-value=(7.26.4;479)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Shorthand influencing the above properties:
xml:lang=(7.31.24;505)

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 442)

Copyright © Crane Softwrights Ltd.

<page-sequence> Object (cont.)

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>
```


Copyright © Crane Softwrights Ltd.

<page-sequence-wrapper> Object

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;475) 1 index-key=(7.24.2;475)
1 index-class=(7.24.1;475)

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

Copyright © Crane Softwrights Ltd.

<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;470)

Optional properties:
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

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

Copyright © Crane Softwrights Ltd.

<flow> Object (cont.)

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>
```


Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

Block-level objects

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

- `<block>`(6.5.2;109)
  - the formatting specification for a block of lines, distinct from any preceding block of lines
- `<block-container>`(6.5.3;227)
  - the specification of a block-level reference area for contained descendant blocks
- `<list-block>`(6.8.2;140)
  - the parent object of a related set of child list members
- `<table-and-caption>`(6.7.2;187)
  - the parent object of a captioned collection of tabular content
- `<table>`(6.7.3;192)
  - 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

Copyright © Crane Softwrights Ltd.

Inline-level objects

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

- `<basic-link>`(6.9.2;161)
  - the inline content of the start of a unidirectional link to a single end point
- `<bidi-override>`(6.6.2;361)
  - overriding the inherent Unicode text direction for a sequence of characters
- `<character/>`(6.6.3;350)
  - the inline display of a single character
- `<external-graphic/>`(6.6.5;152)
  - the inline display of graphical or other externally-supplied information
- 1.1 `<index-key-reference>`(6.10.6;313)
  - an inline display of resolved page numbers
- `<inline>`(6.6.7;112)
  - 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;228)
  - an inline reference area for contained blocks
- `<instream-foreign-object>`(6.6.6;155)
  - the inline display of graphical or other instance-supplied information
- `<leader>`(6.6.9;169)
  - the inline display of a rule or a sequence of characters
- `<multi-toggle>`(6.9.5;380)
  - the specification of those interaction-sensitive formatting objects
- `<page-number/>`(6.6.10;255)
  - an inline-level place holder replaced with the page number of the current page
- `<page-number-citation/>`(6.6.11;113)
  - an inline-level place holder replaced with the page number of the first normal area of the cited formatting object
- 1.1 `<page-number-citation-last/>`(6.6.12;114)
  - an inline-level place holder replaced with the page number of the last normal area of the cited formatting object
- 1.1 `<scaling-value-citation/>`(6.6.15;159)
  - 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

 Copyright © Crane Softwrights Ltd.

Neutral objects

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

- 1.1 `<change-bar-begin/>`(6.13.2;344)
  - the indication of the start of a change bar
- 1.1 `<change-bar-end/>`(6.13.3;345)
  - the indication of the end of a change bar
- `<multi-properties>`(6.9.6;367)
  - 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;373)
  - 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;267)
  - a place holder replaced with the formatting objects of the indicated marker and allowed only within static content
- 1.1 `<retrieve-table-marker/>`(6.13.7;269)
  - 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;108)
  - 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

Copyright © Crane Softwrights Ltd.

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;214)
    - 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

 Copyright © Crane Softwrights Ltd.

Out-of-line inline-level objects

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

- `<footnote>`(6.12.3;219)
    - 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

Copyright © Crane Softwrights Ltd.

Chapter 4 - Area and page basics

- Introduction - Area and page overview
- Section 1 - Area model details
- Section 2 - Block and inline basics
- Section 3 - Page definition and sequencing basics

 Copyright © Crane Softwrights Ltd.

Area and page overview

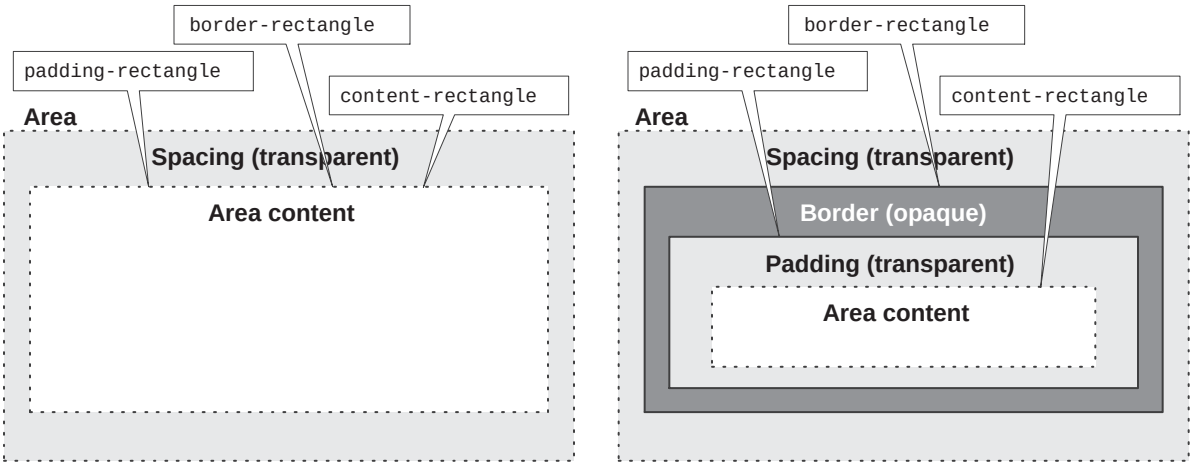
Chapter 4 - Area and page basics

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



Copyright © Crane Softwrights Ltd.

Area and page overview (cont.)

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

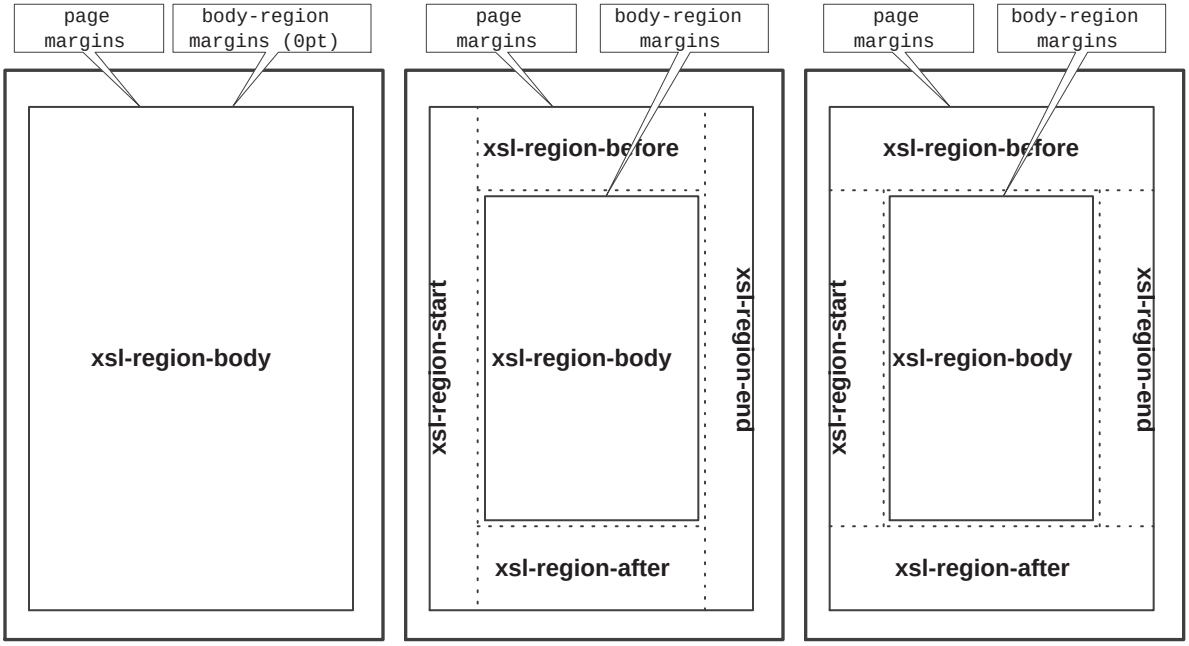
Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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 indications 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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

 Copyright © Crane Softwrights Ltd.

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
- 1 `<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

Copyright © Crane Softwrights Ltd.

Geometric rendered areas

Result of formatting is the generation of geometric areas for rendering based upon an area model

- a superset of CSS2 box formatting model
- includes references to areas by other areas
- relationships and space adjustment between areas generated for letters, words, lines and blocks

Areas arranged hierarchically as a result of interpreting formatting objects

- see Processing model of formatting (page 50)
- different branches of the tree collect areas that are related
    - a single XSL-FO object can generate areas for multiple area tree branches
- each has a relative or absolute position described by a spacing constraint
- each may have content to display
- each may have different visual or aural presentations than other areas
- placement of every glyph, shape and image with spacing constraints

Not all kinds of areas can be directly specified by the XSL-FO instance

- e.g.: lines in a block are areas synthesized only by the formatter during the flowing of information
- objects and properties in the XSL-FO instance can influence the presentation of the synthesized areas
    - e.g. `<initial-property-set>` object influencing the first generated line of a block
        - an empty formatting object that does not generate any areas
    - e.g. `line-height=` property governing how much of a block is taken up by each of the lines

A paginated area and static areas (regions) can be defined for each page

- paginated areas accept content to be flowed and trigger the generation of as many pages as needed to fit the flow of information to be rendered
    - defined as descendants of `<flow>` objects
- static areas are rendered on every page that is generated by pagination
    - defined as descendants of `<static-content>` objects
- a page's geometry need not contain any region that is accepting flow
    - such a page is rendered with only the static content and the formatter moves to the next page geometry in the sequence of pages

 Copyright © Crane Softwrights Ltd.

Geometric rendered areas (cont.)

A single object can create many areas

- not all generated areas of a given sequence need be adjacent nor hierarchically related to each other
  - one object can add areas to separate area tree branches
- the name of the formatting object that produces the desired areas are unimportant
  - the objective is to position information on the page and the choice of which formatting object used to create the needed areas is based on the semantics of the formatting object instead of the name of the formatting object
    - e.g. using a `<list-block>` to lay out aligned pairs of paragraphs of different languages even though the essence of the information being formatted isn't a list

The hierarchical area tree has familial relationships between nodes of the tree

- child, sibling, parent, descendant, ancestor, root
- a set of nodes is ordered within the parent
  - only one order of sibling nodes
  - initial, preceding, following, and final relationships
- two traversals of the tree when dealing with child nodes of the parent
  - pre-order traversal orders the parent before the children
    - e.g. determining the "next" area for `keep-with-next=`
    - e.g. determining preference in markers for `<retrieve-marker>`
  - post-order traversal orders the parent after the children
    - e.g. determining the "previous" area for `keep-with-previous=`
  - neither traversal order changes the order of the children themselves

The root node is not an area

- all other nodes are rectangular areas of result canvas
- areas in the tree may be of zero size
  - useful for carrying identification information in the tree without occupying real estate on the page
  - a page with only a zero-sized area is not an empty or blank page and is considered an area present on the page for the purposes of resolving the first and last areas in a reference area

Copyright © Crane Softwrights Ltd.

Geometric rendered areas (cont.)

Area tree can have very many areas described

- even every glyph (character shape) has its own area
- every area has its own placement direction and spacing constraint
- no obligation on the formatter to serialize or externalize area tree

Most nodes generate a singular area

- exception for ligatures that have leaf nodes combined to produce a single area
  - e.g. combinations of English letters into a single glyph such as "a" and "e" becoming "æ"
  - e.g. combinations of Arabic characters based on location and sibling characters

The formatter has sufficient information in the finalized areas to effect the rendering

- missing or conflicting constraints are resolved by the formatter in an implementation-dependent fashion
  - may be constrained by the rendering technology used by the formatter
- page fidelity is an important portability issue
  - two conforming XSL-FO agents need not produce the identical rendering
  - each formatter can create a different area tree based on interpretation of unspecified traits
    - e.g. hyphenation and justification rules
    - e.g. default value for inter-line leading
  - the rendering process can create a different result based on unspecified traits
    - e.g. font

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Directions and writing modes

Orientations and directions are defined for a reference area

- every reference area has an orientation with respect the definition of the "top" of the area and its writing direction
- descendent reference areas can have orientations modified relative to containing reference area
- descendent areas that are not reference areas cannot have their orientations changed

Reference areas are found in many places in the area tree, including:

- region-reference-area (perimeter regions)
- main-reference-area (page bodies)
- before-float-reference-area (before-floats)
- footnote-reference-area (footnotes)
- span-reference-area (spanned columns)
- normal-flow-reference-area (columns)
- lines in a block
- `<title>`
- `<block-container>` and `<inline-container>`
- `<table>`, `<table-caption>`, and `<table-cell>`

Stacking rules govern how adjacent areas created by objects in flow are related along directions

- all objects are defined as stacking in a particular direction relative to adjacent or other objects
- some objects are "out of line" in the stacking order
    - position is related to objects other than those that are adjacent
- general rules of thumb with powerful nuances available for specific requirements

Conditionality governs how spacing is accommodated at the beginning or termination of reference areas

- the default conditionality for `space-before=` and `space-after=` is "`discard`"
    - can be changed explicitly to "`retain`"
- space that can be discarded is discarded when it appears as the first or last area inside of a reference area
    - accommodates typical need to discard space at the tops of pages and columns when flowing paragraphs with space separations

Copyright © Crane Softwrights Ltd.

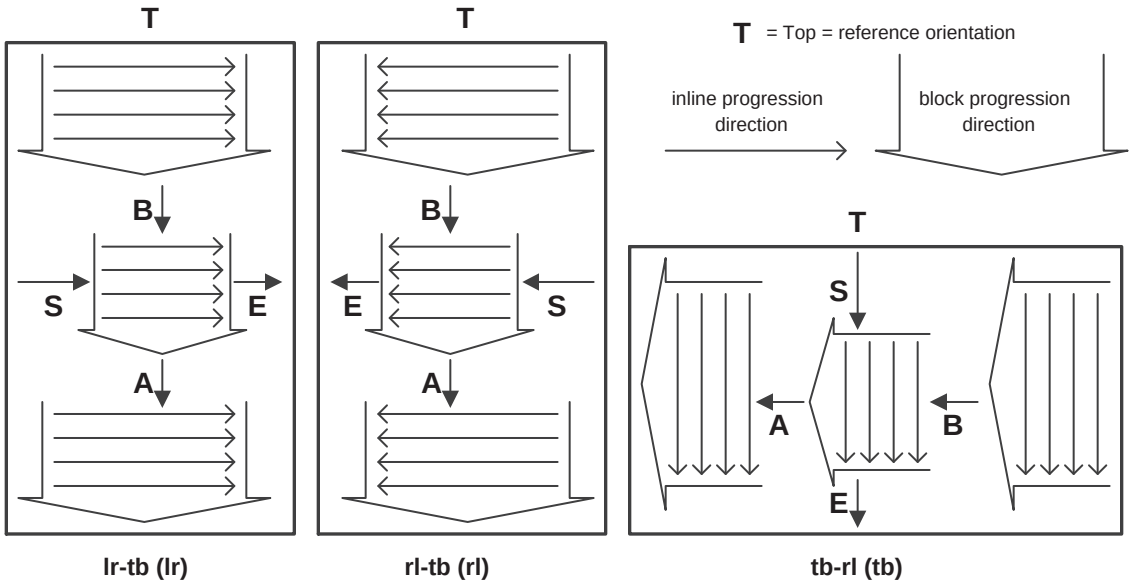
Directions and writing modes (cont.)

Various directions govern the relative placement of objects to their siblings

- inline-progression-direction
  - the stacking of glyph areas and other inline areas within a line
  - the stacking of cells in a table row
  - shift-direction
    - used for subscripts and superscripts rendered in inline areas
    - perpendicular to inline-progression-direction
    - inverse of the initial block-progression-direction
  - glyph-direction
    - direction of the top of the glyph
    - initially same as top of reference area
    - can be rotated or inverted clockwise relative to top of reference area
- block-progression-direction
  - the stacking of lines and blocks relative to other lines and blocks in an area
  - the stacking of rows in a table

Writing-mode-related properties reflect cultural practices of flowing information on a page:

- e.g. an indent at the start of a line in a left-to-right writing mode is on the opposite side of the page than an indent at the start of a line in a right-to-left writing mode
- before, after, start, end edges of each writing mode are relative to the reference orientation that defines the top edge of the area
  - using top, bottom, left and right ignores the writing mode and is less portable



Copyright © Crane Softwrights Ltd.

Directions and writing modes (cont.)

Two properties govern progression direction of adjacent areas on the page:

- `reference-orientation=`
  - defines the direction of the top edge of a reference area relative to containing reference area
  - positive (counter-clockwise) or negative multiples of 90 degrees
- `writing-mode=`
  - defines inline-progression-direction "-" block-progression-direction relative to the top of the reference area
  - `lr-tb` or `lr` = left-to-right for inline, top-to-bottom for block
    - as characterized by Western European writing systems
  - `rl-tb` or `rl` = right-to-left for inline, top-to-bottom for block
    - as characterized by Hebrew and Arabic writing systems
  - `tb-rl` or `tb` = top-to-bottom for inline, right-to-left for block
    - as characterized by traditional Japanese and Chinese writing systems
- combination can be used on other objects to define other direction-related traits
  - e.g. placement and orientation of perimeter regions within pages
  - e.g. placement of columns within body region
  - e.g. placement and orientation of rows and columns within tables
  - can be changed within paginated content and static content in a region
    - e.g. by using `<block-container>` objects at the block level
    - e.g. by using `<inline-container>` objects at the inline level
- can only be specified for reference areas

Bi-directional writing is inherently supported by XSL-FO processors:

- one of the most powerful features of XSL-FO is that the stylesheet writer need not be aware of the writing direction of the characters used in the XML documents being formatted
  - e.g. right-to-left characters can be intermixed with left-to-right characters
- the formatter automatically accommodates bi-directional writing based upon the inherent properties of the Unicode characters in the content
  - makes the job easier for the stylesheet writer in that the directions of the individual characters do not need to be detected and accommodated
- `<bidi-override>` object only used when direction inferred by the Unicode character is not as desired
  - a common misconception is that this must be used to support bi-directionality when, in fact, it is used only when the bi-directionality needs to be embedded or overridden

Copyright © Crane Softwrights Ltd.

Margins, spaces and positioning

Areas are positioned by specifying the spaces around them

- areas that are flowed adjacent to each other are separated by their spacing specifications
- floats and footnotes are positioned out-of-line as defined by the Recommendation
- only a `<block-container>` can be positioned arbitrarily on the page
    - indicated using `absolute-position=`
        - by a relative distance from its siblings
            - as if it were a simple block ("`auto`")
        - by an absolute distance from an ancestor
            - from the parent area boundaries ("`absolute`")
            - from the page area boundaries ("`fixed`")
    - distances from ancestral area edges are specified using `top=`, `bottom=`, `left=`, `right=`

Spaces and non-page margins define the same values, but from different perspectives

- `space-*=` specifications are relative to the writing direction
    - `space-before=`, `space-after=`, `space-start=`, `space-end=`
- `margin-*=` specifications are relative to the reference orientation
    - `margin-top=`, `margin-bottom=`, `margin-left=`, `margin-right=`

Page margins

- margins of page regions are fixed to the page boundaries and physical orientation
    - `margin-top=`, `margin-bottom=`, `margin-left=`, `margin-right=`

Discarding space

- a space specification can have a property of being discarded when the area it is associated with is the first or last in a reference-area

 Copyright © Crane Softwrights Ltd.

Length values for widths and spacing

Lengths can be specified in both absolute and relative terms:

- `cm` - centimeter
- `mm` - millimeter
- `in` - inch = 2.54cm
- `pc` - pica = 12pt = 1/6in
- `pt` - point = 1/72in
- `px` - pixel = 1 device dot (nominally 1/90" or .28mm)
- `em` - current font size (only relative unit of measure)
    - use em-based values for spacing to protect relative appearance from changes in font size (e.g. "`2em`" is twice the current font size)
- note that there is no "ex" measurement
    - sometimes used in formatting specifications as the relative height of a lower case letter in the current font size
    - this is not a concept easily interpreted in all international character repertoires
    - this is used in XSL-FO in the definition of one of the baselines for those scripts that have an ex value

Lengths have a number of components

- minimum, maximum and optimum
    - preferred at optimum but no less than minimum nor more than maximum
- conditionality
    - "`retain`" or "`discard`" controls whether the space-specification is preserved at the beginning or termination of a reference-area
- precedence
    - "`force`" or an integer value controls which of the space-specifications is in play after constraint calculation resolution
        - adjacent sibling space-specifications interact
            - sizes and precedence determine rendered amount of space
            - detailed at Spacing, conditionality and precedence (page 333)

Some properties can be specified using a percentage

- when inherited, percentages are relative to their inherited value
    - e.g. `font-size="150%"`
- when not inherited, percentages are relative to a reference rectangle
    - content-rectangle for XSL-defined properties
    - closest non-line-area content-rectangle for CSS-defined properties
    - exceptions exist for out-of-line constructs and the regions in which they are used (detailed in Recommendation section 7.3)

Copyright © Crane Softwrights Ltd.

Area types

Chapter 4 - Area and page basics
Section 1 - Area model details

Different areas are characterized by their different purposes and uses:

- reference-area - an area that can have different reference-related properties than its parent area
    - by changing reference orientation and/or writing mode
    - also has special behaviors for first and last space specifications
- viewport-area - the clipping/scrolling visible portion of a reference area
- allocation-area - the basis of positioning and alignment of content within parent
    - defines the mechanics of how the stacking works
- block-area - a collection of lines in the block-progression-direction
    - creates a new set of lines, even if block has a zero dimension
- line-area - a collection of inline constructs in a block
- inline-area - atomic construct or collection of glyph constructs in the inline-progression-direction
- glyph-area - single text character

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Stacking area and rectangle relationships

Areas stack in either the block-progression-direction or the inline-progression-direction using rectangles

- the parent area's content-rectangle constrains the position of the child areas therein
- the child areas can only stack in one direction relative to the parent reference orientation
    - one set of opposing edges of child co-incident with the parent content-rectangle
    - other set of opposing edges of child co-incident with the adjacent content-rectangle
        - or with parent content-rectangle if first or last in the stack

Areas that are stacked adjacent to their siblings are considered "normal"

- other areas are considered "non-normal" and stack elsewhere in the area tree
    - e.g. floats to the before, start, or end sides of the body region
    - e.g. footnote bodies to the after edge of the body region
    - e.g. absolutely positioned block containers to arbitrary locations

Property initial values are for "common sense" formatting

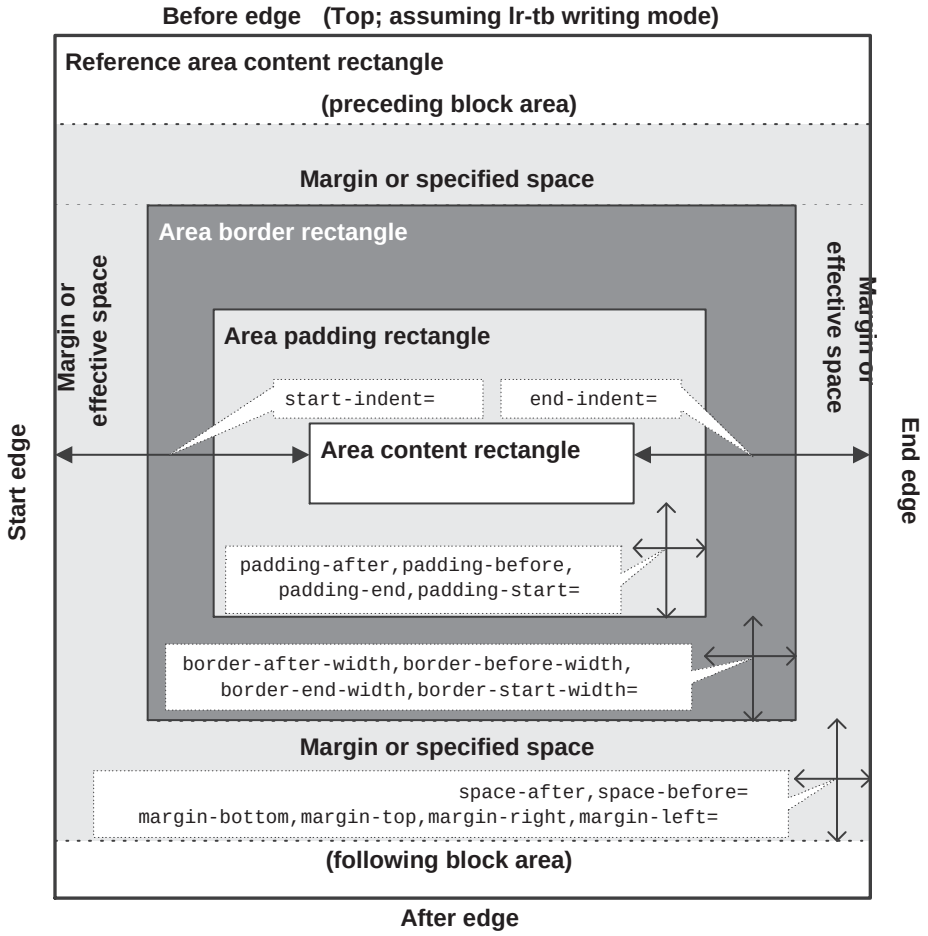
- a lot of work can be done without changing initial values
- a lot of finesse can be accomplished by specifying available alternatives

Copyright © Crane Softwrights Ltd.

Stacking area and rectangle relationships (cont.)

An area is stacked within its parent's content-rectangle using spacing extents:

- names of extents are suffixed using the names of the edges from which they extend
  - allows each edge of the spaces, border or padding to have individual values
  - shorthand properties are available to set all extents in one specification
- reference areas can have a different location of "top" than their parent area



A block's child area without borders is simply positioned between siblings

- `space-before=` and `space-after=` between siblings
- `start-indent=` and `end-indent=` within parent

Copyright © Crane Softwrights Ltd.

Stacking area and rectangle relationships (cont.)

All child areas are contained within nested rectangles

- rigorous specification of formatting intent, spaced relative to directions of the parent area
- border-rectangle - positioned relative to parent or adjacent area using `margin-*=`, `space-*=` or `*-indent=` properties
  - edges of border-rectangle identified by parent reference orientation
  - `space-start=` and `space-end=` are ignored for block-level constructs
  - `space-before=` and `space-after=` are ignored for inline-level constructs
  - specified space results from `space-before=` and `space-after=`
  - effective space results from `start-indent=` and `end-indent=`
- padding-rectangle - positioned relative to border-rectangle using `border-*=` width properties
  - edges of padding-rectangle identified by parent reference orientation
- content-rectangle - positioned relative to padding-rectangle using `padding-*=` spacing properties
  - edges of content-rectangle identified by child reference orientation

The border rectangle is different than the spacing and padding rectangles

- spacing and padding rectangles and dimensions define invisible non-rendered areas
- the border rectangle and border width properties define the thickness of a visible rendered border
- not shown in the diagram are the `*-top=`, `*-bottom=`, `*-left=` and `*-right=` properties for `padding-*=` and `border-*=` which directly correspond to the writing direction dependent properties

Border and padding widths are sometimes ignored

- e.g. not applicable to regions, floats or footnote bodies
- `border-style=` value of "`none`" or "`hidden`" will set the border width to `0pt`
- `border-before-width=` and `border-after-width=` have `.conditionality` component with the initial value of "`discard`"
  - sets the `.length` component to `0pt` if is, respectively, the first or last area in a reference area
  - may be set to "`retain`" to preserve the `.length` component value

Copyright © Crane Softwrights Ltd.

Stacking area and rectangle relationships (cont.)

An area's content-rectangle in turn contains the area's children

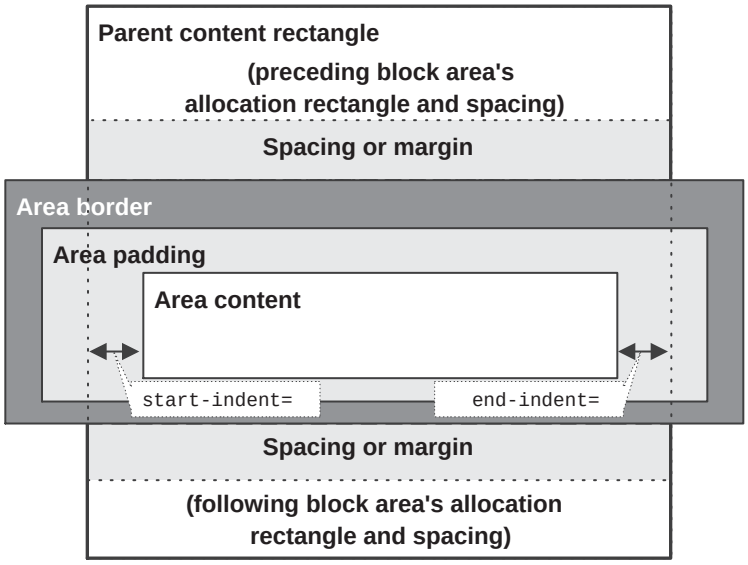
- the area's type governs what is allowed for the child areas
  - only a reference-area's children can discard conditional spaces
  - only when the area is a reference-area can the content-rectangle have a orientation properties different than itself
    - orientation properties and traits are relative to the "top" of the area
    - e.g.: `reference-orientation=`, block-progression-direction, etc.
  - an area's content-rectangle orientation properties are relative to the content's own reference orientation, not the area's reference orientation
- proportional `*-indent=` and `margin-*=` specifications are calculated differently
  - a proportional `*-indent=` specification is based on the ancestral reference-area
  - a proportional `margin-*=` specification is based on the containing block
- specifying conflicting values will over-constrain requirements
  - `margin-*=` properties take precedence over `*-indent=` properties
    - a specified `start-indent=` or `end-indent=` value is ignored if the corresponding `margin-*=` property within the indent value is specified
  - `start-indent=` and the content's specified inline-progression dimension take precedence over `end-indent=`
    - e.g. `inline-progression-dimension=` on the object or specified for the objects children's sizes
    - if the inline-progression dimension is not explicitly set for the object or its children, the `end-indent=` is respected

 Copyright © Crane Softwrights Ltd.

Stacking area and rectangle relationships (cont.)

A small indent pushes the border outside of the parent area

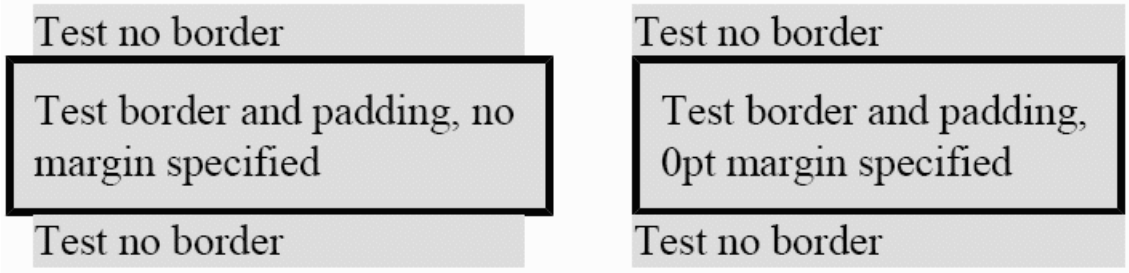
- without a margin specified, the border and padding move outside of the start indent



Specifying a margin property changes the drawing of the border

- with a margin specified, the border and padding start inside the parent area

In the following the middle block on the left has no margin specified, while the middle block on the right has a margin of 0pt specified:



- note how the letters "T" are aligned on their start side when no margin is specified

Copyright © Crane Softwrights Ltd.

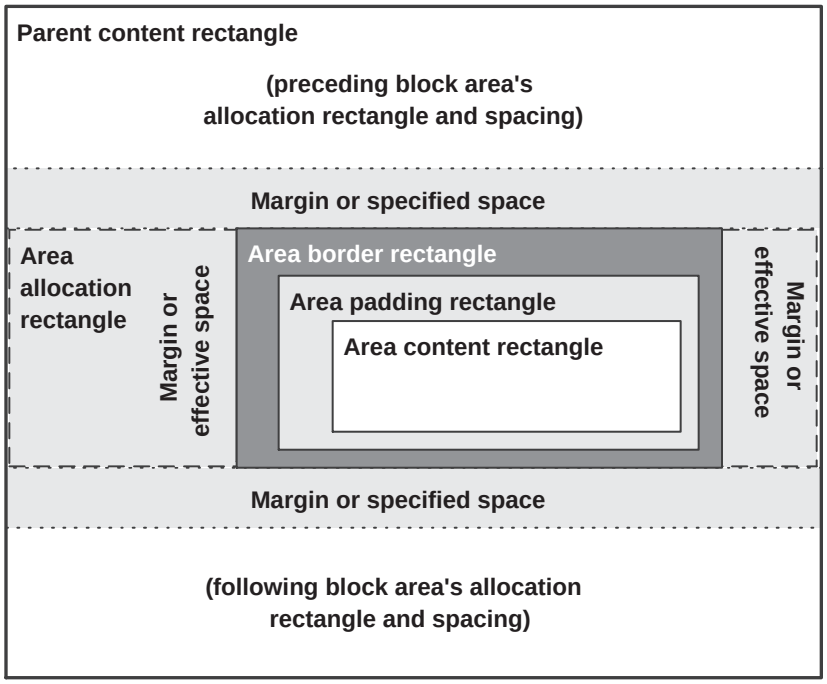
Allocation rectangles and alignment

A child area's `allocation-rectangle` describes constraints of position within parent

- defines the mechanics of the stacking of adjacent areas
- dimension based on either the parent, border or content edges
  - block and inline areas use different edges
  - inline areas of two different types (normal and large) use different edges
- orientation based on parent area
  - may be different than content area
- the start edge of an inline area's `allocation-rectangle` defines the `alignment-point` of the area
  - used in glyph and other inline rendering
- the space outside of the allocation rectangle can be discarded
  - only when the space is the first or last in a reference area
  - the `space-*.conditionality` default of "`discard`" can be overridden to be "`retain`"

Block area stacking and `allocation-rectangle` definition:

- blocks are as wide as their parent area
  - this forces the stacking in the block-progression direction
- all block areas are defined with the same relationship between sibling block areas

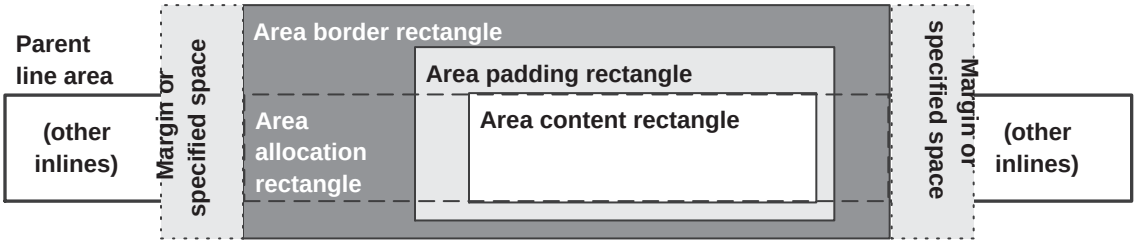


 Copyright © Crane Softwrights Ltd.

Allocation rectangles and alignment (cont.)

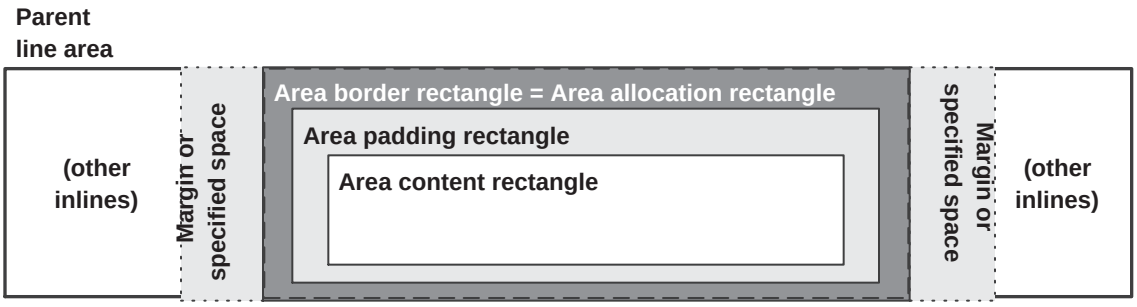
Inline areas defined with a `normal-allocation-rectangle`:

- most inline areas are defined with the normal relationship between areas
- e.g. text
- note how the bordering of such an area will not influence the line stacking and may cause undesirable overwriting of content on the previous and following lines



Inline areas defined with a `large-allocation-rectangle`:

- some inline areas are defined with an alternative relationship between areas
- e.g. inline-containers and graphic images
- note how the bordering of such an area will influence the line stacking and will not cause undesirable overwriting of content on the previous and following lines when the line-stacking strategy is based on the height of the content



Copyright © Crane Softwrights Ltd.

Line areas

Line areas are special areas only created by creating a block area

- line area traits are specified in the block properties
- stack in the block-progression-direction
- no borders or padding

Line stacking strategy determines which rectangle describes the line

- "`nominal-requested-line-rectangle`" - based on font-height
  - selected using `line-stacking-strategy=` of "`font-height`"
  - above the text baseline by the text altitude
  - below the text baseline by the text depth
  - provides constant baseline-to-baseline spacing
- "`maximum-line-rectangle`" - (initial) based on inline objects within line
  - selected using `line-stacking-strategy=` of "`max-height`"
  - maximum of the `allocation-rectangle` of all contained inline objects and the `nominal-requested-line-rectangle`
  - provides constant space between line areas
- "`per-inline-height-rectangle`" - based on leading
  - selected using `line-stacking-strategy=` of "`line-height`"
  - provides CSS-style line box stacking when using zero space-before and space-after

Sometimes the only changes to line height are a result of using `baseline-shift=`:

- e.g. superscript and subscripted text
- using `line-height-shift-adjustment=` value of "`disregard-shifts`" will allow the line stacking strategy to only consider other constructs on and situations regarding the line

 Copyright © Crane Softwrights Ltd.

Line areas (cont.)

Leading is calculated as difference between line height and font size

- leading is the space found between the lines of a block
- the `half-leading` trait is used by the formatter before and after lines
  - the use of half-leading accommodates adjacent lines with vastly different font sizes without having to decide which line's leading should apply
  - both lines' `half-leading` trait applies in the gap between lines
- initial value of `line-height=` is "`normal`"
  - processor can implement any "reasonable" (sic) value based on font size
  - value from "1.0" to "1.2" is recommended
  - when not specified, different initial values in different processors will yield different area trees and resulting formatting
    - important portability issue
- can override "`normal`" by specifying a number (e.g. "1" or "1.1") or a percentage factor of current font size
  - a scaling factor number is inherited and applies to descendent font sizes
  - a percentage value is calculated where specified and the calculation result is inherited (not the percentage)
- can also specify absolute length value

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Line areas (cont.)

Geometric font characteristics are based on the em box

- relative measure of the height of the glyphs in the font
- a box that is 1 em by 1 em in dimension is called the design space
    - note that creating an `<inline-container>` with a square dimension of 1em, aligned on the after-edge of the line and with a visible border will render a check-box without having to use a graphic image
- design space origin (dominant baseline) is different for each of ideographic, Indic, and the group of all other scripts (e.g. alphabetic and syllabic)

Inline constructs align along one of many baselines defined at the same time as detailed in Area Alignment Properties (7.13):

- when not specified, the automatic dominant baseline is based on the current font
- alphabetic
    - used for most alphabetic and syllabic scripts
        - e.g. Western, Southern Indic, non-ideographic Southeast Asian, etc.
- ideographic
    - used by ideographic scripts for the bottom of the ideographic em box
- hanging
    - used for certain Indic scripts
- mathematical
    - used for mathematical symbols
- after-edge
    - after-edge of the reference area after accommodating line contents
- before-edge
    - before-edge of the reference area after accommodating line contents
- central
    - computed center of the em box
- middle
    - center of the ex box sitting on the alphabetic baseline
    - approximated by "central" when the script doesn't have "x-height"
    - use `vertical-align=` when the graphic is taller than the line
- text-after-edge
    - equal to the ideographic baseline for ideographic scripts
    - equal to the bottom of the space for descending characters for non-ideographic scripts
        - often slightly after than the ideographic baseline
- text-before-edge
    - 1em before the text-after-edge

 Copyright © Crane Softwrights Ltd.

Line areas (cont.)

Consider the `samp/align.fo` example, modifying values for `alignment-baseline=`:

Test 1: Xy:
Test 2: Xy: :yX - default; external-graphic
Test 3: Xy: :yX - default; inline-container
Test 4: Xy: :yX - after-edge; external-graphic
Test 5: Xy: :yX - after-edge; inline-container
Test 6: Xy: :yX - middle; external-graphic
Test 7: Xy: :yX - middle; inline-container
Test 8: Xy: :yX - middle; inline-container; no leading

1 a bordered block with simple text along the dominant baseline with descending letters
2 a graphic image sitting on the dominant baseline (default)
   - without manipulating `alignment-baseline=`, the graphic sits "high" on the line
3 a square inline-container of size 1em sitting on the dominant baseline (default; alphabetic in this example)
   - without manipulating `alignment-baseline=`, the container sits "high" on the line
4 a graphic image aligned to the after-edge baseline
   - this often looks more appealing by having a common edge with the text
5 a container aligned to the after-edge baseline
   - this effect aligns all with the "bottom" of the text box
   - this often looks more appealing by not going above the top of the text characters
6 a graphic image aligned to the middle baseline
   - the center line of the graphic is centered to the "middle" of the text box
7 a container aligned to the middle baseline
   - the center line of the inline container is centered to the "middle" of the text box
8 turning off the leading by using `line-height="1"`
   - this shows the before/after edges of the inline container touching the inside edges of the parent line area
   - the half-leading seen in the previous example both above and below the inline container has been removed by adjusting the line height

Copyright © Crane Softwrights Ltd.

Superscript and subscript

There is often the requirement to subscript or superscript text

- use `baseline-shift=` to shift a character's baseline in a direction perpendicular to the flow
- positive values move up when writing in the left-to-write direction
- percentages are based on the line height of the parent area

By default, the movement of a character affects the line height of the line

- will produce incongruous results causing the leading to appear different only on the lines with superscripts or subscripts
- the leading of the entire line can be protected when using the `line-height-shift-adjustment=` property to disregard the shifts

Be careful of apparent line height when using fonts for superscripts and subscripts

- in one real-world example a Greek letter was needed for a subscript and `line-height-shift-adjustment=` was set to disregard the shifts, yet the one line of the block appeared to have the incorrect leading
- it turned out the stylesheet writer used a larger font for just the Greek character to get the desired appearance in relation to the other text, not realizing that the use of the one larger font character changed the maximum height of objects on the line, thus triggering the difference in line height for the entire line
- the fix was to change the `line-stacking-strategy=` to be "`line-height`", thus basing the line spacing on the leading

 Copyright © Crane Softwrights Ltd.

References in the area tree

First normally-flowed area created by an object with a specified `id=` property will have the value as a trait

- recall the area tree produced by interpreting the objects in Processing model of formatting (page 50)
- the application generating the XSL-FO instance is responsible for the uniqueness of `id=` values
- first area with identifier found in the area tree is used (it is not an error to have multiple uses of the same identifier)

Objects can refer to other objects in the tree creating a reference to the area produced

- page number citations using `ref-id=`
- hyperlink targets using `internal-destination=`
- when more than one object has the same identifier, references are resolved to the first area in the area tree created by the first object with the identifier
- when no object has the required identifier, the formatter signals an error of being unable to meet the requirements

Note it is unsafe to use XML ID and IDREF constructs directly in an XSL-FO instance when combining XML source documents for aggregation

- each document has independent value spaces for unique ID values
- using the ID values from multiple documents in a single XSL-FO document can introduce value conflicts
    - the same ID may have been used in two documents and will result in being doubly used in the XSL-FO document

It is safest when using XSLT to convert every ID and corresponding IDREF to the `generate-id()` value for the node for the element with the ID

- the value space for `generate-id()` is guaranteed to be unique for every node of all node trees used in a single transformation process
- for the element with the ID identifier, use the generated identifier of the element
    - e.g. `id="{generate-id(.)}"`
- for the reference with the IDREF identifier, use the generated identifier of the element referenced
    - e.g. `ref-id="{generate-id(id(@idref))}"`
    - e.g. `internal-destination="{generate-id(id(@idref))}"`

Also the safest when synthesizing relationships where ID/IDREF is not used

- e.g. using `generate-id()` for automatically generated tables of content

Copyright © Crane Softwrights Ltd.

Simple block and inline objects

The `<wrapper>` object is a semantic-free container used by other objects or for inherited properties

- does not generate any areas itself
    - returns the areas generated by its descendants
- used by `<multi-properties>` as a wrapper of objects to which the multiple properties apply
- introduces a branch in flowed content in the formatting object tree from which descendent objects can obtain inherited property settings (see Processing model of formatting (page 50))

Only allowed to be positioned where its children are allowed to be positioned

- cannot be used as a short cut to avoid the rules of parentage, such as for inline constructs

 Copyright © Crane Softwrights Ltd.

Simple block and inline objects (cont.)

Only way to create line areas on a page is to create a block of lines

- `<block>`
  - one property for the first line in the block
    - `text-indent=`
      - a positive value incurs into the block (indent)
      - a negative value hangs outside the block (outdent)
  - two properties for the last line in the block
    - `last-line-end-indent=`
      - a positive value incurs into the block (indent)
      - a negative value hangs outside the block (outdent)
    - `text-align-last=`
      - the alignment of only the last line in the block
      - also impacts lines ending with preserved linefeed characters
  - a number of properties for all lines in the block (including first and last)
    - note for `text-align=` the value "`justify`" doesn't apply to the last line of the block
- `<initial-property-set/>`
  - solely a container of properties applicable to the first line
    - only the formatter can determine how much of the flowed information in the block will end up on the first line of the block
    - neither XSLT nor any other XSL-FO generation process can know ahead of time how much will be formatted on the first line of a block
  - must be a child of the block defining the lines to be influenced
- the line of a single-line block is considered to be both the first line and the last line of the block

Inline constructs are placed by default on the dominant baseline

- for non-ideographic fonts results in unexpectedly shifted visual results for graphics and inline containers
- can be adjusted to other baselines using `alignment-baseline=` and `vertical-align=`

Constructs designed to be used only inline can be rendered on their own line

- e.g. graphics needed to be rendered between paragraphs
- e.g. a rule-styled leader needed to be rendered between paragraphs
- to render an inline construct as a block it must be placed within its own block

Copyright © Crane Softwrights Ltd.

Simple block and inline objects (cont.)

Inline areas can have non-inheritable properties set for the area contents

- `<inline>`
  - distinct from `<wrapper>` in that it creates an area in the area tree
  - necessary for non-inherited properties such as `baseline-shift=` for doing superscript and subscript of the contained text

The formatter can supply characters to be rendered in a line

- `<page-number-citation/>` is replaced with the first page number for the areas created by the cited formatting object
  - `ref-id=` points to the encompassing formatting object for the areas in question with an `id=`
  - the choice of characters is dictated by the page sequence for the cited area
  - e.g. consider a book with the front matter numbered in lower-case roman numerals and the book body in digits
    - citing a page of the body when in the front matter returns the digits as used in the body
    - citing a page of the front matter when in the body returns the roman numerals as used in the front matter

 Copyright © Crane Softwrights Ltd.

Simple block and inline objects (cont.)

- Generating a total page count
  - `<page-number-citation-last>` is replaced with the last page number for the areas created by the cited formatting object
    - the areas may be created by descendants of the formatting object or by the influence of neighboring formatting objects
      - a page is blank if conditions cause the formatter to create a page without any flowed content (see Forced blank pages (page 281) for details)
    - `ref-id=` points to the encompassing formatting object for the areas in question with an `id=`
      - e.g. for the page count of the entire book, pointing to the `<root>` would suffice
      - e.g. for the last page of a chapter, pointing to the chapter's `<page-sequence>` would suffice
    - `page-citation-strategy=` indicates which last page of the object is used
      - "`all`" (default)
        - the last of all pages for the formatting object
      - "`normal`"
        - the last page not including any trailing out-of-line constructs (e.g. floats and footnotes) or blank pages
      - "`not-blank`"
        - the last page not including any trailing blank pages
- Generating a total page count in XSL-FO 1.0
  - citing the page number of the last page of the document gives the total page count
  - an empty block with an area with zero dimension can be flowed on the last page
    - not robust for all possible documents, but this works for most documents and is the only way possible in XSL-FO 1.0
      - nuances of interaction with floats, float separator sub-regions and discarded space specifications prevent guaranteed page numbering
  - for the last page of a chapter, an empty block can be flowed at the end of the chapter and then cited

Tip when using XSLT to guarantee uniqueness of the root node or the last block identifier:

- use `id="{generate-id(/)}"` as an identifier not in conflict with ID/IDREF
  - identify the encompassing root node
  - identify the last block of the document
  - the root node of the source node tree cannot be referenced by IDREF
  - guaranteed to be unique from all source documents' use of IDREF when all identifiers are translated using the `generate-id()` function

Copyright © Crane Softwrights Ltd.

<wrapper> Object

Chapter 4 - Area and page basics
Section 2 - Block and inline basics

Purpose:

- a neutral container construct for specifying inherited properties for descendent constructs

Content:

- (6.13.4) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- Referring object:
  - <multi-properties>(6.9.6;367)
- may begin with any number of <marker> children

Optional properties:

id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

Copyright © Crane Softwrights Ltd.

<block> Object

Chapter 4 - Area and page basics
Section 2 - Block and inline basics

Purpose:

- the description of a new set of lines distinct from preceding area content

Content:

- (6.5.2) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- may begin with (in the following order):
  - zero or more <marker> objects
  - at most one <initial-property-set> object

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Hyphenation Properties(7.10;431)
- Common Margin Properties-Block(7.11;432)
- Common Relative Position Properties(7.13;434)

Other optional properties:

break-after=(7.20.1;463)
break-before=(7.20.2;463)
clear=(7.19.1;465)
color=(7.18.1;465)
hyphenation-keep=(7.16.1;474)
hyphenation-ladder-count=(7.16.2;474)
id=(7.30.8;475)
1 index-class=(7.24.1;475)
1 index-key=(7.24.2;475)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
last-line-end-indent=(7.16.3;478)
line-height=(7.16.4;479)
line-height-shift-adjustment=(7.16.5;480)
line-stacking-strategy=(7.16.6;480)
linefeed-treatment=(7.16.7;480)
orphans=(7.20.6;484)
span=(7.21.4;497)
text-align=(7.16.9;500)
text-align-last=(7.16.10;500)
text-altitude=(7.29.4;500)
text-depth=(7.29.5;501)
text-indent=(7.16.11;501)
visibility=(7.30.17;502)
white-space-collapse=(7.16.12;503)
white-space-treatment=(7.16.8;504)
widows=(7.20.7;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Preserving white space

Default properties collapse input white space

- physical lines of text in an XSL-FO instance are streamed along line areas of the page
- line ends from the input by default are treated as spaces

Often necessary to preserve the exact text content of an element

- see example on Training material example (page 42)
- program listings
- markup listings
- text drawings

Combination of properties required to ensure all text is preserved:

- `linefeed-treatment="preserve"`
  - to preserve linefeed characters during refined formatting object tree generation
- `white-space-treatment="preserve"`
  - to preserve white space around linefeed characters during refined formatting object tree generation
- `white-space-collapse="false"`
  - to preserve consecutive white space during area tree generation

Optionally, one could also specify the behavior of content too long for a line:

- `wrap-option="no-wrap"`
  - to clip the line and trigger an overflow error for lines that are too long

Note that all four properties can be manipulated using the `white-space=` shorthand

- remember that shorthand properties are not required to be supported by a processor

 Copyright © Crane Softwrights Ltd.

`<initial-property-set>` Object

Purpose:

- an auxiliary construct for specifying properties applied to the first line of the parent block
  - the information to which the properties applies is determined by the formatter and not by the generation of the block

Content:

- (6.6.4) `EMPTY`

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Relative Position Properties(7.13;434)

Other optional properties:

`color=`(7.18.1;465)
`letter-spacing=`(7.17.2;479)
`line-height=`(7.16.4;479)
`score-spaces=`(7.30.15;495)
`text-decoration=`(7.17.4;500)
`text-shadow=`(7.17.5;501)
`text-transform=`(7.17.6;501)
`word-spacing=`(7.17.8;504)

Shorthand influencing the above properties:

`font=`(7.31.13;470)

Copyright © Crane Softwrights Ltd.

<inline> Object

Chapter 4 - Area and page basics
Section 2 - Block and inline basics

Purpose:

- the specification of inherited and non-inherited properties for content within a line generated in a block

Content:

- (6.6.7) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- Referring object:
  - <footnote>(6.12.3;219)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
block-progression-dimension=(7.15.3;453)
color=(7.18.1;465)
dominant-baseline=(7.14.5;468)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
line-height=(7.16.4;479)
text-decoration=(7.17.4;500)
visibility=(7.30.17;502)
width=(7.15.14;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)
vertical-align=(7.31.22;502)

Properties of interest:

- baseline-shift= used for subscripting and superscripting
- text-decoration= used for underscored text
- font-style= used for italicized text
- font-weight= used for boldfaced text

Copyright © Crane Softwrights Ltd.

<page-number-citation> Object

Chapter 4 - Area and page basics
Section 2 - Block and inline basics

Purpose:

- an inline-level placeholder replaced with the page number of the first normal area of cited formatting object

Content:

- (6.6.11) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other required property:
ref-id=(7.30.13;491)

Other optional properties:
alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
dominant-baseline=(7.14.5;468)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
score-spaces=(7.30.15;495)
text-altitude=(7.29.4;500)
text-decoration=(7.17.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
text-transform=(7.17.6;501)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:
font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Property of interest:

- ref-id= must point to an area with an identifier on the desired page

Of note:

- the formatter supplies to the flow the characters as <character> objects
  - the characters used by the formatter are specified by the format= property of the <page-sequence> for the cited page

Copyright © Crane Softwrights Ltd.

<page-number-citation-last> Object

Purpose:

- an inline-level placeholder replaced with the page number of the last normal area of cited formatting object, consistent with the indicated citation strategy

Content:

- (6.6.12) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other required property:
ref-id=(7.30.13;491)

Other optional properties:
alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
dominant-baseline=(7.14.5;468)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
page-citation-strategy=(7.30.10;488)
score-spaces=(7.30.15;495)
text-altitude=(7.29.4;500)
text-decoration=(7.17.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
text-transform=(7.17.6;501)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:
font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Properties of interest:

- ref-id= must point to an area with an identifier on the desired page
- page-citation-strategy= is either "all", "normal" or "non-blank"

Of note:

- the formatter supplies to the flow the characters as <character> objects
  - the characters used by the formatter are specified by the format= property of the <page-sequence> for the cited page

 Copyright © Crane Softwrights Ltd.

Simple page layout definition

Every different page layout needed to be rendered must be described separately

- `<simple-page-master>`
  - must be named using the `master-name=` property
- differences in reference orientations of regions
- differences in column progression directions
  - changes from defaults are specified by explicitly setting the `writing-mode=` property
- differences in geometry
  - page dimensions
  - region dimensions

A page's page-viewport-area's content-rectangle is the page dimensions

- the page's page-reference-area's content-rectangle is within page margin boundaries
- the body region's region-viewport-area is the page-reference-area's content-rectangle

Copyright © Crane Softwrights Ltd.

Simple page layout definition (cont.)

Every page layout has at least one "middle" body region (see page 79)

- `<region-body/>`
  - may be named differently than the default "`xsl-region-body`" name
    - typically explicitly named to distinguish regions in sequenced pages
    - where the same name is used on more than one page master, all uses of the name must be for the same kind of region
  - defines the main-reference-area that accepts paginated content
    - the main-reference-area will shrink in the presence of before-floats and footnotes
  - 1.0 limited to only one body region in XSL-FO 1.0
  - 1.1 unlimited number of body regions in XSL-FO 1.1
- distinct from optional perimeter regions in their definition but not in the page areas they occupy

Perimeter regions cut into the body region

- may be named differently than the default names by using `region-name=`
  - renaming regions promotes good maintenance practice
  - `<region-before/>` named "`xsl-region-before`"
  - `<region-after/>` named "`xsl-region-after`"
  - `<region-start/>` named "`xsl-region-start`"
  - `<region-end/>` named "`xsl-region-end`"
- if the body region doesn't have margins then the perimeter region content will overlap the body content
  - perimeter regions do not automatically shrink the size of the `content-rectangle` of the body region
- before and after regions can indicate they have precedence over the start and end regions

Regions may have different reference orientations and writing modes

- common need to have a landscape body region orientation inside of a portrait page geometry with portrait before and after regions

Important region name constraints on multiple page masters

- once a given custom name is used, it must be used for the same region position wherever else it used on other page masters
- the reserved names cannot be used regions other than what is already used by the Recommendation
- e.g. the region name "`myheader`" when used for a `region-before` can only be used for a `region-before` in other page masters

 Copyright © Crane Softwrights Ltd.

Spans and columns in simple page geometry

Hierarchy of reference areas within includes spans and columns

- each body region contains one main-reference-area
  - area occupies entire region width
  - area sits between before-float-reference-area and footnote-reference-area
- main-reference-area contains span-reference-area areas
  - groups of columns with the same number of columns spanned
  - groups are stacked in the block-progression-direction
- span-reference-area contains normal-flow-reference-areas (columns)
  - the normally-flowed blocks of information in the flow are flowed into the columns
  - columns are stacked in the inline-progression-direction

Only equal-width evenly-spaced columns are allowed

- only column count and column gap can be specified
- fixed column specification for entire page
  - cannot change column-count within a page other than spanning all columns

Only main-reference-area-wide spans are allowed

- a block can span all columns or only one column
- a spanned block must be at the top of the flow (an immediate child of `<flow>`)

Columns are balanced within each span-reference-area

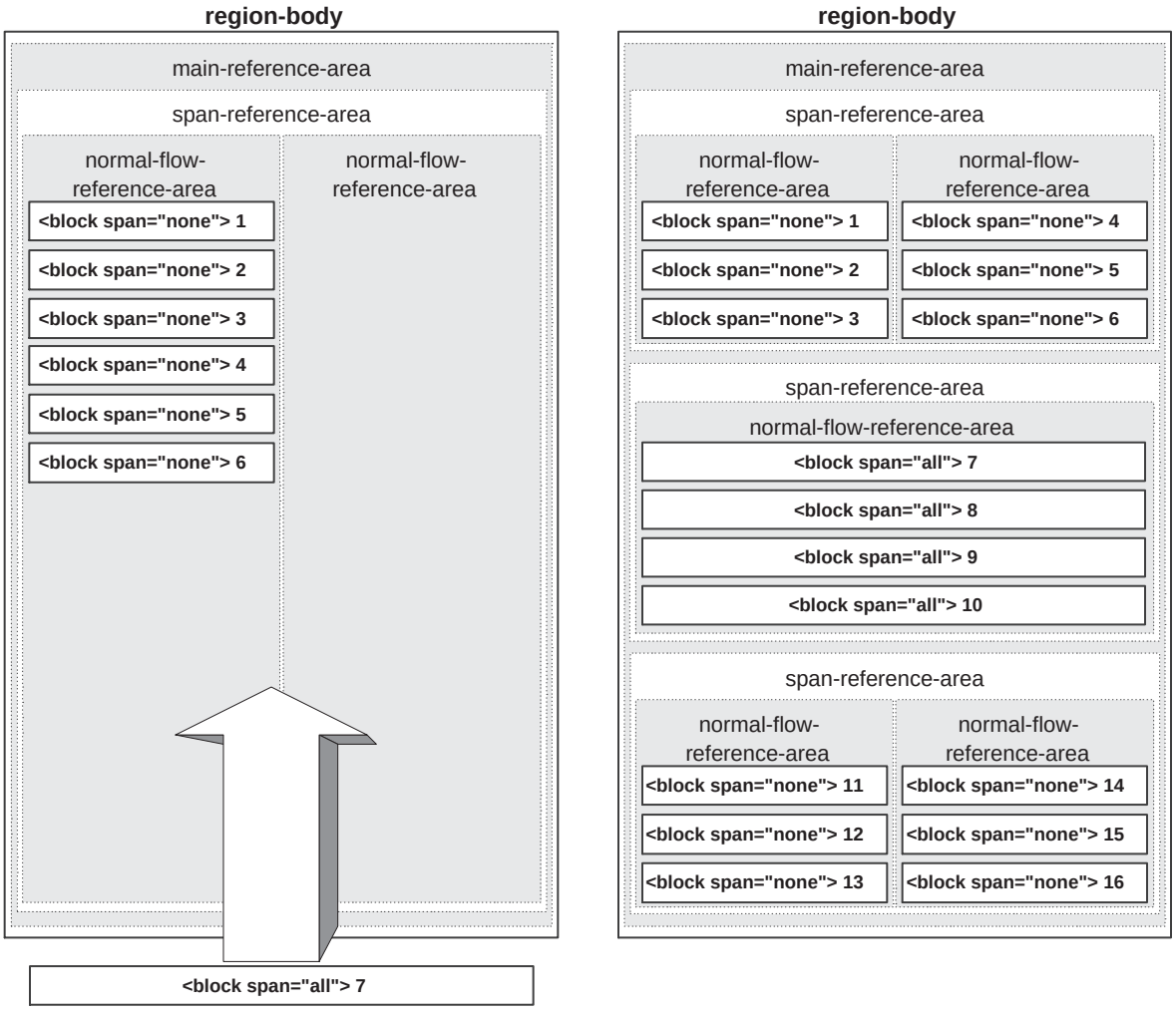
- a given span-reference-area accepts blocks in its normal-flow-reference-area areas until the number of columns spanned by a block changes
  - a new span-reference-area is then introduced on the page
  - all blocks flowed in the interrupted span-reference-area are re-flowed across all columns
  - the balancing happens not based on the count of blocks but on the content of each block and the contained lines
- all blocks flowed into the last span-reference-area of the main-reference-area are balanced across all columns

Copyright © Crane Softwrights Ltd.

Spans and columns in simple page geometry (cont.)

Consider a before/after situation of a formatter flowing a mixture of spanned and un-spanned blocks:

- six un-spanned blocks flow into a normal-flow-reference-area as normal blocks
- a seventh block spans all columns and triggers the generation of a new span-reference-area
- the content of the first six blocks is flowed across all normal-flow-reference-area areas of the first span-reference-area based on the length of the content in the sum total of the six blocks
    - the count of blocks is irrelevant
    - the total length of all blocks is balanced across the columns while accommodating keeps, widows and orphans



 Copyright © Crane Softwrights Ltd.

`<simple-page-master>` Object

Purpose:

- the specification of a given page's physical geometry

Content:

- 1 (6.4.12) (`region-body`,`region-before`?,`region-after`?,`region-start`?, `region-end`?)
- 1+ (6.4.13) (`region-body`+,`region-before`?,`region-after`?,`region-start`?, `region-end`?)
- Child objects (alphabetical):
  - `<region-after>`(6.4.16;239)
  - `<region-before>`(6.4.15;238)
  - `<region-body>`(6.4.14;121)
  - `<region-end>`(6.4.18;242)
  - `<region-start>`(6.4.17;241)
- Referring object:
  - `<layout-master-set>`(6.4.7;63)

Property sets:

- Common Margin Properties-Block(7.11;432)

Other required property:
`master-name=`(7.27.8;481)

Other optional properties:
`page-height=`(7.27.13;488)
`page-width=`(7.27.15;488)
`reference-orientation=`(7.21.3;491)
`writing-mode=`(7.29.7;504)

Shorthand influencing the above properties:
`size=`(7.31.21;495)

Copyright © Crane Softwrights Ltd.

<simple-page-master> Object (cont.)

The XSL-FO page description 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>
```


Copyright © Crane Softwrights Ltd.

<region-body> Object

Purpose:

- the definition of the middle area inside all perimeter regions defined for the page

Content:

- (6.4.14) EMPTY
- Referring object:
  - <simple-page-master>(6.4.13;119)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)

Other required property:
region-name=(7.27.17;491)

Other optional properties:
clip=(7.21.1;465)
column-count=(7.27.2;466)
column-gap=(7.27.3;466)
display-align=(7.14.4;468)
overflow=(7.21.2;484)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Properties of note:

- the region-name= property is required but the default name of "xsl-region-body" is used as this required property if a name is not supplied in the XSL-FO instance
- even though padding= and border-width= properties are indicated as available indirectly through the common property set, these values are fixed at "0pt" in XSL-FO 1.0
- display-align= is used to keep the information in the region snug against the before edge, snug against the after edge, or centered in the middle of the two in the block progression direction

The placement of this region is illustrated in Page geometry (page 79)

Copyright © Crane Softwrights Ltd.

<region-body> Object (cont.)

The page body description element:

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="http://www.w3.org/1999/XSL/Format" font-size="16pt">
   <layout-master-set>
      <simple-page-master master-name="bookpage"
                          page-height="297mm" page-width="210mm"
                          margin-top="15mm" margin-bottom="15mm"
                          margin-left="15mm" margin-right="15mm">
         <region-body region-name="bookpage-body"
                      margin-top="5mm" margin-bottom="5mm"/>
      </simple-page-master>
   </layout-master-set>
   <page-sequence master-reference="bookpage">
      <title>Hello world example</title>
      <flow flow-name="bookpage-body">
         <block>Hello XSL-FO!</block>
      </flow>
   </page-sequence>
</root>
```

 Copyright © Crane Softwrights Ltd.

Page sequence titling

A sequence of pages can be assigned a title

- presented to the user by the formatter in an implementation dependent fashion
  - the formatter may choose to render the title on the result canvas
    - once per page sequence
    - once per page
  - an interactive user agents could display in the title bar of a screen window
  - the title may be suppressed altogether by the formatter

Portability issue with respect to development

- a stylesheet writer may be using a formatter that presents the title information outside of the page canvas
- the user of a stylesheet may be using a formatter that presents the title information on the page canvas
  - the result may not be as desired by the stylesheet writer

Copyright © Crane Softwrights Ltd.

<title> Object

Purpose:

- a page sequence's ancillary description not rendered on the page canvas

Content:

- (6.4.21) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - <page-sequence>(6.4.5;65)
- must not have a marker or out-of-line descendant:
  - <marker>
  - <float>
  - <footnote>
  - <block-container> with an absolute position specification

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)

Other optional properties:

color=(7.18.1;465) visibility=(7.30.17;502)
line-height=(7.16.4;479)

Shorthand influencing the above properties:

font=(7.31.13;470)

 Copyright © Crane Softwrights Ltd.

<title> Object (cont.)

The XSL-FO page body description 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>
```


Copyright © Crane Softwrights Ltd.

Chapter 5 - Generic body constructs

- Introduction - Often-used formatting constructs
- Section 1 - Lists
- Section 2 - Graphics and foreign objects
- Section 3 - Links
- Section 4 - Leaders

Outcomes:

- understand the use of list constructs
- understand graphic and foreign object concepts
- understand link concepts
- understand the use of leader constructs

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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

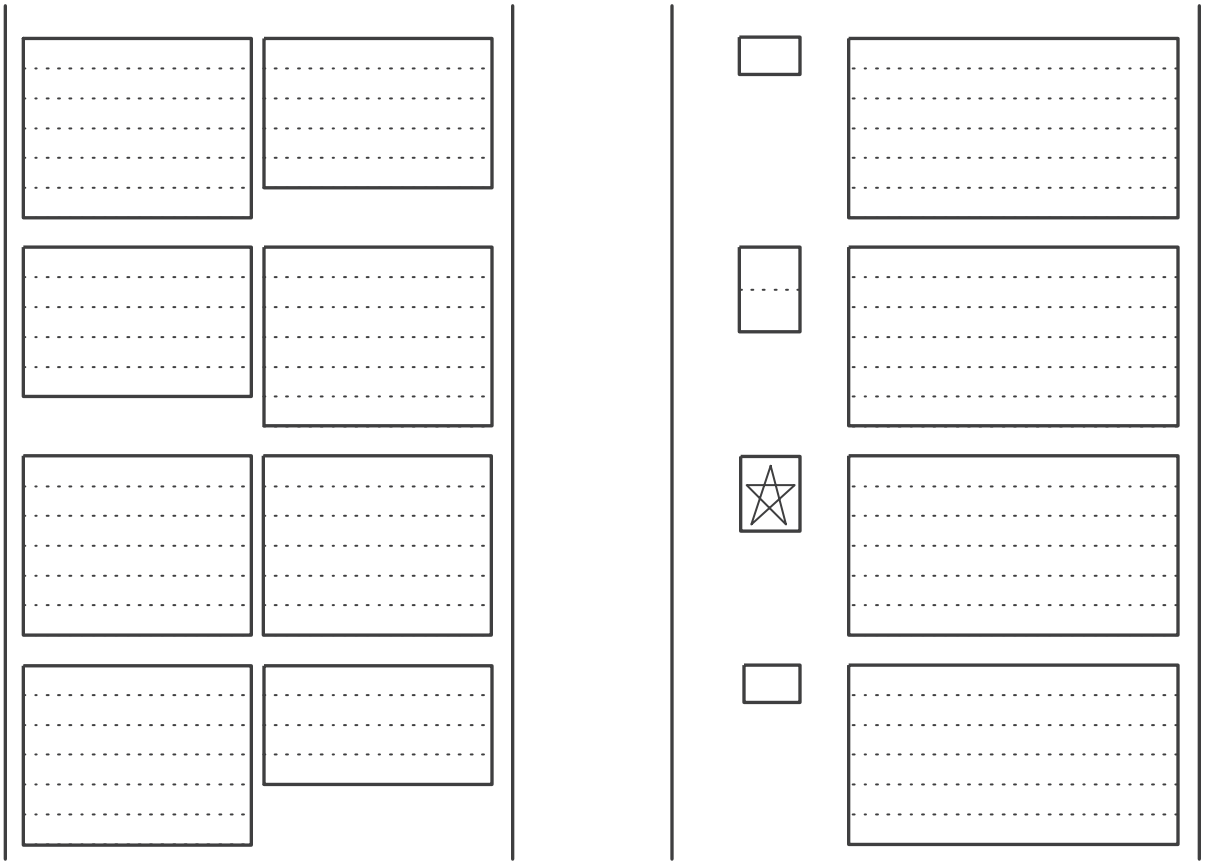
 Copyright © Crane Softwrights Ltd.

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.



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

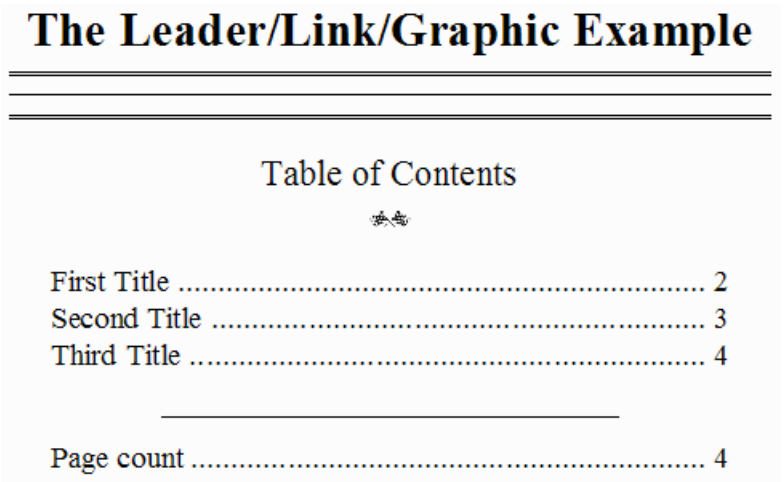
Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Often-used formatting constructs (cont.)

Chapter 5 - Generic body constructs

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



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

Copyright © Crane Softwrights Ltd.

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
- 1.1 `<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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Aligned pairs of block-level constructs

Standalone blocks cannot be adjacent to each other in the inline-progression direction

- a block-level construct typically restarts in the block-progression direction as a following sibling

Pairs of adjacent blocks are useful to associate the information in one of the pair to the information in the other of the pair

- side-by-side polyglot (multi-language) text
    - e.g. aligned translation paragraphs
- terms and their definitions
- generic list item formatting
    - the item enumerator and the item content

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

- don't consider that only lists in your XML information can use this construct for layout
- regard the construct as a layout facility for two block-level constructs in the inline-progression direction of the parent block
- a two-column table construct may suffice or offer other layout features you need

Specialized properties are available for behaviors characteristic of list processing

- provisional property values can be used to make the calculated property values of the two blocks relative to each other instead of independent of each other
- no obligation to use the specialized properties

The content of the blocks does not impact on the start/end edges of the blocks

- the start and end edges of each block are set absolutely or relative to each other, but not relative to the length of the content
- when using provisional values and functions, the edges are fixed to the same values for all members of the entire list
    - planning ahead is necessary to ensure any undesired wrapping within edges is avoided
- when specifying margins on a per list item basis, the edges can be explicitly specified differently than other list items
    - cannot measure the content of a label or body in order to set the margins relative to the content

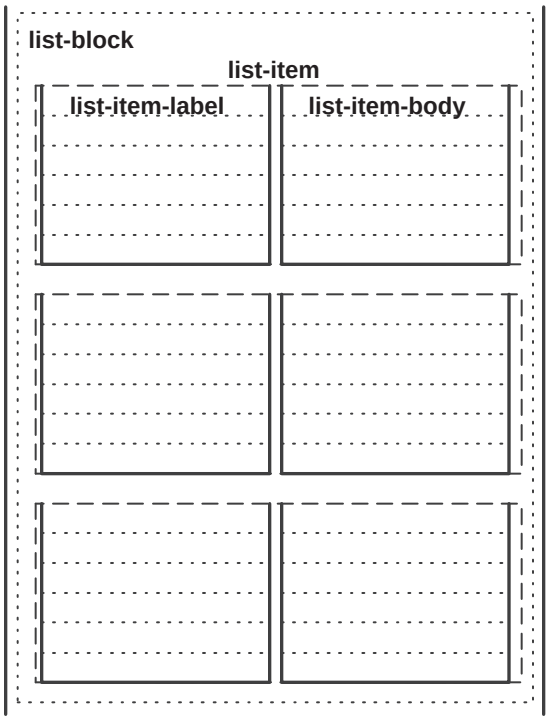
 Copyright © Crane Softwrights Ltd.

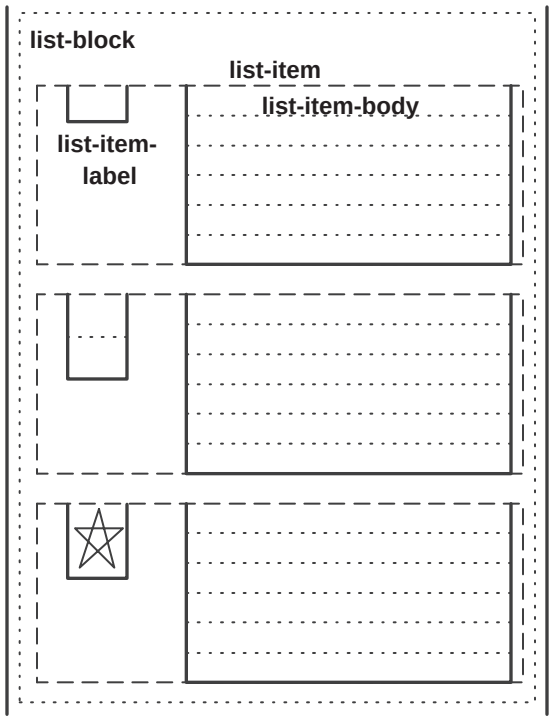
List constructs

Chapter 5 - Generic body constructs
Section 1 - Lists

A list is a block-level object:

- it contains only list-items (pairs)
  - it requires a nested list to itself be a list item unless it is already in a list item
- each item has the two components of the pair:
  - the list item label (the member of the pair on the start side)
  - the list item body (the member of the pair on the end side)





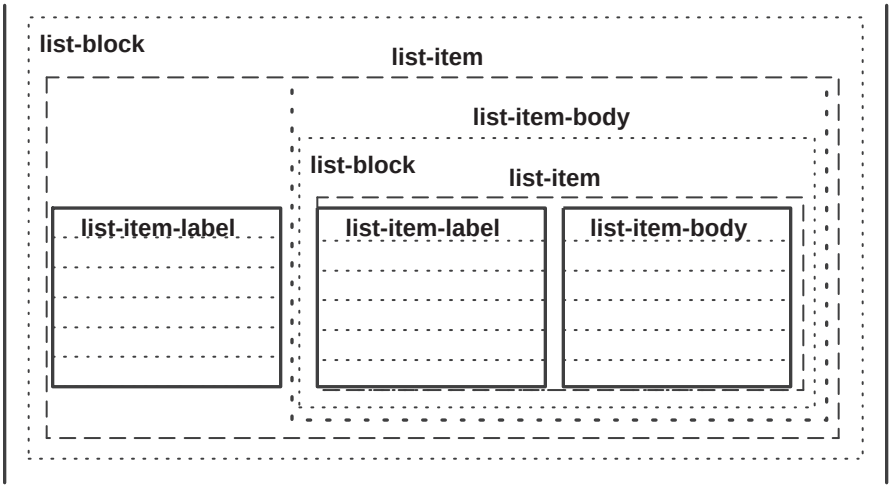
Copyright © Crane Softwrights Ltd.

List constructs (cont.)

Chapter 5 - Generic body constructs
Section 1 - Lists

Block alignment can involve more than two blocks:

- the body of a list item can itself contain a list with two aligned blocks
- the initial value of relative alignment is the before edge for all blocks



One could consider using a table for more than two blocks

- table properties allow proportional sizes of the block widths

Copyright © Crane Softwrights Ltd.

List constructs (cont.)

Vertical alignment of list item label defaults to a common before edge

- ensures the before edge of the first block on each side is aligned on the page
- `relative-align=` can be set to common text baseline of first lines
  - when different font sizes are used without aligning to the text baseline, the smaller of the two font sizes will appear to be superscripted

Horizontal indentation of the list item components is the responsibility of the stylesheet writer:

- the item label's `start-indent=` is specified explicitly (or inherited)
- the item body's `end-indent=` is specified explicitly (or inherited)
- the item label's `end-indent=` and item body's `start-indent=` can be independently set and may be different for each item
  - will result in inconsistent presentation of the labels in parent areas of different widths
    - e.g. in differing media (US-letter and A4 page sizes have different widths)
  - without proper planning can inadvertently result in display problems
    - overlapping label content on top of body content (an error)
    - undesired wrapping of the label content
- the item label's `end-indent=` and item body's `start-indent=` can be set relative to the `start-indent=` of the item label and is the same for each item
  - guarantees consistent label presentation in parent areas of different widths
    - e.g. changing the paper width will not change the presentation of the labels
  - the stylesheet can ask the processor to dynamically set these values on a per list item basis
  - protects the content of the two areas from sharing the same place on the rendered output

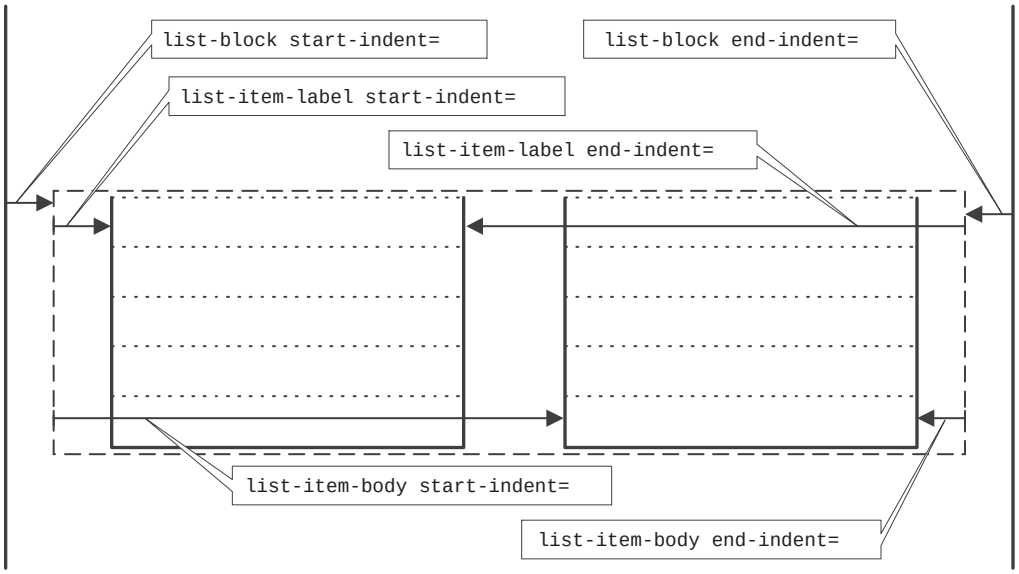
Copyright © Crane Softwrights Ltd.

List constructs (cont.)

Chapter 5 - Generic body constructs
Section 1 - Lists

Not obliged to use `body-start()` or `label-end()`:

- can specify the distances from the parent edges explicitly for each of the label and body



Risk of overlap is borne by the stylesheet writer

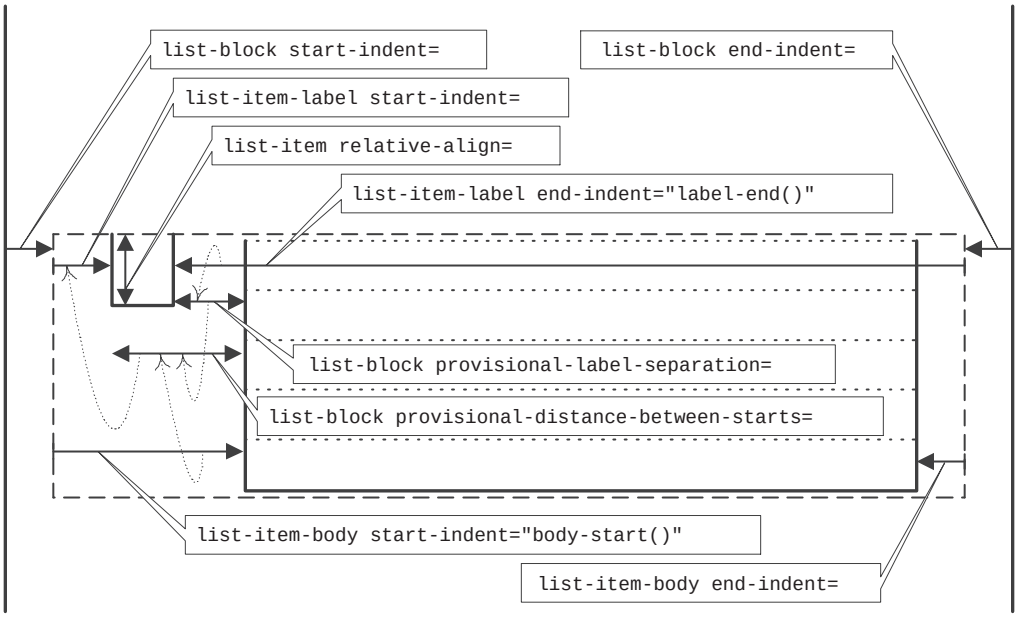
- a common error when starting a stylesheet is to forget to set any margins and to witness both blocks of content superimposed within the parent area
- another common error after writing a stylesheet is to change the page width and not think to change the list item member indents
  - e.g. widening the page dimensions from A4 to US-letter
  - the blocks could overlap in the middle when the distances from the edges are used
  - XSL-FO 1.0 states the overlap is an error, so this cannot be relied upon as an effect

Copyright © Crane Softwrights Ltd.

List constructs (cont.)

Chapter 5 - Generic body constructs
Section 1 - Lists

Using `body-start()` and `label-end()` will prevent edges from overlapping and keep label area formatting consistent:



Relative indentation of the list item components:

- processor notes `provisional-distance-between-starts=` and `provisional-label-separation=` values for built-in functions
  - "provisional" indicates the value is used as part of an arithmetic calculation
  - values are properties of the list and cannot be set on an individual item
  - specifying a separation ensures to overlap of the label and body
- two built-in functions reflect the above two values in the context of the item label's `start-indent=` and can (should) be used to protect the adjacent area boundaries from each other and ensure consistent label area formatting for different parent widths:
  - `label-end()`
    - the distance from the end of the label to the end of the line accommodating the indent of the label
    - used as the value for the `end-indent=` property of the item label
  - `body-start()`
    - the distance from the start of the line to the start of the body accommodating the indent of the item label
    - used as the `start-indent=` property of the item body

Copyright © Crane Softwrights Ltd.

Nested list constructs

The nesting of lists may require the use of hidden list-item members

- an XSL-FO list can only contain list items (see page 133)
- a nested list in user's XML vocabulary is usually one of either:
  - a block within an existing list item
    - e.g. `list/listitem/list`
    - the block-level `<list-block>` formatting object can flow with other blocks inside a `<list-item-body>`
  - its own list item
    - e.g. `list/list`
    - a `list-block` can only contain `list-item` children
      - thus requiring a hidden `list-item` within which the list block is placed in the body
    - the hidden `list-item` children must have blocks for labels
      - the label blocks can be empty to hide the fact they are list items

Determining the need depends on:

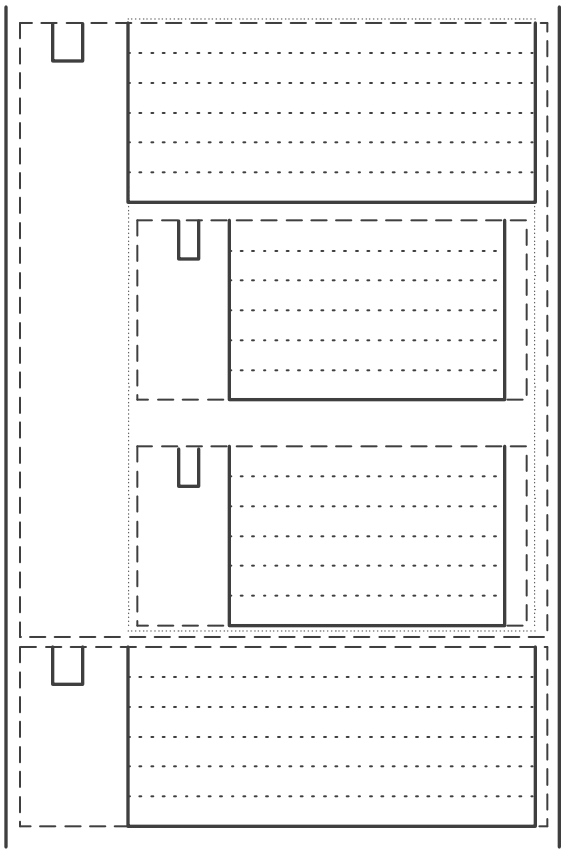
- for dependence of the indentation of the nested list
  - is its indentation fixed?
    - use the body of a hidden list item with a fixed indentation for the body
  - is its indentation relative to the indentation of the label of the list item in which it is flowed?
    - could use the calculated body start indentation of the preceding list item
    - could flow the nested list at the end of the preceding list item's body
- the nature of the source list
  - (typical) is the nested list inside one of the list items? (e.g. HTML, DocBook)
    -
```
<list>
 <item>
  <p>...</p>
  <list>nested list</list>
 </item>
 <item>...
```
  - (atypical) is the nested list a sibling of other list items?
    -
```
<list>
 <item>
  <p>...</p>
 </item>
 <list>nested list</list>
 <item>...
```

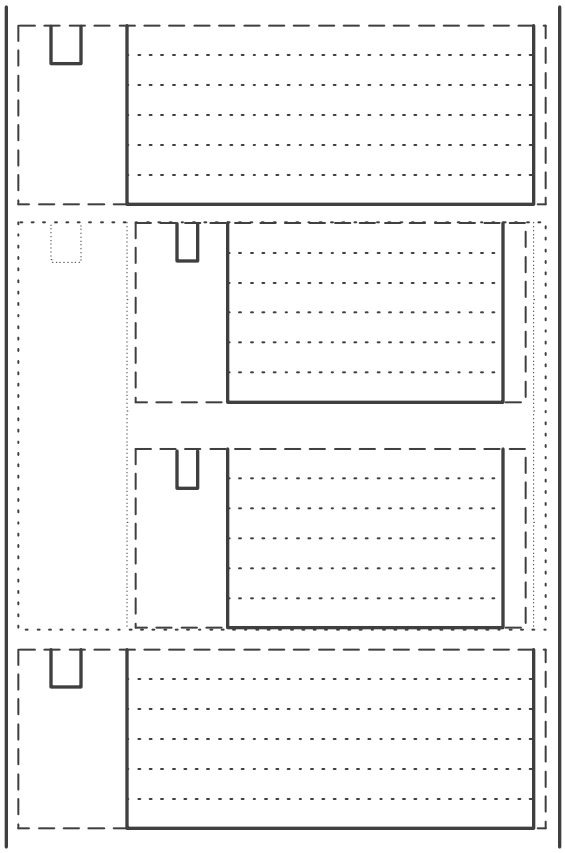
 Copyright © Crane Softwrights Ltd.

Nested list constructs (cont.)

Chapter 5 - Generic body constructs
Section 1 - Lists

The use of a ghost list item prevents violating the content model rules for the containing list





Cannot use just a `<block>` or `<list-block>`

- a `<list-block>` can only contain `<list-item>` children
- the ghost list item contains the nested list in the body of the item
    - requires an empty block in the label to not be visible on the page

Copyright © Crane Softwrights Ltd.

<list-block> Object

Chapter 5 - Generic body constructs
Section 1 - Lists

Purpose:

- the parent object of a related set of pairs of synchronized block-level areas laid out in the inline-progression direction

Content:

- (6.8.2) (list-item+)
- Child object:
  - <list-item>(6.8.3;142)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)
- Common Relative Position Properties(7.13;434)

Other optional properties:

break-after=(7.20.1;463)
break-before=(7.20.2;463)
clear=(7.19.1;465)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
provisional-distance-between-starts=(7.30.12;490)
provisional-label-separation=(7.30.11;491)

Shorthands influencing the above properties:

page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

Properties of interest:

- provisional-distance-between-starts= specifies the desired distance between the start of the list item label and the start of the list item body
- provisional-label-separation= specifies the desired distance between the end of the list item label and the start of the list item body

Copyright © Crane Softwrights Ltd.

<list-block> Object (cont.)

A <list-block> construct from the earlier A detailed example of flowed content (page 41):

```
<block space-before="6pt" font-size="14pt">
  This page's material as an instructor-led handout:</block>
<list-block provisional-distance-between-starts=".43in"
            provisional-label-separation=".1in"
            space-before="6pt">
 <list-item relative-align="baseline">
  <list-item-label text-align="end" end-indent="label-end()">
   <block>-</block>
  </list-item-label>
  <list-item-body start-indent="body-start()">
   <block font-size="14pt">excerpts of formatting objects created
          through the use of an XSLT stylesheet</block>
  </list-item-body>
 </list-item>
</list-block>
```

Separation between the label and body is specified

- the label is .33 inches wide and .1 inches away from the body

Copyright © Crane Softwrights Ltd.

`<list-item>` Object

Chapter 5 - Generic body constructs
Section 1 - Lists

Purpose:

- the parent of a single pair of synchronized block-level areas laid out in the inline-progression direction
  - indents are relative to containing `<list-block>` object

Content:

- (6.8.3) (`list-item-label`,`list-item-body`)
- Child objects (alphabetical):
  - `<list-item-body>`(6.8.4;146)
  - `<list-item-label>`(6.8.5;144)
- Referring object:
  - `<list-block>`(6.8.2;140)
- may begin with any number of `<marker>` children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)
- Common Relative Position Properties(7.13;434)

Other optional properties:

`break-after=`(7.20.1;463)
`break-before=`(7.20.2;463)
`id=`(7.30.8;475)
`index-class=`(7.24.1;475)
`index-key=`(7.24.2;475)
`intrusion-displace=`(7.19.3;477)
`keep-together=`(7.20.3;477)
`keep-with-next=`(7.20.4;477)
`keep-with-previous=`(7.20.5;477)
`relative-align=`(7.14.6;492)

Shorthands influencing the above properties:

`page-break-after=`(7.31.16;487)
`page-break-before=`(7.31.17;487)
`page-break-inside=`(7.31.18;487)

Property of interest:

- `relative-align=` determines the alignment of the list item with either the before edge of the list body (default) or the baseline of the first line area generated by the list body
  - the value "`baseline`" is typically used when both sides are using text
  - the value "`before`" is typically used when one or both sides are graphics

 Copyright © Crane Softwrights Ltd.

<list-item> Object (cont.)

A <list-item> construct from the earlier A detailed example of flowed content (page 41):

```
<block space-before="6pt" font-size="14pt">
  This page's material as an instructor-led handout:</block>
<list-block provisional-distance-between-starts=".43in"
            provisional-label-separation=".1in"
            space-before="6pt">
 <list-item relative-align="baseline">
  <list-item-label text-align="end" end-indent="label-end()">
   <block>-</block>
  </list-item-label>
  <list-item-body start-indent="body-start()">
   <block font-size="14pt">excerpts of formatting objects created
          through the use of an XSLT stylesheet</block>
  </list-item-body>
 </list-item>
</list-block>
```


Copyright © Crane Softwrights Ltd.

`<list-item-label>` Object

Chapter 5 - Generic body constructs
Section 1 - Lists

Purpose:

- the start-side member of a pair of synchronized block-level areas laid out in the inline-progression direction

Content:

- (6.8.5) `(%block;)+`
- Child object:
  - `%block;`(6.2;71)
- Referring object:
  - `<list-item>`(6.8.3;142)
- may begin with any number of `<marker>` children

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

`id=`(7.30.8;475)
`index-class=`(7.24.1;475)
`index-key=`(7.24.2;475)
`keep-together=`(7.20.3;477)

Shorthand influencing the above properties:

`page-break-inside=`(7.31.18;487)

Property of interest:

- `end-indent=` can utilize the `label-end()` function

Copyright © Crane Softwrights Ltd.

`<list-item-label>` Object (cont.)

A `<list-item-label>` construct from the earlier A detailed example of flowed content (page 41):

```
<block space-before="6pt" font-size="14pt">
  This page's material as an instructor-led handout:</block>
<list-block provisional-distance-between-starts=".43in"
            provisional-label-separation=".1in"
            space-before="6pt">
 <list-item relative-align="baseline">
  <list-item-label text-align="end" end-indent="label-end()">
   <block>-</block>
  </list-item-label>
  <list-item-body start-indent="body-start()">
   <block font-size="14pt">excerpts of formatting objects created
          through the use of an XSLT stylesheet</block>
  </list-item-body>
 </list-item>
</list-block>
```

Of note:

- the end alignment is specified on the label to move the content of the label as close as possible to the body
- this supports the appearance of variable length labels (i.e. Roman characters) with a common distance between the label and the body

Copyright © Crane Softwrights Ltd.

<list-item-body> Object

Chapter 5 - Generic body constructs
Section 1 - Lists

Purpose:

- the end-side member of a pair of synchronized block-level areas laid out in the inline-progression direction

Content:

- (6.8.4) (%block;)+
- Child object:
  - %block;(6.2;71)
- Referring object:
  - <list-item>(6.8.3;142)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-together=(7.20.3;477)

Shorthand influencing the above properties:

page-break-inside=(7.31.18;487)

Property of interest:

- start-indent= can utilize the body-start() function

Copyright © Crane Softwrights Ltd.

<list-item-body> Object (cont.)

A <list-item-body> construct from the earlier A detailed example of flowed content (page 41):

```
<block space-before="6pt" font-size="14pt">
  This page's material as an instructor-led handout:</block>
<list-block provisional-distance-between-starts=".43in"
            provisional-label-separation=".1in"
            space-before="6pt">
 <list-item relative-align="baseline">
  <list-item-label text-align="end" end-indent="label-end()">
   <block>-</block>
  </list-item-label>
  <list-item-body start-indent="body-start()">
   <block font-size="14pt">excerpts of formatting objects created
          through the use of an XSLT stylesheet</block>
  </list-item-body>
 </list-item>
</list-block>
```

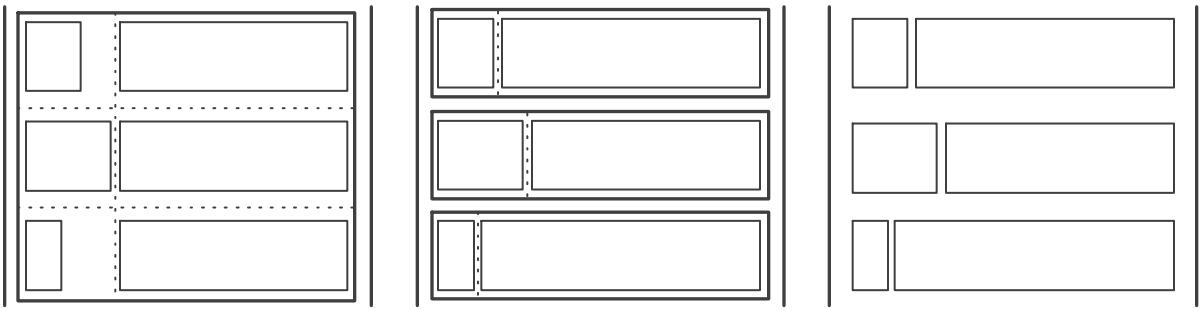

Copyright © Crane Softwrights Ltd.

When is a list not a list?

Some formatting requirements for lists cannot be satisfied with the `<list-block>` construct when the margins of the list items need to reflect the contents of the list

- `<list-block>` forces the stylesheet writer to specify the start and end edges of the labels and bodies of the items, which cannot be based on the length of the content
- one is not obliged to use `<list-block>` to format a list-like source construct

Consider three alternative approaches to satisfy the need to format three list items where the item labels are distinctly different lengths, none of which is known by the stylesheet writer



Basing the indentation of all item bodies on the longest length of all item labels

- can use one table for entire list contents (e.g. left page image)
  - hidden borders and auto-calculated column widths with all rows reflecting a single set of column widths
  - use one row for each list item with auto width calculation of the first column determining longest label
    - may not work well if the label has spaces that will promote cell balancing

Basing the indentation of each item body on the length of corresponding item labels

- can use a one-row table for each list item (e.g. center page image)
  - hidden borders and auto-calculated column widths for each individual row in separate table
  - use as many tables as there are list items
- can use a side-float aligned with the block (e.g. right page image)
  - each body block is prefaced with a side float block with the label whose size squeezes the body block width

 Copyright © Crane Softwrights Ltd.

Non-textual information

Graphic and other non-textual constructs can come from two possible sources

- external files
  - `<external-graphic/>`
  - a URI specifies how to obtain the external resource
- embedded use of non-XSL-FO namespace constructs
  - `<instream-foreign-object>`
  - the use of namespaces ensures no confusion with XSL-FO objects
  - e.g. Scalable Vector Graphics (SVG), Math Markup Language (MathML), etc.
  - promotes the generation of images from the stylesheets that generate the XSL-FO
    - e.g. charting numbers from the XML instance into SVG constructs

XSL-FO processor need not recognize the formats, only pass on the content from whatever source to rendering agent for imaging

- the formatter is only responsible for measuring and allocating the required area on the page as indicated by the properties
- rendering agent must recognize the file type by either name or content
  - no notation information communicated through the stylesheet from the DTD

Content need not be a static graphic image

- could be a window into an application user interface
  - e.g. spreadsheet

Copyright © Crane Softwrights Ltd.

Non-textual information (cont.)

Scaling is necessary when the image intrinsic size is larger than the page and you need it to fit

- `content-width="scale-to-fit" width="100%"`
- could otherwise choose to base scaling on the height

When smaller images need to stay small but larger images need to scale

- rely on the engine's on-the-fly choice of the smaller of two width sizes:
  ```
  scaling="uniform"
  width="100%" content-width="scale-to-fit"
  content-height="100%"
  ```
- or rely on the engine's on-the-fly choice of the smaller of two height sizes:
  ```
  scaling="uniform"
  height="100%" content-height="scale-to-fit"
  content-width="100%"
  ```
- the `scaling="uniform"` ensures the resulting aspect ratio is not distorted
- need to choose one of the two dimensions as a governing dimension for the calculations, for example, by choosing width:
  - the `content-width="scale-to-fit"` will scale the image to the specified viewport `width="100%"` of the available width
  - the `content-height="100%"` ensures that the image width is not enlarged to 100% of the viewport when the width is smaller
- the similar interpretation would work in the other dimensions when choosing height as the governing dimension

 Copyright © Crane Softwrights Ltd.

Non-textual information (cont.)

The `src=` property has specific syntactic requirements

- safest to use: `url("`*`uri-value-here`*`")`
  - the author of the XML document may be creating the URI being used in the resulting syntax
  - the URI character set includes the single quote that would bring about an improperly formed attribute value
- could use no quote delimiters, but not if content has a single quote
- could use single quote delimiters, but not if content has a single quote
- ensure XML syntax for attribute use of quotes is acceptable:
  - e.g. `src='url("`*`uri-value-here`*`")'`
  - e.g. `src="url("`*`uri-value-here`*`")"`
- this is not a function call but a syntactic convention
  - without it the URI syntax could ambiguously be interpreted as expression syntax
  - without it any embedded single quotes (which are allowed in URI syntax and could be in the source XML without the stylesheet writer knowing) could confuse the interpretation of the literal string

Copyright © Crane Softwrights Ltd.

<external-graphic> Object

Chapter 5 - Generic body constructs
Section 2 - Graphics and foreign objects

Purpose:

- the inline-level display of graphical or other externally-supplied information
  - displayed matter does not reside as part of XSL-FO instance content
    - needs reference to an addressed external resource
  - can be placed in a <block> object for block-level display

Content:

- (6.6.5) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other required property:

src=(7.30.16;498)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
allowed-height-scale=(7.15.1;449)
allowed-width-scale=(7.15.2;449)
baseline-shift=(7.14.3;453)
block-progression-dimension=(7.15.3;453)
clip=(7.21.1;465)
content-height=(7.15.4;466)
content-type=(7.30.7;466)
content-width=(7.15.5;466)
display-align=(7.14.4;468)
dominant-baseline=(7.14.5;468)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
line-height=(7.16.4;479)
overflow=(7.21.2;484)
scaling=(7.15.12;494)
scaling-method=(7.15.13;494)
text-align=(7.16.9;500)
width=(7.15.14;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Copyright © Crane Softwrights Ltd.

Properties of interest:

- `text-align=` and `display-align=` are used to align the scaled image's reference area within the viewport area
  - ancestral specifications will align the viewport area within its parent area
  - `width=` and `height=` specify the viewport (size of the graphic window), while `content-width=`, `content-height=` and `scaling=` specify the size of the image, using `overflow=` to hide the image outside the viewport
- `alignment-baseline=` is used to move the image after-edge off the text baseline

Copyright © Crane Softwrights Ltd.

<external-graphic> Object (cont.)

An excerpt from the samp/leadlink.fo sample (page 130):

```
<block font-size="24pt" space-after="16pt" space-before="1cm"
       text-align="center">
       Table    of      Contents
       <block/>
       <external-graphic src="url(&quot;smflags.bmp&quot;)"/>
 </block>
```

Note the following regarding the above structure

- by default, white-space is collapsed so that the words "Table of Contents" are flowed on the page as if separated by only one space each
- the use of the empty block to introduce a line of zero size thus breaking the title and the graphic onto separate lines
- the use of the protected quoted syntax for the URI of the graphic image
- the centering of all inline constructs on all lines of the outside block using text-align=

 Copyright © Crane Softwrights Ltd.

<instream-foreign-object> Object

Purpose:

- the inline display of graphical or other instance-supplied information
  - displayed matter resides as part of the XSL-FO instance content
    - descendent content of object using a non-XSL-FO namespace
  - object can be placed in a <block> object for block-level display

Content(6.6.6):

- a single child element from a non-XSL-FO namespace

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
¦ allowed-height-scale=(7.15.1;449)
¦ allowed-width-scale=(7.15.2;449)
baseline-shift=(7.14.3;453)
block-progression-dimension=(7.15.3;453)
clip=(7.21.1;465)
content-height=(7.15.4;466)
content-type=(7.30.7;466)
content-width=(7.15.5;466)
display-align=(7.14.4;468)
dominant-baseline=(7.14.5;468)
height=(7.15.6;474)
id=(7.30.8;475)
¦ index-class=(7.24.1;475)
¦ index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
line-height=(7.16.4;479)
overflow=(7.21.2;484)
scaling=(7.15.12;494)
scaling-method=(7.15.13;494)
text-align=(7.16.9;500)
width=(7.15.14;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Properties of interest:

- text-align= and display-align= are used to align the scaled image's reference area within the viewport area
  - ancestral specifications will align the viewport area within its parent area
  - width= and height= specify the viewport (size of the graphic window), while content-width=, content-height= and scaling= specify the size of the image, using overflow= to hide the image outside the viewport
- alignment-baseline= is used to move the image after-edge off the text baseline

Copyright © Crane Softwrights Ltd.

`<instream-foreign-object>` Object (cont.)

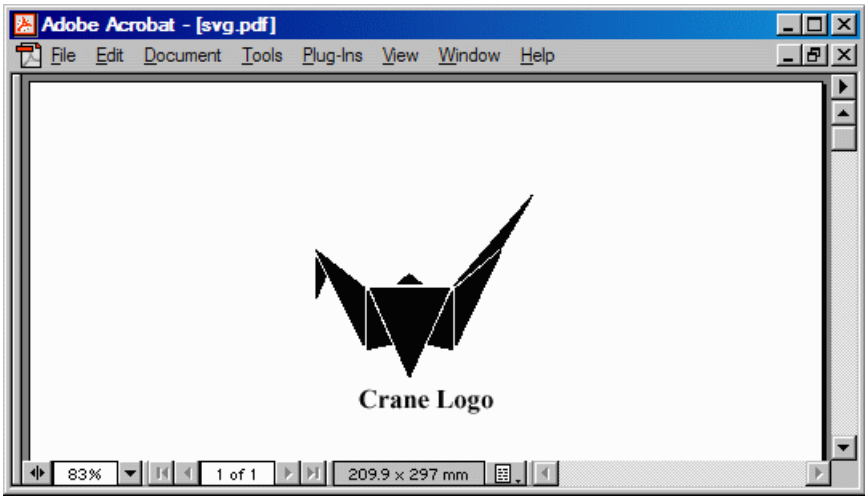
An example of embedding an alternate namespace (in this case, Scalable Vector Graphics (SVG)) in XSL-FO instances:

```
<block text-align="center">
  <block>
   <instream-foreign-object>
    <svg:svg xmlns:svg="http://www.w3.org/2000/svg"
             width="170" height="145">
        <svg:g style="stroke:black; fill:black">
          <svg:polygon points="  5, 50,  5, 81, 12, 64"/>
          <svg:polygon points="  5, 45, 41,116, 41, 73"/>
          <svg:polygon points=" 44, 76, 44,119, 61,115"/>
          <svg:polygon points=" 46, 73, 75,140,105, 73"/>
          <svg:polygon points="107, 76,106,119, 89,115"/>
          <svg:polygon points="144, 45,109,116,109, 73"/>
          <svg:polygon points="145, 41,167,  4,109, 71"/>
          <svg:polygon points=" 66, 72, 75, 63, 84, 72"/>
        </svg:g>
    </svg:svg>
   </instream-foreign-object>
  </block>
  <block font-size="20pt" font-weight="bold">Crane Logo</block>
 </block>
```

 Copyright © Crane Softwrights Ltd.

`<instream-foreign-object>` Object (cont.)

The rendering of the embedded image



Copyright © Crane Softwrights Ltd.

Exposing the image size scaling factor

All images have an intrinsic size

- that size built in to the object's specification
- for an external image it is the image dimensions
  - each external format has a method of specifying the "default" dimension
- for an inline image it is the image content width
  - e.g. for SVG the size is defined by the grid specified in the `height=` and `width=` properties of the document element

`<scaling-value-citation>` exposes as an inline character string the scaling factor used by the formatter for the rendering of an image

- the integer number of the scaling factor percentage is expressed as a set of digits (without the percent sign)
- when rendered full size without scaling the value is "`100`" (without quotes)
  - a shrunken image returns a value less than 100
  - a stretched image returns a value greater than 100
- `ref-id=` points to the image
  - the image needs to have an `id=` property
- `scale-option=` indicates width (default) or height
- `intrinsic-scale-value=` indicates the intrinsic size of the image specification compared to what is considered the true image size
  - stating `intrinsic-scale-value="25%"` indicates the intrinsic size found in the object expression is only one quarter of what is considered the image's actual size
- no arithmetic can be performed with the resulting number the formatting object
  - it is just a sequence of characters to be displayed

 Copyright © Crane Softwrights Ltd.

<scaling-value-citation> Object

Chapter 5 - Generic body constructs
Section 2 - Graphics and foreign objects

Purpose:

- expose the scaling factor applied to a graphic

Content:

- 1 (6.6.15) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other required property:
ref-id=(7.30.13;491)

Other optional properties:
alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
country=(7.10.1;467)
dominant-baseline=(7.14.5;468)
format=(7.26.1;473)
grouping-separator=(7.26.2;473)
grouping-size=(7.26.3;474)
id=(7.30.8;475)
1 index-class=(7.24.1;475)
1 index-key=(7.24.2;475)
1 intrinsic-scale-value=(7.30.9;476)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
language=(7.10.2;478)
letter-spacing=(7.17.2;479)
letter-value=(7.26.4;479)
line-height=(7.16.4;479)
1 scale-option=(7.30.14;494)
score-spaces=(7.30.15;495)
text-altitude=(7.29.4;500)
text-decoration=(7.17.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
text-transform=(7.17.6;501)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:
font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)
1 xml:lang=(7.31.24;505)

Properties of interest:

- ref-id= points to the image being reported
- scale-option= indicates width or height
- intrinsic-scale-value= indicates the intrinsic size ratio of the images full size
- 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 442)

Copyright © Crane Softwrights Ltd.

Link requirements

Unidirectional hyperlinks from clickable content to a target location

- `<basic-link>`
  - content is the clickable content used by the document reader to traverse to the target of the link
    - requires support by the user agent
    - no need to be supported for a print-only medium
    - any content can be wrapped up in a link construct
  - a strict interpretation of the XSL-FO recommendation enforces only the construct's inline area to be "hot" and not the areas generated by descendants of the formatting object
- location can be another point within the document
  - use `internal-destination=` property
  - the user agent software reading the document would navigate the reader to the target page
- location can be a point outside the document
  - use `external-destination=` property
  - the operating system running the user agent software would probably invoke the application associated with the file being pointed to
    - e.g. a web browser for an HTML page

The object is an inline construct

- must be placed in a block to be a block-level construct
- may contain either block-level or inline-level constructs

The construct is unidirectional

- there is no back-link property associated with the link itself
- the ability for a user agent to "go back" to where the link originated is a function of the user agent based on a history of the user's locations
  - the user agent is not traversing a property of the link itself that indicates where the link was made from

 Copyright © Crane Softwrights Ltd.

<basic-link> Object

Chapter 5 - Generic body constructs
Section 3 - Links

Purpose:

- the inline display of the start resource of a unidirectional link to a single end point

Content:

- (6.9.2) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
destination-placement-offset=(7.23.5;467)
dominant-baseline=(7.14.5;468)
external-destination=(7.23.6;469)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
indicate-destination=(7.23.7;476)
internal-destination=(7.23.8;476)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
line-height=(7.16.4;479)
show-destination=(7.23.9;495)
target-presentation-context=(7.23.12;499)
target-processing-context=(7.23.13;500)
target-stylesheet=(7.23.14;500)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)
vertical-align=(7.31.22;502)

Copyright © Crane Softwrights Ltd.

<basic-link> Object (cont.)

An excerpt from the samp/leadlink.fo sample (page 130):

```
<block text-align-last="justify"
       end-indent="1cm" start-indent="1cm">
 <basic-link internal-destination="N66">Third Title</basic-link>
 <leader leader-pattern="dots"/>
 <basic-link internal-destination="N66">
  <page-number-citation ref-id="N66"/>
 </basic-link>
</block>
...
<block space-before="10pt" text-align-last="justify"
       end-indent="1cm" start-indent="1cm">
 <basic-link internal-destination="N1">Page count</basic-link>
 <leader leader-pattern="dots"/>
 <basic-link internal-destination="N1">
  <page-number-citation ref-id="N1"/>
 </basic-link>
</block>
...
<page-sequence master-reference="bookpage">
 <title>The Third Chapter Title</title>
 <flow flow-name="bookpage-body">
  <block id="N66" font-size="16pt * 1.5">Third Title</block>
  <block space-before="16pt">Third entry is very short.</block>
  <block id="N1"/>
 </flow>
</page-sequence>
```


Copyright © Crane Softwrights Ltd.

Elastic and inelastic inline areas

A leader gives the stylesheet writer a number of basic functions:

- fixed-length horizontal space, rules or patterns typically used as a separator
  - e.g. a gap between words of precisely .5cm not using spaces
  - e.g. a rule of a particular length used in a form presentation
- elastic leader rules or patterns in tables of content used to aid the eye movement
- elastic inline gaps used to push away adjacent areas

An inline construct that renders a leader along the text baseline of the line

- `<leader>` - elastic
  - all elastic leaders on a justified line are expanded before any space character or inter-character space on the line is expanded
  - expansion grows evenly across all leaders on the line until the line is full
  - no justification expansion is done on any space characters or inter-character spacing
- `<leader leader-length="`*`length-value`*`">` - inelastic measured amount
  - a fixed length leader is useful for spacing items on a line
- `<leader leader-length.optimum=".5em">`
  - a "half-font-size" starting size is useful when justifying crowded lines so as to not occupy too much space to begin with
  - when using only `.optimum=`, the `.maximum=` is 100% and the `.minimum=` is 0pt
  - when using `leader-length=` the optimum, min and max are the same
  - the default `.optimum=` size is 12pt which might be too big for the line
- must be put into a stand-alone block to act as a block construct
  - note the default line height of the block is the font size, not the line height of the highest construct
  - without specifying a small line height on the block, adjacent blocked lines will not appear close together

Copyright © Crane Softwrights Ltd.

Elastic and inelastic inline areas (cont.)

When used as a leader from one margin to the other margin, must use `text-align-last=` property on the containing block

- using `text-align=` is insufficient for the value "`justify`" because the line of a single-line block is considered the last line of the block
- the leader grows between the information at the start of the line and the information at the end of the line
    - e.g. a dot leader used in table of contents entries
    - e.g. an empty leader pushing two pieces of information to each end of the same line

The `samp/leadlink.fo` sample (page 130) illustrates basic needs

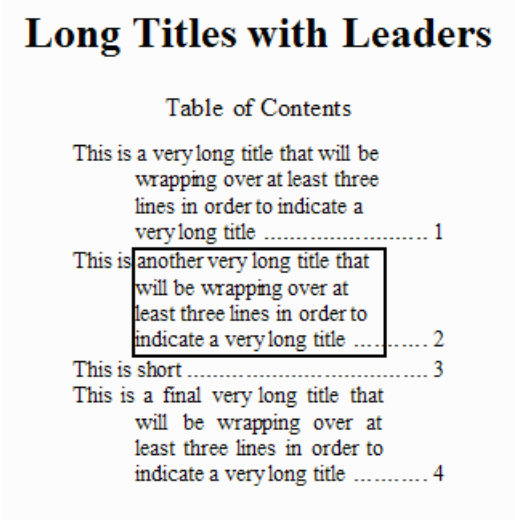
- block-level leaders are used as separators and decoration
    - special block-level considerations for tightly-spaced leaders
    - block-level leaders need not be full block width
- title entries are short
    - leaders expand between end of title and start of page number

 Copyright © Crane Softwrights Ltd.

Elastic and inelastic inline areas (cont.)

The `samp/leaders.fo` example illustrates complex table of contents formatting requirements:

- the author of the XML being formatted is responsible for the length of titles
- the stylesheet is collecting and presenting the titles in a table of contents
- short titles are easily accommodated through simple margin specifications
- if the user could possibly create long titles, the stylesheet writer should be prepared to accommodate wrapping conditions to avoid confusion when reading the table
    - wrapping at the start of the line could confuse the count of entries in the table of contents
        - the wrapping point should be indented on the start side
    - wrapping at the end of the line could confuse the presentation of page numbers
        - the wrapping point should be indented on the end side
- a block's first and last lines can be hanging in respective outdents to give the desired unambiguous presentation at the starts and ends of the lines



Note that the lines of the block are shown only to illustrate the dimension of the block construct itself; this was done simply by turning on the `border-style="solid"` property

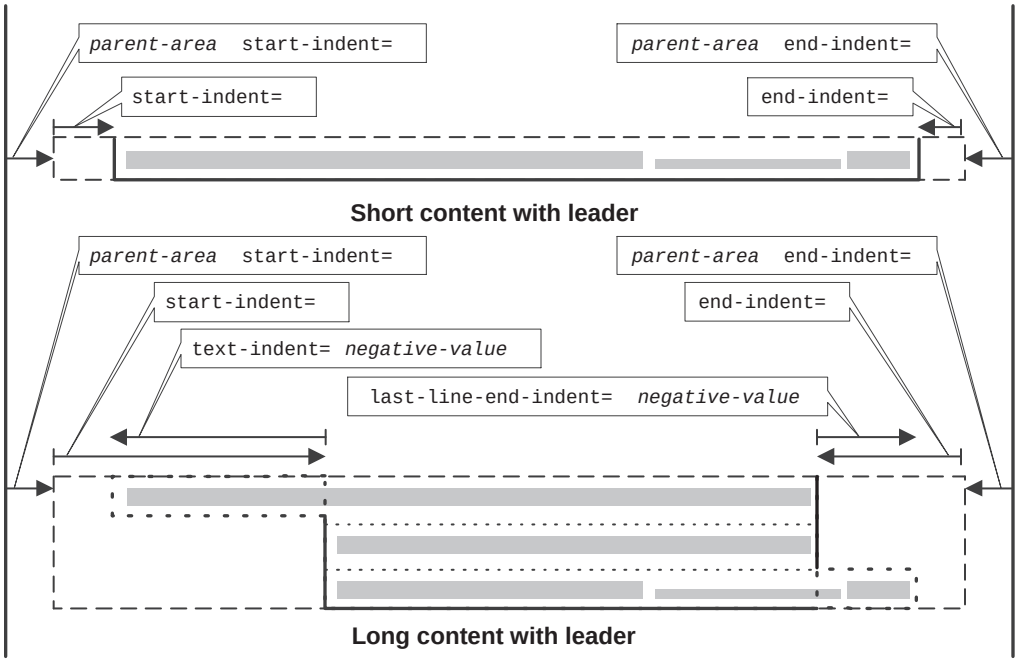
- the indents of the first and last lines of the block are clearly shown outside the dimensions of the block

Copyright © Crane Softwrights Ltd.

Elastic and inelastic inline areas (cont.)

Multiple-line table-of-content-entry blocks are more complex than single-line blocks:

- special considerations for overhanging first and last lines of block
  - negative values for indents specify outdents and allow the content to go beyond the boundaries of the box
- may also specify text alignment for the body of the block



Of note:

- the last line is justified in order to grow the leader to the complete width of the last line including the outdent
- the net indents of the first and last line are the differences between the indents of the block and the respective outdents of the first and last line

 Copyright © Crane Softwrights Ltd.

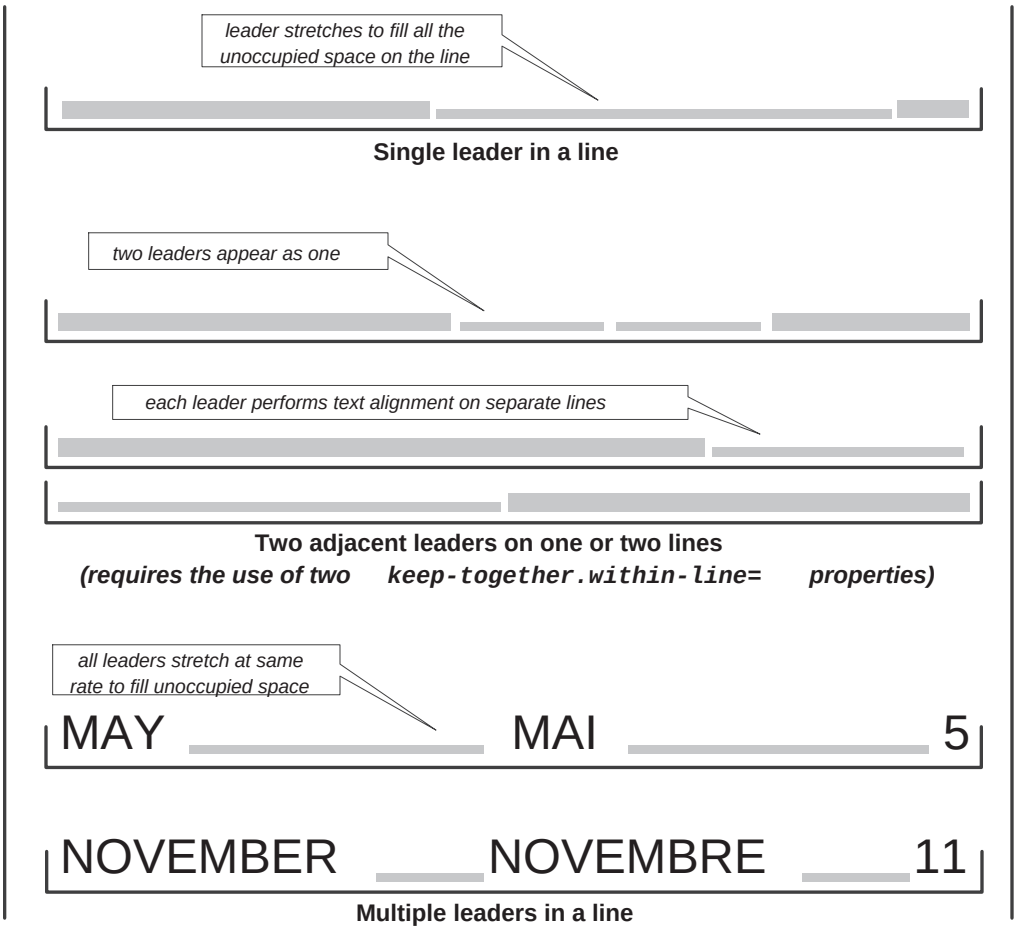
Multiple leaders on a single line

Chapter 5 - Generic body constructs
Section 4 - Leaders

When using multiple leaders on a single line, the length of each leader grows evenly in tandem:

- the length of all leaders on a single justified line is the same
- lengths grow until line is filled
- best way to push two pieces of contents to opposite ends of the line

Examples in `samp/leader-align.fo` of using one or more leaders

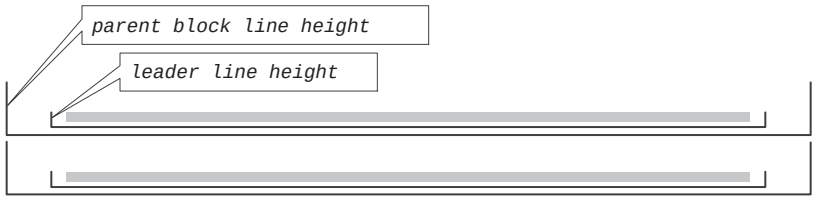


Copyright © Crane Softwrights Ltd.

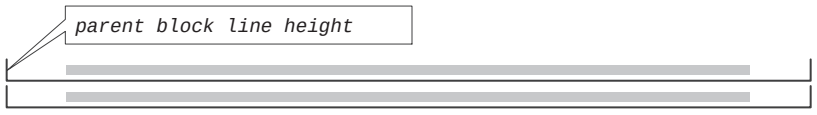
Controlling the distance between leaders

The intuitive approach to controlling the distance between lines does not apply

- consider two standalone leaders that necessarily require two standalone blocks, one leader in each block as in the `samp/leadlink.fo` sample (page 130):
- the line height of the block is governed by block properties and by default is the maximum of the font size and any inline constructs found on the line
- it is intuitive to adjust the line height of the leader, but that if that line height is less than the line height of the block, there will be no change in the result because of the line height of the block will govern the placement of the leaders



Changing leader line height may not affect intervening distance



Changing block line height brings leaders closer together

Important to keep font size consistent when positioning adjacent lines

- leaders are drawn along the font baseline
- changing the line height does not change the font size
- changing the font size will change both the distance between the alignment point and the baseline and the height of the line in the block
- changing only the line height will not change the distance between the alignment point and the baseline
- more consistent distances are rendered when only changing one variable instead of two

 Copyright © Crane Softwrights Ltd.

<leader> Object

Chapter 5 - Generic body constructs
Section 4 - Leaders

Purpose:

- the inline display of a rule or a sequence of characters
  - object can be placed in a <block> object for block-level display

Content:

- (6.6.9) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- cannot contain as a child or descendant any <leader>, <inline-container>, <block-container>, <float>, <footnote>, or <marker> objects

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
color=(7.18.1;465)
dominant-baseline=(7.14.5;468)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
leader-alignment=(7.22.1;478)
leader-length=(7.22.4;478)
leader-pattern=(7.22.2;478)
leader-pattern-width=(7.22.3;479)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
rule-style=(7.22.5;494)
rule-thickness=(7.22.6;494)
text-altitude=(7.29.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Copyright © Crane Softwrights Ltd.

Properties of note:

- the `line-height=` property of the leader does not impact the `line-height=` property of the containing block
  - it may be necessary to adjust the `line-height=` property of the block to achieve the desired result
- when `leader-pattern=` is "`use-content`" the object children of this object are replicated in whole and are not clipped to the desired length of the leader
  - the actual distance covered by the leader may be shorter than desired
  - using short sequences for repeated content will produce better results

 Copyright © Crane Softwrights Ltd.

<leader> Object (cont.)

An excerpt from the samp/leadlink.fo sample (page 130):

```
<block font-weight="bold" font-size="16pt * 2"
  text-align="center">The Leader/Link/Graphic Example</block>
<block line-height="3px">
 <leader leader-pattern="rule" leader-length="100%"/>
</block>
<block line-height="3px">
 <leader leader-pattern="rule" leader-length="100%"/>
</block>
<block>
 <leader leader-pattern="rule" leader-length="100%"/>
</block>
<block line-height="3px">
 <leader leader-pattern="rule" leader-length="100%"/>
</block>
<block line-height="3px">
 <leader leader-pattern="rule" leader-length="100%"/>
</block>
...
<block text-align-last="justify"
      end-indent="1cm" start-indent="1cm">
 <basic-link internal-destination="N9">First Title</basic-link>
 <leader leader-pattern="dots"/>
 <basic-link internal-destination="N9">
  <page-number-citation ref-id="N9"/>
 </basic-link>
</block>
...
<block space-before="10pt" text-align="center">
 <leader leader-pattern="rule" leader-length="60%"/>
</block>
```


Copyright © Crane Softwrights Ltd.

<leader> Object (cont.)

An excerpt from the samp/leaders.fo sample (page 165):

```
01   <block text-align-last="justify" border-style="solid"
02          start-indent="5cm" end-indent="5cm"
03          text-indent="-2cm" last-line-end-indent="-2cm">
04    <inline>This is another very long title that will be
05 wrapping over at least three lines in order to indicate a
06 very long title</inline>
07    <leader leader-pattern="dots"/>
08    <inline>2</inline>
09   </block>
10
11   <block text-align-last="justify"
12          start-indent="5cm" end-indent="5cm"
13          text-indent="-2cm" last-line-end-indent="-2cm">
14    <inline>This is short</inline>
15    <leader leader-pattern="dots"/>
16    <inline>3</inline>
17   </block>
18
19   <block text-align-last="justify" text-align="justify"
20          start-indent="5cm" end-indent="5cm"
21          text-indent="-2cm" last-line-end-indent="-2cm">
22    <inline>This is a final very long title that will be
23 wrapping over at least three lines in order to indicate a
24 very long title</inline>
25    <leader leader-pattern="dots"/>
26    <inline>4</inline>
27   </block>
```

 Copyright © Crane Softwrights Ltd.

Chapter 6 - Tables

- Introduction - Tabular presentation of information
- Section 1 - Tabular structure
- Section 2 - Tabular appearance

Outcomes:

- understand basic concepts of tabular presentation

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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 128) 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

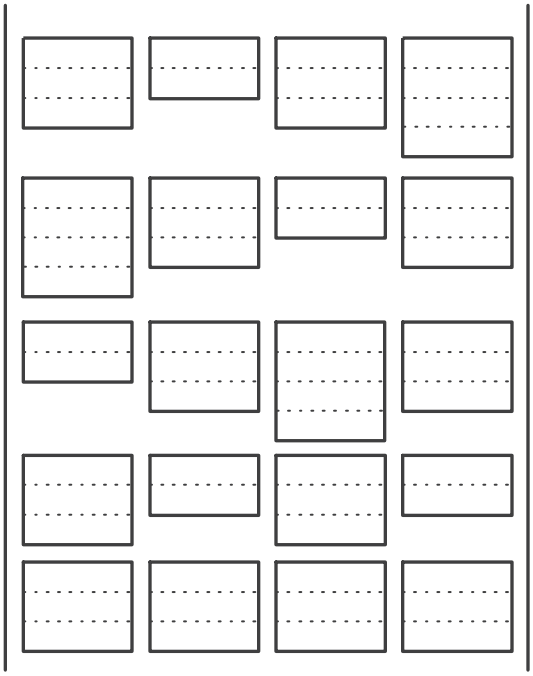
- table contains rows (in block-progression direction) of cells (in inline-progression direction) where each cell contains block-level areas (in block-progression direction)
  - block-level areas would otherwise be arranged in the block-progression direction
- column widths can differ but are fixed for 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

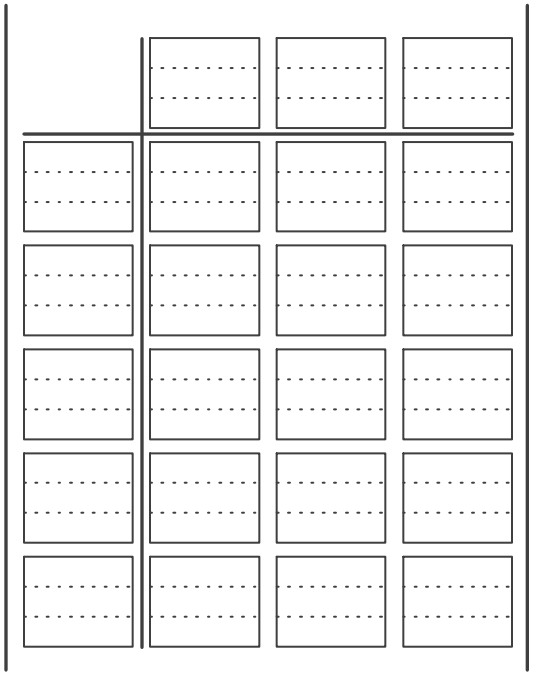
 Copyright © Crane Softwrights Ltd.

Tabular presentation of information (cont.)

Two layout objectives using the same layout constructs:

- 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





Of note:

- 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)

Copyright © Crane Softwrights Ltd.

Tabular presentation of information (cont.)

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

Copyright © Crane Softwrights Ltd.

Tabular presentation of information (cont.)

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Tabular presentation of information (cont.)

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

 Copyright © Crane Softwrights Ltd.

Aligned tuples of block-level constructs

Standalone blocks cannot be adjacent to each other in the inline-progression direction

- a block-level construct typically breaks the block-progression direction

Tuples of adjacent blocks are useful to associate a set of pieces of content

- sets of one-dimensional information
  - each row of the table is a set of information items
  - as many rows exist as there are sets of information
- two-dimensional information
  - the columns represent an axis of the information
  - the rows represent an axis of the information
  - the cells represent the relationship between the members of each axis
- arbitrary layout positioning using a borderless grid
  - spanning columns and spanning rows provides for fine-grained control
  - nesting tables can provide a lot of flexibility

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

- don't consider that only tables in your XML information can use this construct for layout
- regard the construct as a layout facility for multiple block-level constructs in the inline-progression direction of the parent block
- for two-column tables, a list construct may suffice or offer other layout features you need

The content of the columns may determine the edges of the columns

- an auto-layout table adapts column widths to the widths of the cell contents
  - the start and end edges of each block in a set are relative to the contents of respective members of all sets
- the fixed-width table layout ignores the widths of the cell contents
- once determined, the widths of the columns are fixed for the entire table

Copyright © Crane Softwrights Ltd.

Table-related formatting objects

Chapter 6 - Tables
Section 1 - Tabular structure

A table is a block-level construct

- two ways to begin a table-related hierarchy
  - `<table-and-caption>`
    - associates a caption that is rendered on the first page on which part of the table is presented
  - `<table>`
    - the collections of header, footer and body contents of the table
- table widths are specified as with other block-level constructs
  - possible fixed width
  - possible width and height relative to containing block
  - possible sum of all column widths
- no limitations on the location of `<table>` or `<table-and-caption>` objects in an instance
  - tables can be nested within table cells
  - tables can be positioned inline through use of the `<inline-container>` object
- same kinds of attributes as other block-level constructs

A caption can be rendered outside of the boundaries of the table

- `<table-caption>` (optional)
  - positioned according to the `caption-side=` property
  - this is optional so that the `<table-and-caption>` properties can be used to align the nested `<table>`

The indeterminate room on a page for rows necessitates special handling for page breaks

- the transformation creating the XSL-FO cannot know how much of the table will fit on each page
  - the formatter is in charge of determining where the page-breaks belong
- body content dictates how many pages are used by the rows that are generated
  - `<table-body>` (mandatory and possibly repeated)
    - the rows of the table that are not repeated and make up the body of the table content

Copyright © Crane Softwrights Ltd.

Table-related formatting objects (cont.)

Column widths may be calculated from content

- requires that the table has the `table-layout=` property of "`auto`"
- this is the initial value for the property and must be changed to get fixed layout behavior
- implemented-defined algorithm will not ensure the same presentation between different processors

Table column properties specified using `<table-column/>`

- an empty formatting object that does not generate any areas
- the column number need not be specified if only one greater than the previous sibling specification's column number
  - first column specification is assumed to be for column one if the column number is not specified
- formatting properties in this object can be utilized by table cells and their descendants by explicitly using the `from-table-column()` function

Specified `column-width=` column widths can be relative or explicit

- may need to be specified either way for all columns
  - when `inline-progression-dimension=` is an explicit length and the `table-layout=` property is "`fixed`"
- possible relative value using a "unit" specification
  - each relatively sized column is specified as a quantity of abstract units
  - the sum of units of all columns with units is the basis of a single unit's value
  - `proportional-column-width(`*`number-of-units`*`)`
    - the length value returned is the length of the number of units requested
  - the `table-layout=` must be "`fixed`" and the `inline-progression-dimension=` cannot be "`auto`" (but may be "`100%`" to be as wide as possible)
- possible fixed value using a length specification
- can mix columns with relative and explicit specifications
  - the relative values are calculated from the remainder of the table after subtracting all of the explicit values

Copyright © Crane Softwrights Ltd.

Table-related formatting objects (cont.)

Constructs of a global nature are rendered on each page that gets table content

- `<table-header>` (optional)
  - rendered before the before edge of the rows of the table body used on each page with table body content
  - if the header need not be repeated, define all rows of the header once in the body before the first of the body content
- `<table-footer>` (optional)
  - rendered after the after edge of the rows of the table body used on each page with table body content
  - if the footer need not be repeated, define all rows of the footer once in the body after the last of the body content

The strategy for building rows in a header, footer, or body row group must either all be containerized or all be triggered by cell properties

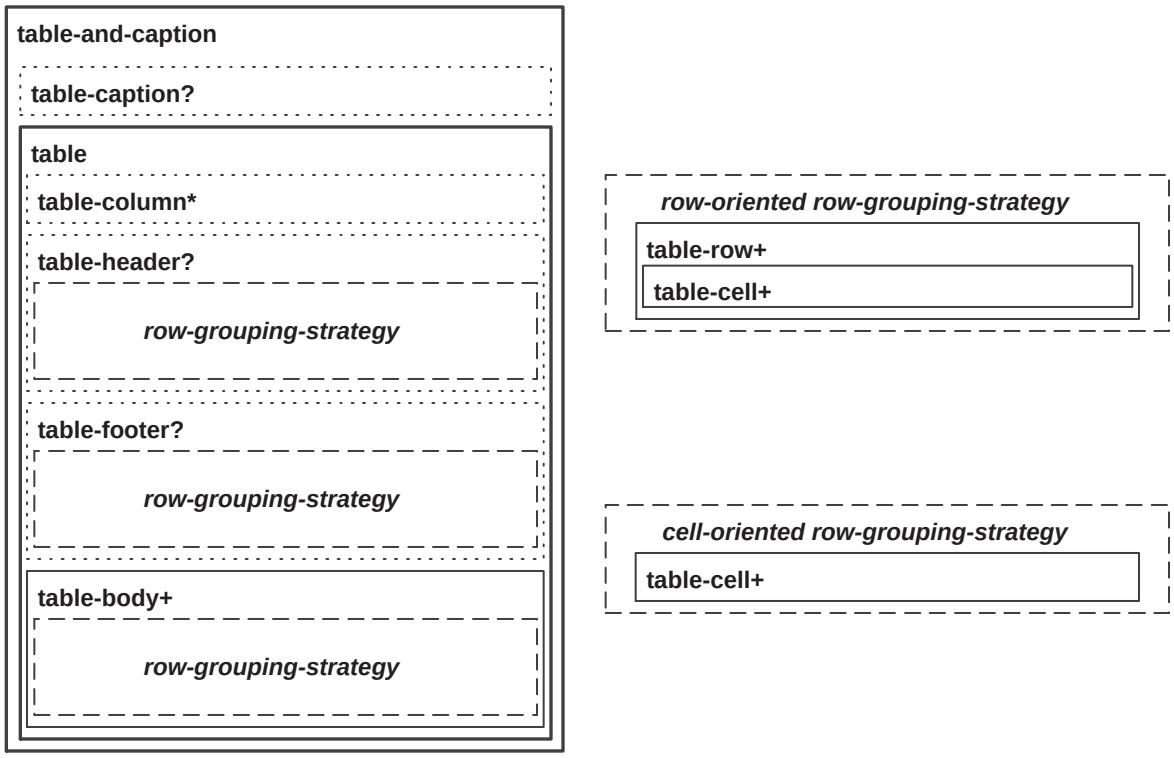
- containerized use of `<table-row>` objects
  - each containing `<table-cell>` objects
  - supports transformation of source information row-related container constructs
  - opportunity for row-wide inheritance of properties
- triggered use of only `<table-cell>` objects
  - using `starts-row=` and `ends-row=` properties
  - supports algorithmic breaking of cells into rows when source is not containerized
  - properties represent conditions to be met and not actions to be performed
    - a cell with `starts-row=` following a cell with `ends-row=` does not produce an empty row
- multiple `<table-body>` objects allowed in a single table
  - each `<table-body>` object allows a portion of the table to use a different row strategy than previous body portion
  - each row group may be defined differently than other row groups, but a single row group is only either containerized or triggered

 Copyright © Crane Softwrights Ltd.

Table-related formatting objects (cont.)

Chapter 6 - Tables
Section 1 - Tabular structure

Summary of table structures and row grouping strategies:



- thick lines indicate only those objects that can be the apex of a table structure
- the choice of row strategy is within each sibling row group (header, footer, or body)
    - one choice for a given row group does not dictate any required choice for the sibling row groups

Copyright © Crane Softwrights Ltd.

Table and cell borders

Table and cell borders can be visible or invisible

- border sides named by writing directions
- visible borders can be a specified thickness, pattern or color
  - specified for the table, table row and table cell perimeters
    - applicable to table rows only when borders are collapsed, not when separated
  - the formatter can have any default value for thickness if not specified by the property
- some border-related properties are not inherited unless explicitly specified as "`inherit`"

Co-incident cell and table borders can be arbitrated or separated

- use `border-collapse=` property to specify behavior
- "`collapse`" (initial value) specifies arbitration of properties based on relative values
  - arbitrates `border-width=`, `border-style=` and `border-color=`
    - summarized later on in Borders (page 335)
- "`collapse-with-precedence`" limits arbitration to only a numeric precedence value (implied or specified)
- "`separate`" will distinguish borders of adjacent cells and cells adjacent to table borders
  - visibly separated by amount of the `border-separation=` property
- border properties are allowed on `<table-row>` only with collapsed borders

Precedence guides the arbitration between two conflicting border specifications

- each table construct has a different initial value for border precedence
  - 5 - `<table>`
  - 4 - `<table-cell>`
  - 3 - `<table-row>`
  - 2 - `<table-body>`
  - 1 - `<table-header>`
  - 0 - `<table-footer>`
- a numeric value can be explicitly specified using `border-*-precedence=`
- "`force`" value overrides all numeric values

Empty cells can be treated specially

- by default an empty cell will have its border properties respected
- using `empty-cells=` on `<table-cell>` can hide the border of a cell that has no content, as is displayed in browsers for HTML
  - initial value is "`show`" to render separated borders for empty cells
- if all cells of a row are hidden, the row disappears entirely

 Copyright © Crane Softwrights Ltd.

Spanning cells

Cells can occupy more than one column or row

- useful in headers and stubs for "span heads" across multiple columns or rows
- useful in body content to reflect a single value for multiple rows or columns
- specified using an integer count greater-than or equal to 1
    - floating values are rounded
    - numbers less than 1 are interpreted as 1
    - a value calculated as 1 does not trigger any spanning

Cell spanning progresses towards the end and after edges, starting with current cell

- in the column-progression-direction for column spanning
    - `number-columns-spanned=`
- in the row-progression-direction for row spanning
    - `number-rows-spanned=`

Spanning rows takes away from subsequent rows' unnumbered columns

- the cell does not exist that would be in a row except for the cell above whose spanning value affects the row
- an unnumbered cell will "skip" over spanned cells from preceding rows
- a `column-number=` numbered cell will override a spanning cell from a preceding row

Copyright © Crane Softwrights Ltd.

Table and cell alignment

Two ways to position a table in the inline-progression-direction within its parent area

- `start-indent=` on given `<table>`
- `text-align=` on parent `<table-and-caption>`
  - the `<table-caption>` is optional thus not obliging an aligned table to have a caption

Inheritable property values will be inherited by the table cells, usually producing undesirable results

- could specify resetting values for properties on `<table-body>`, `<table-header>`, or other suitable ancestral location to be inheritable by descendant cells

The content of cells can be aligned through inherited properties

- cell before/center/after alignment through the use of `display-align=` property
  - default is "`auto`"
- cell start/center/end alignment through use of individual blocks' `text-align=` property
  - only significant use of a string argument for `text-align=`
    - e.g. using the decimal separator character as the string to align a column of cells with financial information vertically
    - a string argument used in the property any other context is interpreted as the value "`start`"
  - need not be specified in the cell object (though could be specified to be inherited by blocks)
- cell row before/baseline alignment through the use of `relative-align=` property
  - default is "`before`"
  - may want to use "`baseline`" for the use of fonts of different sizes
  - only applies if `display-align=` is "`auto`"

 Copyright © Crane Softwrights Ltd.

<table-and-caption> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- the parent object of a captioned collection of tabular content

Content:

- (6.7.2) (table-caption?,table)
- Child objects (alphabetical):
  - <table>(6.7.3;192)
  - <table-caption>(6.7.5;190)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)
- Common Relative Position Properties(7.13;434)

Other optional properties:

break-after=(7.20.1;463)
break-before=(7.20.2;463)
caption-side=(7.28.7;463)
clear=(7.19.1;465)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
text-align=(7.16.9;500)

Shorthands influencing the above properties:

page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

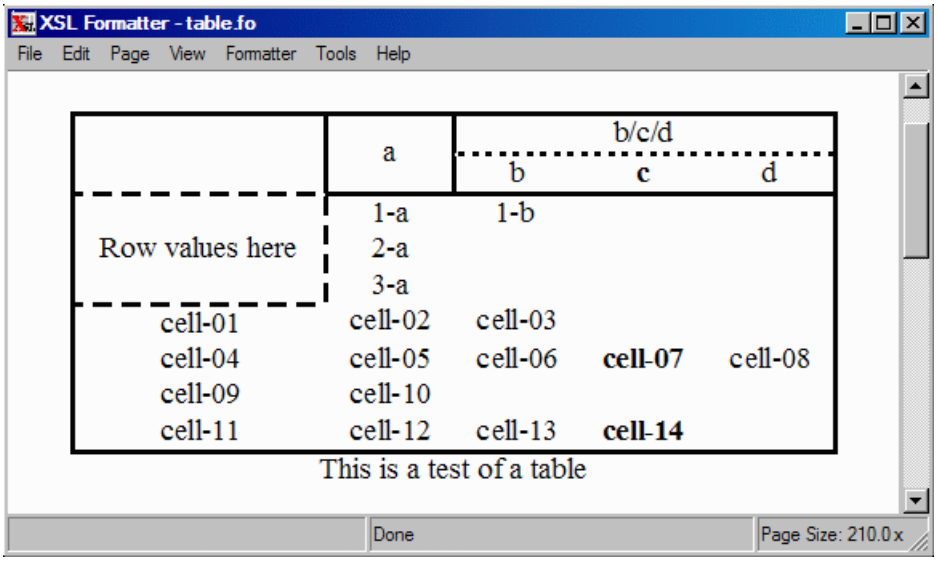
Property of interest:

- text-align= used to align the table
  - this is the only non-mathematical way of aligning a table within its parent area

Copyright © Crane Softwrights Ltd.

`<table-and-caption>` Object (cont.)

The `samp/table.fo` example:



Of note:

- the table has adjacent body segments whose boundaries cannot be distinguished
- all the cells in the column with the "c" heading are bold
- the caption follows after the table

Copyright © Crane Softwrights Ltd.

<table-and-caption> Object (cont.)

An excerpt from the samp/table.fo sample:

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    ...
    <table-header>
      ...
    </table-header>
    <table-body>
      ...
    </table-body>
    <table-body text-align="center">
      ...
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

<table-caption> Object

Purpose:

- the caption of a captioned collection of tabular content

Content:

- (6.7.5) (%block;)+
- Child object:
  - %block;(6.2;71)
- Referring object:
  - <table-and-caption>(6.7.2;187)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

block-progression-dimension=(7.15.3;453)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
width=(7.15.14;504)

Shorthand influencing the above properties:

page-break-inside=(7.31.18;487)

Properties of interest:

- caption-side= indicates where the caption is relative to the table

Properties of descendent blocks within the caption block

- text-align= used to align the caption content
- the typical CSS use of vertical-align= does not apply

 Copyright © Crane Softwrights Ltd.

<table-caption> Object (cont.)

An excerpt from the samp/table.fo sample (page 188):

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    ...
    <table-header>
      ...
    </table-header>
    <table-body>
      ...
    </table-body>
    <table-body text-align="center">
      ...
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

<table> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- the parent object of an uncaptioned collection of tabular content

Content:

- (6.7.3) (table-column*,table-header?,table-footer?,table-body+)
- Child objects (alphabetical):
  - <table-body>(6.7.8;199)
  - <table-column>(6.7.4;194)
  - <table-footer>(6.7.7;198)
  - <table-header>(6.7.6;196)
- Referring object:
  - <table-and-caption>(6.7.2;187)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)
- Common Relative Position Properties(7.13;434)

Other optional properties:

block-progression-dimension=(7.15.3;453)
border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-collapse=(7.28.3;457)
border-end-precedence=(7.28.4;457)
border-separation=(7.28.5;460)
border-start-precedence=(7.28.6;461)
break-after=(7.20.1;463)
break-before=(7.20.2;463)
clear=(7.19.1;465)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
table-layout=(7.28.16;499)
table-omit-footer-at-break=(7.28.17;499)
table-omit-header-at-break=(7.28.18;499)
width=(7.15.14;504)
writing-mode=(7.29.7;504)

Shorthands influencing the above properties:

border-spacing=(7.31.9;460)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

Copyright © Crane Softwrights Ltd.

<table> Object (cont.)

An excerpt from the samp/table.fo sample (page 188):

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    ...
    <table-header>
      ...
    </table-header>
    <table-body>
      ...
    </table-body>
    <table-body text-align="center">
      ...
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

`<table-column>` Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- the specification of common columnar properties

Content:

- (6.7.4) `EMPTY`
- Referring object:
  - `<table>`(6.7.3;192)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)

Other optional properties:

`border-after-precedence=`(7.28.1;454)
`border-before-precedence=`(7.28.2;455)
`border-end-precedence=`(7.28.4;457)
`border-start-precedence=`(7.28.6;461)
`column-number=`(7.28.8;466)
`column-width=`(7.28.9;466)
`number-columns-repeated=`(7.28.12;483)
`number-columns-spanned=`(7.28.13;483)
`visibility=`(7.30.17;502)

Property of interest:

- `column-width="proportional-column-width(`*`number-of-units`*`)"`
  - calculated as a unit proportion of calculating the table width less all specified column widths
- inheritable properties specified here are not automatically inherited by cells of the column
  - the cells are not descendants in the formatting object tree during refinement
  - cells must use the `from-table-column()` function to find the table column properties

Copyright © Crane Softwrights Ltd.

`<table-column>` Object (cont.)

An excerpt from the `samp/table.fo` sample (page 188):

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    <table-column column-width="proportional-column-width(2)"/>
    <table-column column-width="proportional-column-width(1)"/>
    <table-column column-width="proportional-column-width(1)"/>
    <table-column column-width="proportional-column-width(1)"
                  font-weight="bold"/>
    <table-column column-width="proportional-column-width(1)"/>
    <table-header>
      ...
    </table-header>
    <table-body>
      ...
    </table-body>
    <table-body text-align="center">
      ...
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-06</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-07</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)"
                  ends-row="true">
        <block>cell-08</block></table-cell>
      ...
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

<table-header> Object

Purpose:

- the rows of tabular content repeated at the before edge of every break in body content
- must choose one row grouping strategy for the entire header

Content:

- (6.7.6) (table-row+|table-cell+)
- Child objects (alphabetical):
  - <table-cell>(6.7.10;203)
  - <table-row>(6.7.9;201)
- Referring object:
  - <table>(6.7.3;192)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-end-precedence=(7.28.4;457)
border-start-precedence=(7.28.6;461)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
visibility=(7.30.17;502)

Copyright © Crane Softwrights Ltd.

`<table-header>` Object (cont.)

An excerpt from the `samp/table.fo` sample (page 188):

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    ...
    <table-header>
      <table-row>
        ...
      </table-row>
      <table-row>
        ...
      </table-row>
    </table-header>
    <table-body>
      <table-row text-align="center">
      ...
      <table-row text-align="center">
      ...
    </table-body>
    <table-body text-align="center">
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-01</block></table-cell>
      ...
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-14</block></table-cell>
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

<table-footer> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- the rows of tabular content repeated at the after-edge of every break in body content
- must choose one row grouping strategy for the entire footer

Content:

- (6.7.7) (table-row+|table-cell+)
- Child objects (alphabetical):
  - <table-cell>(6.7.10;203)
  - <table-row>(6.7.9;201)
- Referring object:
  - <table>(6.7.3;192)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-end-precedence=(7.28.4;457)
border-start-precedence=(7.28.6;461)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
visibility=(7.30.17;502)

Copyright © Crane Softwrights Ltd.

<table-body> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- the rows of tabular content flowed as the body content
- must choose one row grouping strategy for each set of body rows
  - each set of body rows may use a different strategy
  - the boundaries between sets of body rows are seamless

Content:

- (6.7.8) (table-row+|table-cell+)
- Child objects (alphabetical):
  - <table-cell>(6.7.10;203)
  - <table-row>(6.7.9;201)
- Referring object:
  - <table>(6.7.3;192)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-end-precedence=(7.28.4;457)
border-start-precedence=(7.28.6;461)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
visibility=(7.30.17;502)

Copyright © Crane Softwrights Ltd.

<table-body> Object (cont.)

An excerpt from the samp/table.fo sample (page 188):

```
<table border="solid" border-collapse="collapse">
    ...
    <table-header>
      ...
    </table-header>
    <table-body>
      <table-row text-align="center">
        ...
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>2-a</block></table-cell>
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>3-a</block></table-cell>
      </table-row>
    </table-body>
    <table-body text-align="center">
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-01</block></table-cell>
      ...
      <table-cell font-weight="from-table-column(font-weight)"
                  starts-row="true">
        <block>cell-04</block></table-cell>
      ...
      <table-cell font-weight="from-table-column(font-weight)"
                  ends-row="true">
        <block>cell-08</block></table-cell>
      ...
      <table-cell font-weight="from-table-column(font-weight)"
                  starts-row="true">
        <block>cell-11</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-12</block></table-cell>
    </table-body>
  </table>
```

Of note:

- first set of body rows is using the row-based row-grouping-strategy
- second set of body rows is using the cell-based row-grouping-strategy

 Copyright © Crane Softwrights Ltd.

<table-row> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- a row of tabular content

Content:

- (6.7.9) (table-cell+)
- Child object:
  - <table-cell>(6.7.10;203)
- Referring objects:
  - <table-header>(6.7.6;196)
  - <table-footer>(6.7.7;198)
  - <table-body>(6.7.8;199)

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

block-progression-dimension=(7.15.3;453)
border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-end-precedence=(7.28.4;457)
border-start-precedence=(7.28.6;461)
break-after=(7.20.1;463)
break-before=(7.20.2;463)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
visibility=(7.30.17;502)

Shorthands influencing the above properties:

page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

Copyright © Crane Softwrights Ltd.

<table-row> Object (cont.)

An excerpt from the samp/table.fo sample (page 188):

```
<table-and-caption caption-side="after">
  <table-caption>
    <block text-align="center">This is a test of a table</block>
  </table-caption>
  <table border="solid" border-collapse="collapse">
    ...
    <table-header>
      <table-row>
        <table-cell number-rows-spanned="2">
        ...
      </table-row>
      <table-row>
        <table-cell border-before-style="dotted"
                    border-after-style="solid">
        ...
      </table-row>
    </table-header>
    <table-body>
      <table-row text-align="center">
        ...
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>2-a</block></table-cell>
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>3-a</block></table-cell>
      </table-row>
    </table-body>
    <table-body text-align="center">
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-01</block></table-cell>
      ...
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-14</block></table-cell>
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

<table-cell> Object

Chapter 6 - Tables
Section 2 - Tabular appearance

Purpose:

- a column of a row of tabular content

Content:

- (6.7.10) (%block;)+
- Child object:
  - %block;(6.2;71)
- Referring objects:
  - <table-header>(6.7.6;196)
  - <table-footer>(6.7.7;198)
  - <table-body>(6.7.8;199)
  - <table-row>(6.7.9;201)
- may begin with any number of <marker> children

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Relative Position Properties(7.13;434)

Other optional properties:

block-progression-dimension=(7.15.3;453)
border-after-precedence=(7.28.1;454)
border-before-precedence=(7.28.2;455)
border-end-precedence=(7.28.4;457)
border-start-precedence=(7.28.6;461)
column-number=(7.28.8;466)
display-align=(7.14.4;468)
empty-cells=(7.28.10;469)
ends-row=(7.28.11;469)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
number-columns-spanned=(7.28.13;483)
number-rows-spanned=(7.28.14;483)
relative-align=(7.14.6;492)
starts-row=(7.28.15;498)
width=(7.15.14;504)

Properties of interest:

- display-align= is used to align the content in the block-progression-direction

Copyright © Crane Softwrights Ltd.

`<table-cell>` Object (cont.)

An excerpt from the `samp/table.fo` sample (page 188) showing the row-based row-grouping-strategy:

```
<table-and-caption caption-side="after">
  ...
    <table-header>
      <table-row>
        <table-cell font-weight="from-table-column(font-weight)"
                    number-rows-spanned="2">
          <block/>
        </table-cell>
        <table-cell font-weight="from-table-column(font-weight)"
                    number-rows-spanned="2" border="solid"
                    display-align="center">
          <block text-align="center">a</block>
        </table-cell>
        <table-cell font-weight="from-table-column(font-weight)"
                    number-columns-spanned="3">
          <block text-align="center">b/c/d</block>
        </table-cell>
      </table-row>
      ...
    </table-header>
    <table-body>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)"
                    number-rows-spanned="3" border="dashed"
                    display-align="center">
          <block>Row values here</block>
        </table-cell>
        <table-cell font-weight="from-table-column(font-weight)">
           <block>1-a</block></table-cell>
        <table-cell font-weight="from-table-column(font-weight)">
           <block>1-b</block></table-cell>
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>2-a</block></table-cell>
      </table-row>
      <table-row text-align="center">
        <table-cell font-weight="from-table-column(font-weight)">
          <block>3-a</block></table-cell>
      </table-row>
    </table-body>
```

 Copyright © Crane Softwrights Ltd.

<table-cell> Object (cont.)

Another excerpt from the samp/table.fo sample (page 188) showing the cell-based row-grouping-strategy:

```
<table-and-caption caption-side="after">
  ...
    <table-body text-align="center">
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-01</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-02</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-03</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)"
                  starts-row="true">
        <block>cell-04</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-05</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-06</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-07</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)"
                  ends-row="true">
        <block>cell-08</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-09</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-10</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)"
                  starts-row="true">
        <block>cell-11</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-12</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-13</block></table-cell>
      <table-cell font-weight="from-table-column(font-weight)">
        <block>cell-14</block></table-cell>
    </table-body>
  </table>
</table-and-caption>
```


Copyright © Crane Softwrights Ltd.

Chapter 7 - Floats, footnotes and containers

- Introduction - Out-of-line publishing constructs
- Section 1 - Conditional reference areas and sub-regions
- Section 2 - Floats
- Section 3 - Footnotes
- Section 4 - Container basics

Outcomes:

- understand basic concepts of out-of-line constructs

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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

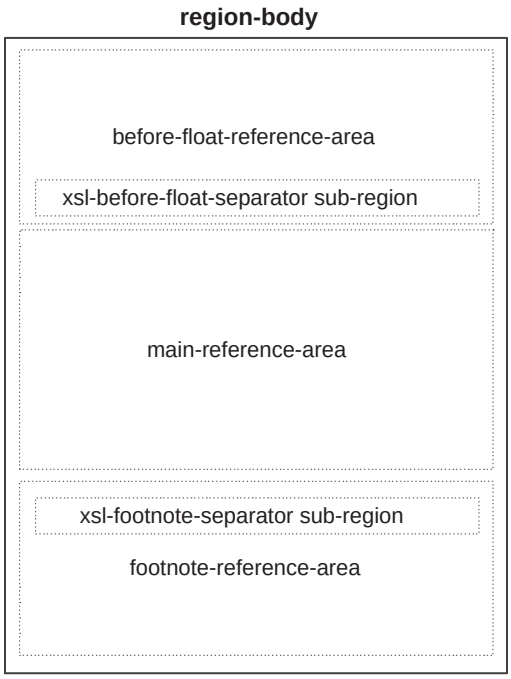
- consider the need mentioned above to format a disclaimer at the bottom of the last page of a document
- one can flow the disclaimer in the body of a footnote with an empty citation in an empty block at the end of the document

 Copyright © Crane Softwrights Ltd.

Out-of-line publishing constructs (cont.)

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

Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

Conditional reference areas and sub-regions

Conditional reference areas exist on a given page only if the formatter needs them on that page

- takes room away from the main-reference-area (see Out-of-line publishing constructs (page 209))
- a conditional reference area grows in the block-progression-direction for as long as is necessary to fit the content flowed into it
- the formatter is responsible for deciding when a conditional reference area is "too big" and must break to the next page
    - no control by the stylesheet writer over where a large conditional reference area breaks to the next page

Sub-region visually delimits the conditional reference-area from the main reference-area

- sub-region content is flowed only once on the page
- prevents need to know the last content of the before-float sub-region or the first content of the footnote sub-region

A sub-region always exists and only appears whenever the conditional reference area exists

- if the conditional reference area is not needed on the page, the sub-region is not seen
- there is no obligation to define sub-region content
    - the region exists but it has no dimension of it has no content

Copyright © Crane Softwrights Ltd.

Conditional reference areas and sub-regions (cont.)

Sub-region content is defined on a per page-sequence basis using static content

- content for the sub-region need only be defined and must be defined if there is chance the formatter might trigger the corresponding conditional reference area
  - if the stylesheet writer knows a particular page-sequence will never have before-floats or footnotes, no sub-region content need be declared
  - sub-region content must be repeated in every page-sequence in which it might be needed
  - it is not an error to define sub-region content for a page-sequence that doesn't need it
- `<static-content>` is a child of `<page-sequence>` and is described in detail in Flowed content vs. static content (page 244)
- the flow in the static-content declaration is directed to the corresponding sub-region on the page
- sub-region names are reserved and cannot be changed
  - ```
    <static-content flow-name="xsl-before-float-separator">
       ...block-level-constructs...
    </static-content>
    ```
    - inside and at the end of the before-float-reference-area
  - ```
    <static-content flow-name="xsl-footnote-separator">
       ...block-level-constructs...
    </static-content>
    ```
    - inside and at the start of the footnote-reference-area

Sub-regions are reference areas

- necessary to set space `.conditionality` to "`retain`" to make room on the page between the conditional reference-area and the main reference-area

 Copyright © Crane Softwrights Ltd.

Float definition

Chapter 7 - Floats, footnotes and containers
Section 2 - Floats

Positioning out-of-line information at the before, start or end sides of a page using the `<float>` object

- before floats are flowed in the before-float-reference-area (see Out-of-line publishing constructs (page 209))
- creates a dimensionless anchor area the area tree
- anchor is tied to the information that precedes the float in the flow
- out-if-line information is placed relative to the anchor
    - typically the top or sides of the page in which the anchor is flowed
- content of the float is always a set of block-level constructs

Information can be defined at the block level or inline level of the flow

- the anchor is treated as a block if defined at the block level
- the anchor is typically treated as an inline construct if defined at the inline level
    - treated as a block object if the line area created by the inline constructs consists only of anchors
- must be defined in the body-region of the page
- must be defined as a descendant of a relatively positioned block
    - absolutely positioned objects cannot have out-of-line descendants
- cannot have any `<float>`, `<footnote>` or `<marker>` descendants

The blocks of the float definition are positioned in the area tree accordingly

- before-floats are stacked in the before-float-reference-area
- side-floats are stacked in a side-float-reference-area
    - child of the specifying block's ancestral span-reference-area
- float content that does not float is flowed as normal content
    - perhaps by a faulty definition or placement
    - a float is an area that is not normally flowed and cannot be defined within any other area that is not normally flowed (e.g. another float, a footnote, a perimeter region, an absolutely positioned block container, etc.)
- float lengths cannot be preset and are always derived from the content of the float

Out-of-line areas have no border or padding, though descendant areas may

- padding-, border-, and content-rectangles are coincident

Side-floats do not overlap and are typically placed beside each other

- makes a further incursion into the inline-progression-dimension of the block
- can be forced to not be placed beside each other by using the `clear=` property
    - different values control which floats are clear of other floats

Copyright © Crane Softwrights Ltd.

<float> Object

Chapter 7 - Floats, footnotes and containers
Section 2 - Floats

Purpose:

- 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

Content:

- (6.12.2) (%block;)+
- Child object:
  - %block;(6.2;71)

Optional properties:

clear=(7.19.1;465)
float=(7.19.2;470)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

Copyright © Crane Softwrights Ltd.

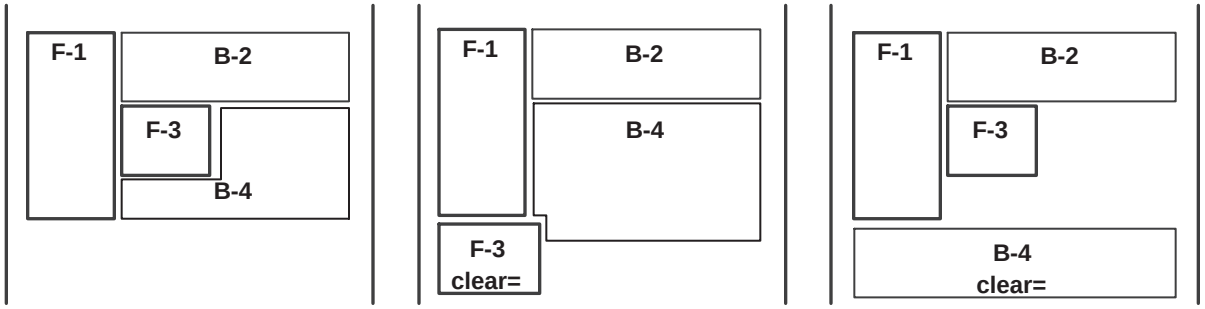
The interaction of blocks and floats

The `clear=` property is used to prevent two side floats from being beside each other or a block-level object not to be beside a float

- not documented as a property with the individual block-level object descriptions
  - applies to all such constructs as documented in the property definition
- the default is for side floats to continue intruding into the main reference area
  - use `clear=` to start a side float back at the edge, clear of any previous float
- the default is that lines of a block will contour around the accumulation of side floats
  - use `clear=` to start a block back at the edge, clear of any previous float
- those constructs with the `clear=` value clear the float they would otherwise be beside

Consider three situations that illustrate the use of `clear=` on each of floats and blocks

- the numbers reflect the order in which the constructs are flowed
- the thick-lined blocks with "F-" prefixes are floats
- the thin-lined blocks with "B-" prefixes are blocks



Of note in the diagram:

- the left-most page fragment shows the default situation where side floats intrude and blocks flow around floats
  - the second float ("F-3") is to the right of the first float ("F1")
  - the second block ("B-4") wraps around both floats
- the center page fragment shows the second float being clear of the first float
  - the second block ("B-4") starts abutted to the first block ("B2")
- the right-most page fragment shows the second block being clear of all floats
  - the second float ("F-3") is to the right of the first float ("F1")
  - the second block ("B-4") is clear of both floats

Copyright © Crane Softwrights Ltd.

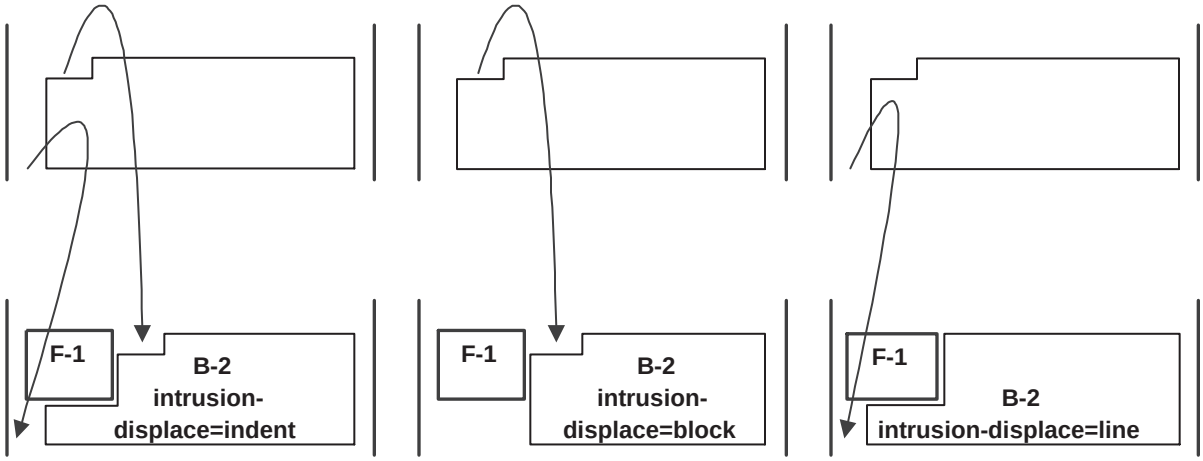
The interaction of blocks and floats (cont.)

The `intrusion-displace=` property indicates the strategy of locating the start (or end) edges and indents of the lines of the block being intruded upon

- note that displacing the edge of a block necessarily displaces all lines in that block

Consider three situations that illustrate the use of `intrusion-displace=` on a block with both `text-indent=` and `start-indent=`

- a block is defined with both `start-indent=` and `text-indent=` properties, and it follows a float that is rendered immediately before, thus impacting on these values
- "`indent`" respects `text-indent=` and `start-indent=`
- "`block`" respects `text-indent=` but limits all lines by the width of the float
- "`line`" ignores `text-indent=` if occupied by the float, and respects `start-indent=`



 Copyright © Crane Softwrights Ltd.

Footnote definition

Positioning out-of-line information at the bottom of a page using the `<footnote>` object

- the anchor area is the last area defined by the `<inline>` child
- out-of-line information is placed relative to the anchor
    - typically the bottom of the page in which the anchor is flowed
- out-of-line content is the content of the `<footnote-body>` child
    - always a set of block-level constructs

Information must be defined at the inline level of the flow

- the object supplies the content of the inline anchor

The blocks of the footnote body are positioned in the area tree accordingly

- flowed in the footnote-reference-area (see Out-of-line publishing constructs (page 209))
- footnote body lengths cannot be preset and are always derived from the content of the float

Copyright © Crane Softwrights Ltd.

Footnote definition (cont.)

Same restrictions as `<float>`

- a footnote's body is an area that is not normally flowed and cannot be defined within any other area that is not normally flowed (e.g. another footnote, a float, a perimeter region, an absolutely positioned block container, etc.)
- must be defined in the body-region of the page as a descendant of a relatively positioned block
- cannot have any `<float>`, `<footnote>` or `<marker>` descendants

No built-in semantics of footnote numbering or citations

- numbering and presentation is the responsibility of the stylesheet or application generating the flow objects
  - what is displayed inline and how it is displayed
    - e.g. superscripting, italics, font-size, etc.
  - how the inline reference is reflected in the out-of-line information for correlation by the reader
- numbering cannot be restarted on a per-page basis
  - the page boundaries are determined by the formatter and are not known to the transformation process
- the XSLT process producing the XSL-FO instance can probably use the same criteria for determining the page sequence boundaries to do footnote numbering either across the page sequence or across the entire publication

No automatic copying of the inline content to the start of the footnote body

- it is the stylesheet writer's responsibility to define both the footnote citation and the footnote body separately
- one may wish to use different formatting properties for each use of the footnote citation

 Copyright © Crane Softwrights Ltd.

<footnote> Object

Chapter 7 - Floats, footnotes and containers
Section 3 - Footnotes

Purpose:

- 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
    - includes the inline content to render in the flow where the footnote body content is defined

Content:

- (6.12.3) (inline,footnote-body)
- Child objects (alphabetical):
    - <footnote-body>(6.12.4;222)
    - <inline>(6.6.7;112)

Property sets:

- Common Accessibility Properties(7.5;426)

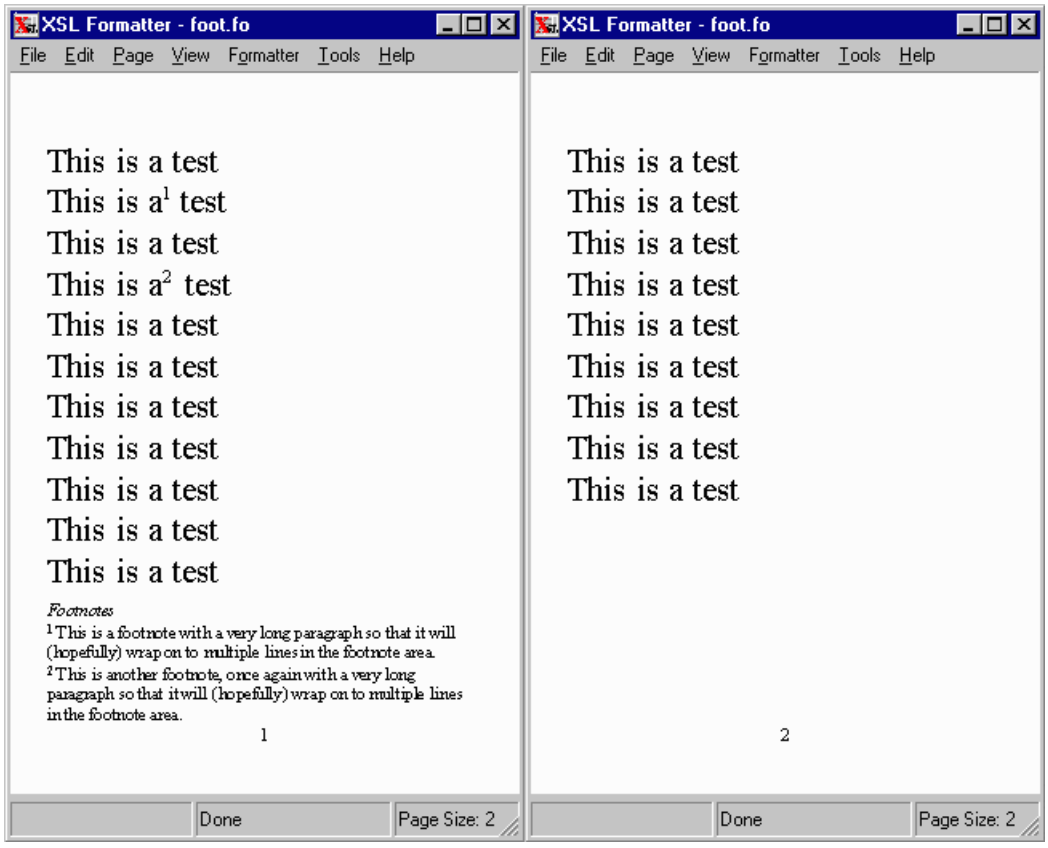
Other optional properties:

id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

Copyright © Crane Softwrights Ltd.

<footnote> Object (cont.)

The samp/foot.fo example:



Copyright © Crane Softwrights Ltd.

<footnote> Object (cont.)

An excerpt from the samp/foot.fo sample:

```
<page-sequence master-reference="frame">
 <static-content flow-name="xsl-footnote-separator">
   <block font-style="italic">Footnotes</block>
 </static-content>

<flow flow-name="frame-body" font-size="40pt">
<block>This is a test</block>
<block>This is a<footnote>
<inline baseline-shift="15pt" font-size="20pt">1</inline>
<footnote-body>
  <block font-size="20pt">
    <inline baseline-shift="5pt" font-size="15pt"
      >1 </inline>This is a footnote with a very
long paragraph so that it will (hopefully) wrap on to
multiple lines in the footnote area.
  </block>
</footnote-body>
</footnote> test</block>
<block>This is a test</block>
<block>This is a<footnote>
<inline baseline-shift="15pt" font-size="20pt">2</inline>
<footnote-body>
  <block font-size="20pt">
    <inline baseline-shift="5pt" font-size="15pt"
      >2 </inline>This is another footnote, once
again with a very long paragraph so that it will
(hopefully) wrap on to multiple lines in the footnote area.
  </block>
</footnote-body>
</footnote> test</block>
<block>This is a test</block>
<block>This is a test</block>
...
```

Of note:

- the footnote definitions are embedded in the blocks at the point where the footnote citation is rendered
- the footnote citation in the block is formatted differently than the footnote citation echoed in the footnote-reference-area

Copyright © Crane Softwrights Ltd.

`<footnote-body>` Object

Chapter 7 - Floats, footnotes and containers
Section 3 - Footnotes

Purpose:

- the portion of footnote content rendered towards the after-edge of the body region

Content:

- (6.12.4) (`%block;`)+
- Child object:
  - `%block;`(6.2;71)
- Referring object:
  - `<footnote>`(6.12.3;219)

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

`id=`(7.30.8;475)
`index-class=`(7.24.1;475)
`index-key=`(7.24.2;475)

Copyright © Crane Softwrights Ltd.

<footnote-body> Object (cont.)

An excerpt from the samp/foot.fo sample <footnote> Object (page 220):

```
<page-sequence master-reference="frame">
 <static-content flow-name="xsl-footnote-separator">
   <block font-style="italic">Footnotes</block>
 </static-content>

<flow flow-name="frame-body" font-size="40pt">
<block>This is a test</block>
<block>This is a<footnote>
<inline baseline-shift="15pt" font-size="20pt">1</inline>
<footnote-body>
  <block font-size="20pt">
    <inline baseline-shift="5pt" font-size="15pt"
      >1 </inline>This is a footnote with a very
long paragraph so that it will (hopefully) wrap on to
multiple lines in the footnote area.
  </block>
</footnote-body>
</footnote> test</block>
<block>This is a test</block>
<block>This is a<footnote>
<inline baseline-shift="15pt" font-size="20pt">2 </inline>
<footnote-body>
  <block font-size="20pt">
    <inline baseline-shift="5pt" font-size="15pt"
      >2 </inline>This is another footnote, once
again with a very long paragraph so that it will
(hopefully) wrap on to multiple lines in the footnote area.
  </block>
</footnote-body>
</footnote> test</block>
<block>This is a test</block>
<block>This is a test</block>
...
```


Copyright © Crane Softwrights Ltd.

Containers

Containers can be used to introduce new reference areas in their context

- reference properties can only be changed for reference areas
  - e.g. `reference-orientation=` and `writing-mode=`
- useful for temporarily changing the direction of text on the rendered result
- areas created by container objects can alter their behavior to meet specific requirements that differ from the parent area
  - can specify an absolute position outside of the parent for a block container
  - can specify an overflow behavior
  - can specify a different writing direction or reference orientation
  - can specify minimum and maximum sizes to accept flow
- both block-level and inline-level container objects are available to be used within their respective types of parent objects

Container objects are named by where they are used

- `<block-container>` either inside or between blocks
  - an absolutely-positioned block-container is displayed out-of-line
    - does not break the block progression direction
  - other block-containers stay in the flow
    - breaks the block progression direction
- `<inline-container>` can only be used on a line inside of a block
  - note that by default an inline container sits on top of the dominant baseline
  - use `alignment-baseline="after-edge"` for the container to sit on the same after edge as the text

 Copyright © Crane Softwrights Ltd.

Containers (cont.)

Dimensions may be specified for both the inline-progression-direction and block-progression-direction

- not specifying a block-progression-dimension maximum allows the block to grow
  - `block-progression-dimension=` sets minimum and maximum
  - `block-progression-dimension.minimum=` sets minimum only
- specifying a block-progression-dimension allocates the specified amount of area on the page between sibling

Positioned in the flow to allocate a specified amount of real-estate on the page between sibling block-level constructs

- can specify different behaviors for overflow situations
  - automatic handling (default) is processor dependent based on the medium
  - can be visible beyond the pre-allocated area (but still within the medium)
  - can be hidden (clipped) without error
  - can be hidden (clipped) with an error reported to the operator
  - can be captured within a scrolling mechanism for interactive media

Copyright © Crane Softwrights Ltd.

Block containers

Chapter 7 - Floats, footnotes and containers
Section 4 - Container basics

May be positioned in the flow of neighboring blocks or positioned out of line of neighboring blocks

- must contain block-level constructs
- can be relatively positioned between sibling block-level constructs (default)
- can be positioned at a given location
    - `absolute-position="absolute"`
        - offset from the containing area
    - `absolute-position="fixed"`
        - offset from the medium
    - out-of-line position is indicated by specifying the space offsets
        - `top=`, `bottom=`, `left=` and `right=`
        - distance from the edge of the area from which it is offset

May underlie or overlay other areas on the page by using `z-index=`

- all content by default at index level 0
- items with a higher index value are "on top" of items with a lower index value
- can achieve "exclusive rendering" effects when using opaque backgrounds

 Copyright © Crane Softwrights Ltd.

<block-container> Object

Purpose:

- the specification of a block-level reference area for contained descendant blocks

Content:

- (6.5.3) (%block;)+
- Child object:
  - %block;(6.2;71)
- if not absolutely positioned, then may begin with any number of <marker> children

Property sets:

- Common Absolute Position Properties(7.6;425)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Block(7.11;432)

Other optional properties:

block-progression-dimension=(7.15.3;453)
break-after=(7.20.1;463)
break-before=(7.20.2;463)
clear=(7.19.1;465)
clip=(7.21.1;465)
display-align=(7.14.4;468)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
intrusion-displace=(7.19.3;477)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
overflow=(7.21.2;484)
reference-orientation=(7.21.3;491)
span=(7.21.4;497)
width=(7.15.14;504)
writing-mode=(7.29.7;504)
z-index=(7.30.18;505)

Shorthands influencing the above properties:

page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)

Copyright © Crane Softwrights Ltd.

<inline-container> Object

Purpose:

- the specification of an inline-level reference area for contained descendant blocks
    - used to place block-oriented constructs inline

Content:

- (6.6.8) (%block;)+
- Child object:
    - %block;(6.2;71)
- may begin with any number of <marker> children

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
block-progression-dimension=(7.15.3;453)
clip=(7.21.1;465)
display-align=(7.14.4;468)
dominant-baseline=(7.14.5;468)
height=(7.15.6;474)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
inline-progression-dimension=(7.15.7;476)
keep-together=(7.20.3;477)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
line-height=(7.16.4;479)
overflow=(7.21.2;484)
reference-orientation=(7.21.3;491)
width=(7.15.14;504)
writing-mode=(7.29.7;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
page-break-inside=(7.31.18;487)
vertical-align=(7.31.22;502)

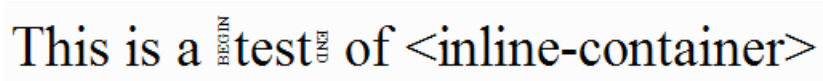
 Copyright © Crane Softwrights Ltd.

`<inline-container>` Object (cont.)

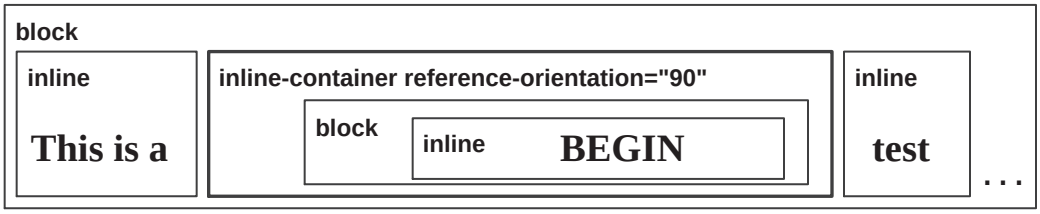
Must be positioned in the flow of inline constructs

- must contain block-level constructs

Consider the `samp/refornt.fo` example:



The nesting of the constructs used to create the example is as follows:




```
01  <block font-size="40pt">This is a
02    <inline-container reference-orientation="90">
03      <block font-size="12pt">BEGIN</block>
04    </inline-container>test<inline-container
05                        reference-orientation="270">
06      <block font-size="12pt">END</block>
07    </inline-container>
08    of &lt;inline-container>
09  </block>
```

Of note:

- to prevent a break in the line, an `<inline-container>` is used in the text
- the container's reference orientation is changed by 90 degrees counterclockwise
- the container only contains block-level constructs, so a block is used to contain the text
- the text is in an inline construct (though in this particular example this is redundant)

Copyright © Crane Softwrights Ltd.

Chapter 8 - Flows, static content and page geometry sequencing

- Introduction - Extended page model for pagination
- Section 1 - Page regions, headers, and footers
- Section 2 - Content definition
- Section 3 - Page Sequence Master Interleave (PSMI)
- Section 4 - Page geometry sequencing

Outcomes:

- understand basic concepts of pagination constructs

 Copyright © Crane Softwrights Ltd.

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)

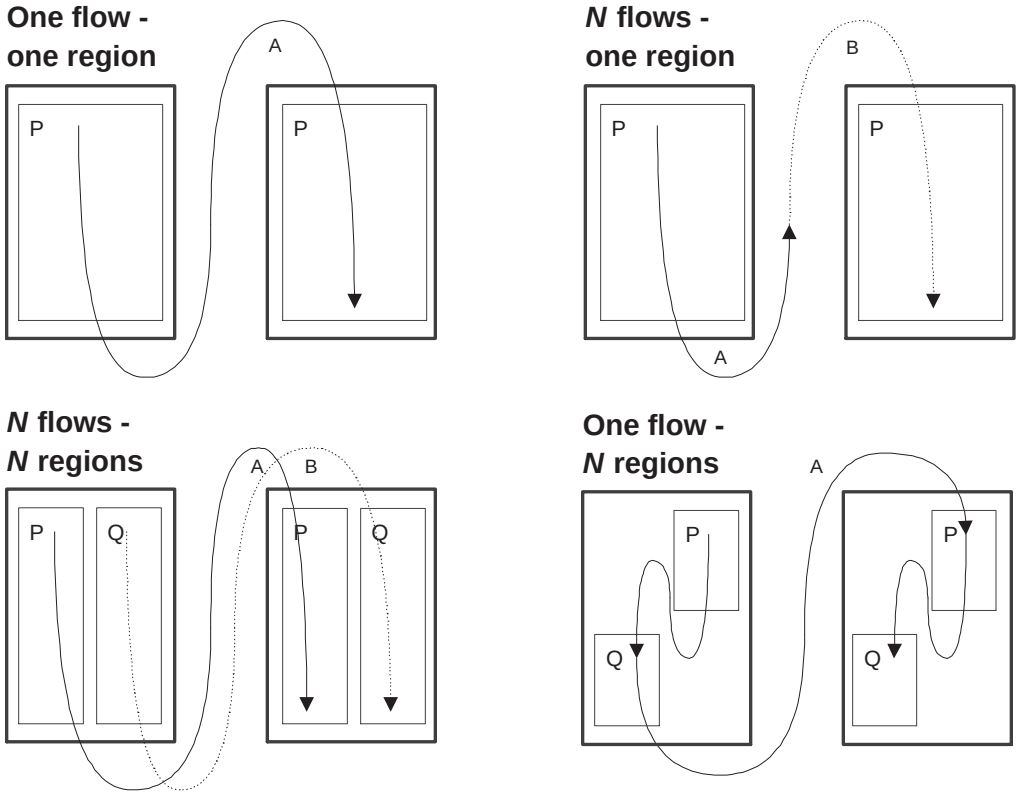
- pagination differentiates XSL-FO semantics from web browser display semantics

1.0 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

1.1 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



Copyright © Crane Softwrights Ltd.

Extended page model for pagination (cont.)

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

 Copyright © Crane Softwrights Ltd.

Extended page model for pagination (cont.)

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

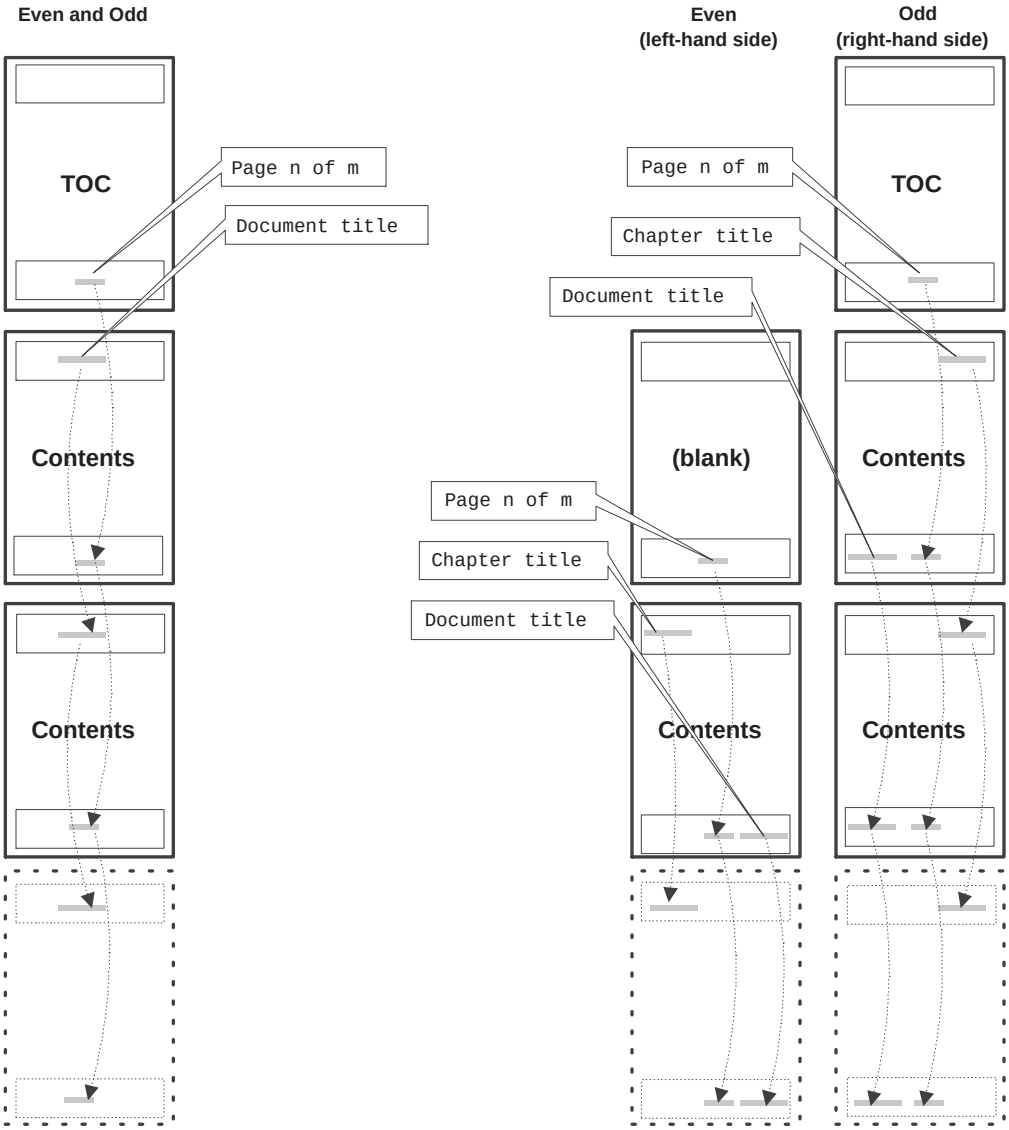
- the stylesheet writer choreographs the page geometries by the anticipated needs for the quantity of flow expected to be formatted
- all static content definitions needed for a page sequence must be present in that sequence
    - often requires stylesheet writing techniques that copy information repeatedly

Copyright © Crane Softwrights Ltd.

Extended page model for pagination (cont.)

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



Copyright © Crane Softwrights Ltd.

Extended page model for pagination (cont.)

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
- 1.1 `<folio-prefix>` (6.6.13)
  - a sequence of text defined to prefix page numbers of a page sequence
- 1.1 `<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
- 1.1 `<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

Copyright © Crane Softwrights Ltd.

Extended page model for pagination (cont.)

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

 Copyright © Crane Softwrights Ltd.

Region dimensions

Neither the page nor regions have border or padding rectangles

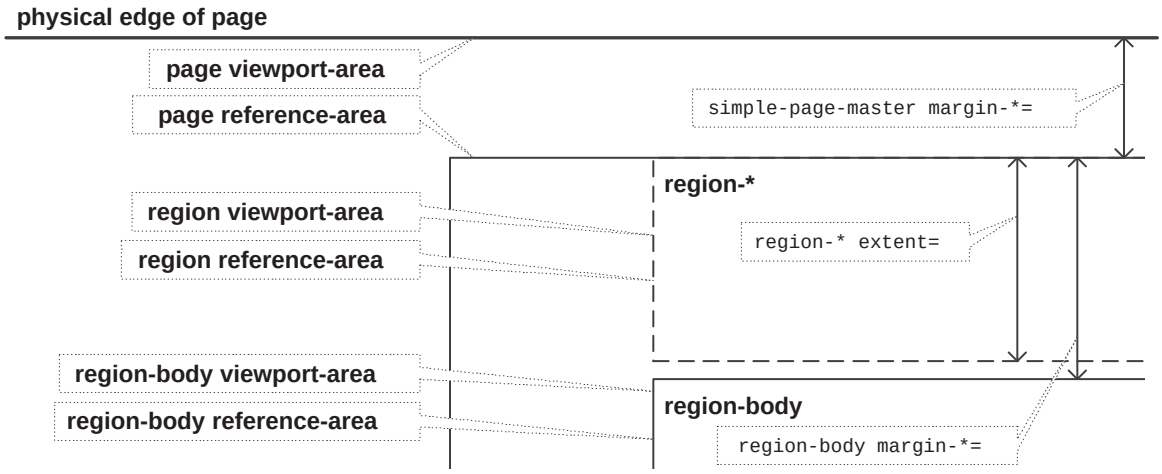
- may in future versions of XSL-FO have border and padding

Page definition is slightly different than region definitions

- the page reference edges are top, bottom, left and right relative to the physical medium
  - region reference edges are relative to the reference orientation
- the page's viewport-area is larger than the page's reference-area
  - a region's viewport-area is equal to the region's reference-area

The perimeter region incursions must be accommodated explicitly by the body region

- the incursion is illustrated in Page geometry (page 79)
  - precedence determines which perimeter region occupies the before or after pairs of corners of a page
  - note the default is to keep before/after region boundaries between start/end regions
- must move border of body region within extents of perimeter regions for border not to interfere
  - this is not automatic and will result in overwritten content if not accommodated



Copyright © Crane Softwrights Ltd.

<region-before> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 1 - Page regions, headers, and footers

Purpose:

- the definition of the perimeter area whose before edge is co-incident with the before edge of the page's content rectangle

Content:

- (6.4.15) EMPTY
- Referring object:
  - <simple-page-master>(6.4.13;119)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)

Other required property:
region-name=(7.27.17;491)

Other optional properties:
clip=(7.21.1;465)
display-align=(7.14.4;468)
extent=(7.27.4;469)
overflow=(7.21.2;484)
precedence=(7.27.16;490)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Properties of note:

- the region-name= property is required but the default name of "xsl-region-before" is used as this required property if a name is not supplied in the XSL-FO instance
- precedence= defaults to "false" indicating the before region does not override the start and end regions, but can be changed to "true" to stretch the region to the content rectangle of the page reference area
- even though padding= and border-width= properties are indicated as available indirectly through the common property set, these values are fixed at "0pt" in XSL-FO 1.0
- display-align= is used to keep the information in the region snug against the before edge, snug against the after edge, or centered in the middle of the two in the block progression direction

The placement of this region is illustrated in Page geometry (page 79)

 Copyright © Crane Softwrights Ltd.

<region-after> Object

Purpose:

- the definition of the perimeter area whose after-edge is co-incident with the after-edge of the page's content rectangle

Content:

- (6.4.16) EMPTY
- Referring object:
  - <simple-page-master>(6.4.13;119)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)

Other required property:
region-name=(7.27.17;491)

Other optional properties:
clip=(7.21.1;465)
display-align=(7.14.4;468)
extent=(7.27.4;469)
overflow=(7.21.2;484)
precedence=(7.27.16;490)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Properties of note:

- the region-name= property is required but the default name of "xsl-region-after" is used as this required property if a name is not supplied in the XSL-FO instance
- precedence= defaults to "false" indicating the after region does not override the start and end regions, but can be changed to "true" to stretch the region to the content rectangle of the page reference area
- even though padding= and border-width= properties are indicated as available indirectly through the common property set, these values are fixed at "0pt" in XSL-FO 1.0
- display-align= is used to keep the information in the region snug against the before edge, snug against the after edge, or centered in the middle of the two in the block progression direction

The placement of this region is illustrated in Page geometry (page 79)

Copyright © Crane Softwrights Ltd.

<region-after> Object (cont.)

Excerpts from a draft of a training rendition of this material:

```
  <layout-master-set>
    <simple-page-master master-name="frame-left"
      page-height="11in" page-width="8.5in"
      margin-top=".45in - .3in" margin-bottom=".45in - .3in"
      margin-left=".6in" margin-right=".6in">
      <region-body region-name="pages-body"
        margin-top=".3in" margin-bottom=".3in"/>
      <region-before extent=".3in" region-name="pages-before"/>
      <region-after extent=".3in" region-name="pages-after-left"/>
    </simple-page-master>
    <simple-page-master master-name="frame-right"
      page-height="11in" page-width="8.5in"
      margin-top=".45in - .3in" margin-bottom=".45in - .3in"
      margin-left=".6in" margin-right=".6in">
      <region-body region-name="pages-body"
        margin-top=".3in" margin-bottom=".3in"/>
      <region-before extent=".3in" region-name="pages-before"/>
      <region-after extent=".3in" region-name="pages-after-right"/>
    </simple-page-master>
    ...
```

Note that perimeter regions are defined *after* the body region

- some processors accept this without error while others can properly report an error

 Copyright © Crane Softwrights Ltd.

<region-start> Object

Purpose:

- the definition of the perimeter area whose start-edge is co-incident with the start-edge of the page's content rectangle

Content:

- (6.4.17) EMPTY
- Referring object:
  - <simple-page-master>(6.4.13;119)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)

Other required property:
region-name=(7.27.17;491)

Other optional properties:
clip=(7.21.1;465)
display-align=(7.14.4;468)
extent=(7.27.4;469)
overflow=(7.21.2;484)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Properties of note:

- the region-name= property is required but the default name of "xsl-region-start" is used as this required property if a name is not supplied in the XSL-FO instance
- even though padding= and border-width= properties are indicated as available indirectly through the common property set, these values are fixed at "0pt" in XSL-FO 1.0

The placement of this region is illustrated in Page geometry (page 79)

Copyright © Crane Softwrights Ltd.

<region-end> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 1 - Page regions, headers, and footers

Purpose:

- the definition of the perimeter area whose end-edge is co-incident with the end-edge of the page's content rectangle

Content:

- (6.4.18) EMPTY
- Referring object:
  - <simple-page-master>(6.4.13;119)

Property sets:

- Common Border, Padding, and Background Properties(7.8;428)

Other required property:
region-name=(7.27.17;491)

Other optional properties:
clip=(7.21.1;465)
display-align=(7.14.4;468)
extent=(7.27.4;469)
overflow=(7.21.2;484)
reference-orientation=(7.21.3;491)
writing-mode=(7.29.7;504)

Properties of note:

- the region-name= property is required but the default name of "xsl-region-end" is used as this required property if a name is not supplied in the XSL-FO instance
- even though padding= and border-width= properties are indicated as available indirectly through the common property set, these values are fixed at "0pt" in XSL-FO 1.0

The placement of this region is illustrated in Page geometry (page 79)

 Copyright © Crane Softwrights Ltd.

Flowed content vs. static content

The paginated content and all static contents are assigned to flows

- not assigned to regions or region positions
- geometry defines at which position (if any) a named region is on the page
- the target of flowed or static content is specified by `flow-name=`
- regions are just well-defined areas of the page into which content can be placed
- regions have default names but can be renamed
    - the same names can be used for the same regions in different page definitions

Flows are named the same as the region name

- the content of pagination, headers, and footers is placed into regions by the region name

Flows can be mapped by flow name to a region by region name

- `<flow-map>` describes a mapping
- the content of pagination, headers, and footers is grouped in an arbitrary number of flows
- the selection of regions accepting a flow is arbitrarily ordered and grouped
- the process flows content into the different regions, moving to the next region available on the page when a given region on the page is full
- see Extended page model for pagination (page 231) for an illustration
- naming a flow that is not in the flow map is not an error
    - the content of the flow disappears
- naming a flow more than once in a flow map is an error
    - there is no automated duplication of flow content

Only flowed content triggers the pages to be produced from a `<page-sequence>`

- each `<page-sequence>` construct must have exactly one paginated flow specified
    - must indicate the region (by name) to which the content is targeted
- content fills a region (any region) until the region overflows
    - the `overflow=` property dictates whether another page is generated to create a new region
    - can choose to error on overflow
- a page geometry in the page sequence need not have the named region accepting the flow
    - such a page is produced using only static content for the regions named on the page definition
    - the next page is obtained from the page sequence and examined for the named region accepting the flow

Copyright © Crane Softwrights Ltd.

Flowed content vs. static content (cont.)

The formatter generates a page using the next page geometry in the page sequence

- the amount of the flowed content dictates how many pages get generated for the sequence
- the page geometry or pattern of page geometries dictate which regions are on each page generated to accommodate the paginated flow
- all of the static content assigned to named regions found in the geometry is rendered

Static content is repeated in the named regions that appear on each page generated

- `<static-content>`
- each `<page-sequence>` construct may have any number of static content specifications
  - must indicate the region (by name) to which the content is targeted
- it is common to include a `<page-number/>` object in the static content
  - cannot be placed in a flow; can only be placed in static content
  - generates a sequence of character areas reflecting the page number
  - the choice in characters used in the generated areas is governed by properties of the `<page-sequence>` object
    - formal definition of properties of page number characters comes from XSLT
    - 1 folio prefix and suffix included when citing and using page numbers
  - the generated areas inherit properties ancestral to the referencing object, not the referred object
  - page numbers cannot be used in any kind of arithmetic or conditional test

1 Each page sequence can have its page numbers prefixed and suffixed

- 1 `<folio-prefix>`
  - defines what precedes the rendering of a page number when displayed or cited
- 1 `<folio-suffix>`
  - defines what precedes the rendering of a page number when displayed or cited

Flowed and static content are defined on a page-sequence basis

- every page-sequence must have its own definitions of flowed content and static content
  - cannot point to another page-sequence's content
- `<flow>` and `<static-content>` constructs are children of the `<page-sequence>` construct

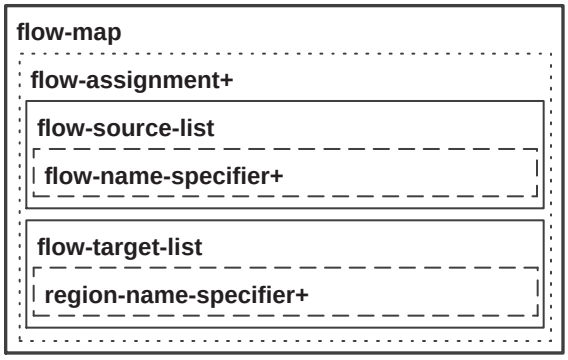
It is not an error when content is defined for a named region and that region is not present on the page being generated

- nothing is rendered from the definition of the flowed content
- if there is no region for the flowed content, the static content (if any) is rendered and a new page is generated
- e.g. consider the example where one interleaves a page of normally flowed content with a page entirely made of static content of a set of ruled lines: the result is in a workbook-like format where the page opposite of the material is used for keeping notes

 Copyright © Crane Softwrights Ltd.

Flow maps

- `<flow-map>` is a very simple structure
  - maps flows by flow name to regions by region name
  - stored as a child of `<layout-master-set>`
  - when needed it is named in `<page-sequence>`



For example, an excerpt from `flows.fo` with the flows illustrated on page 229:

```
01    <flow-map flow-map-name="n-flows-n-regions">
02      <flow-assignment>
03        <flow-source-list>
04          <flow-name-specifier flow-name-reference="A"/>
05        </flow-source-list>
06        <flow-target-list>
07          <region-name-specifier region-name-reference="P"/>
08        </flow-target-list>
09      </flow-assignment>
10      <flow-assignment>
11        <flow-source-list>
12          <flow-name-specifier flow-name-reference="B"/>
13        </flow-source-list>
14        <flow-target-list>
15          <region-name-specifier region-name-reference="Q"/>
16        </flow-target-list>
17      </flow-assignment>
18    </flow-map>
```

  - note that the regions in the `flows.fo` test file are not positioned as illustrated for the "one flow - n regions" example, but any positioning is acceptable.

When a `<page-sequence>` does not specify a flow map, an implicit flow map is used:

- every flow name is implicitly matched with the region of the same name

Copyright © Crane Softwrights Ltd.

<flow-map> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 2 - Content definition

Purpose:

- specifies a collection of assignments of a set of named flows to a set of named regions

Content:

- 1 (6.4.22) (flow-assignment+)
- Child object:
  - 1 <flow-assignment>(6.4.23;247)
- Referring object:
  - <layout-master-set>(6.4.7;63)

Required property:

1 flow-map-name=(7.27.18;470)

Copyright © Crane Softwrights Ltd.

`<flow-assignment>` Object

Purpose:

- specifies a single assignment of a set of named flows to a set of named regions

Content:

- 1 (6.4.23) (`flow-source-list`,`flow-target-list`)
- Child objects (alphabetical):
  - 1 `<flow-source-list>`(6.4.24;248)
  - 1 `<flow-target-list>`(6.4.26;250)
- Referring object:
  - 1 `<flow-map>`(6.4.22;246)

No properties are defined for this formatting object.

Copyright © Crane Softwrights Ltd.

`<flow-source-list>` Object

Purpose:

- specifies a set of named flows in a single flow assignment

Content:

- 1 (6.4.24) (`flow-name-specifier+`)
- Child object:
  - 1 `<flow-name-specifier>`(6.4.25;249)
- Referring object:
  - 1 `<flow-assignment>`(6.4.23;247)

No properties are defined for this formatting object.

 Copyright © Crane Softwrights Ltd.

<flow-name-specifier> Object

Purpose:

- specifies a single named flow in a set of named flows

Content:

- 1 (6.4.25) EMPTY
- Referring object:
  - 1 <flow-source-list>(6.4.24;248)

Required property:

1 flow-name-reference=(7.27.20;470)

Copyright © Crane Softwrights Ltd.

`<flow-target-list>` Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 2 - Content definition

Purpose:

- specifies a set of named regions in a single flow assignment

Content:

- 1 (6.4.26) (`region-name-specifier+`)
- Child object:
  - 1 `<region-name-specifier>`(6.4.27;251)
- Referring object:
  - 1 `<flow-assignment>`(6.4.23;247)

No properties are defined for this formatting object.

Copyright © Crane Softwrights Ltd.

<region-name-specifier> Object

Purpose:

- specifies a single named region in a set of named regions

Content:

- 1 (6.4.27) EMPTY
- Referring object:
  - 1 <flow-target-list>(6.4.26;250)

Required property:

1 region-name-reference=(7.27.21;491)

Copyright © Crane Softwrights Ltd.

Headers and footers

It is a common requirement for different pages to have different static content

- consider the need for rendering page numbers on alternate sides of the footer
  - on the right side of odd pages
  - on the left side of even pages
- use different region names for the same regions on different pages
  - the odd page `<region-after>` constructs could be named "`footer-odd`"
  - the even page `<region-after>` constructs could be named "`footer-even`"
  - the page sequence would then define two different `<static-content>` constructs
    - the construct flowed to the region named "`footer-odd`" would place the page number citation on the right
    - the construct flowed to the region named "`footer-even`" would place the page number citation on the left

It is a common requirement for different pages to have the same content in different places

- consider the need for rendering information in the outside edge of all pages
  - the same content but alternating the right side on odd pages and the left side on even pages
- cannot use the same region name for different regions on different pages
  - each time you use a given custom name in the set of page geometries, it must always be for the same body or perimeter region wherever else that name is used in other page geometries
- must duplicate the content if there are two differently named regions in two page geometries in order to obtain identical results
  - the odd page `<region-end>` construct could be named "`outside-odd`"
  - the even page `<region-start>` construct could be named "`outside-even`"
  - the page sequence would define two identical `<static-content>` constructs to get the same appearance in both regions
  - the same results would be better obtained by using the same region names in the two geometries

It is the stylesheet writer's responsibility to supply in each page sequence the definitions of all content desired for the possible regions triggered by the sequence of pages

- stylesheets are often repetitive in order to satisfy this requirement

 Copyright © Crane Softwrights Ltd.

<static-content> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 2 - Content definition

Purpose:

- the definition of content that is primarily unchanged from page to page in a page sequence

Content:

- (6.4.20) (%block;)+
- Child object:
  - %block;(6.2;71)
- Referring object:
  - <page-sequence>(6.4.5;65)

Required property:
flow-name=(7.27.5;470)

Optional properties:
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

<static-content> Object (cont.)

Excerpts from a draft of a training rendition of this material:

```
  <page-sequence id="fo_region-before" master-reference="frames">
    <static-content flow-name="pages-before" font-style="italic">
      <block text-align="center">Practical Formatting
Using XSL-FO</block></static-content>
    <static-content flow-name="pages-after-right"
                    font-style="italic" font-size="9pt">
      <table>...<block text-align="start">Slide 173 of
287 <inline font-size="8pt">&lt;frame_static-content&gt;</inline
></block>...<block text-align="center">
                <inline font-size="8pt" font-style="normal"
>Information subject to restrictive legend on title page.</inline>
              </block>...
              <block text-align="end">Page <page-number/> of
<page-number-citation ref-id="N0"/></block>...
      </table></static-content>
    <static-content flow-name="pages-after-left"
                    font-style="italic" font-size="9pt">
      <table>...<block text-align="start">Page <page-number/> of
<page-number-citation ref-id="N0"/></block>...
                <block text-align="center">
                  <inline font-size="8pt" font-style="normal"
>Information subject to restrictive legend on title page.</inline>
                </block>...
                <block text-align="end"><inline
font-size="8pt">&lt;frame_static-content&gt;</inline> Slide 173 of
287</block>...</table></static-content>
    <flow>...
```

Of note:

- the header is a single centered title
- two footers are defined, one for each of the right-hand and left-hand pages
    - the right-hand footer has the page number on the right hand side of the page
    - the left-hand footer has the page number on the left hand side of the page
- each footer is a table of three columns
    - the start-side column is aligned to the start edge
    - the center column is aligned to the center
    - the end-side column is aligned to the end edge

 Copyright © Crane Softwrights Ltd.

<page-number> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 2 - Content definition

Purpose:

- an inline-level placeholder replaced with the page number of the current page
- there are no methods of incorporating a formatted page number in arithmetic of any kind or explicitly in any kind of conditional test

Content:

- (6.6.10) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other optional properties:

alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
dominant-baseline=(7.14.5;468)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
score-spaces=(7.30.15;495)
text-altitude=(7.29.4;500)
text-decoration=(7.17.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
text-transform=(7.17.6;501)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)
wrap-option=(7.16.13;504)

Shorthands influencing the above properties:

font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Of note:

- the formatter supplies to the flow the characters as <character> objects
  - the characters used by the formatter are specified by the format= property of the <page-sequence> for the page

Copyright © Crane Softwrights Ltd.

<folio-prefix> Object

Purpose:

- describe the content flowed immediately preceding the characters of a page number in the page sequence
- also used by citations into the page sequence

Content:

- (6.6.13) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - <page-sequence>(6.4.5;65)

No properties are defined for this formatting object.

Child of a <page-sequence> formatting object

 Copyright © Crane Softwrights Ltd.

<folio-prefix> Object (cont.)

An example from the XSL-FO specification:

```
<fo:page-sequence id="glossary" initial-page-number="1">
  <fo:page-number-folio-prefix>
    <fo:inline>G-</fo:inline>
  </fo:page-number-folio-prefix>
  <fo:flow>...</fo:flow>
</fo:page-sequence>
```


Copyright © Crane Softwrights Ltd.

<folio-suffix> Object

Purpose:

- describe the content flowed immediately following the characters of a page number in the page sequence
- also used by citations into the page sequence

Content:

- 1 (6.6.14) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - <page-sequence>(6.4.5;65)

No properties are defined for this formatting object.

Child of a <page-sequence> formatting object

 Copyright © Crane Softwrights Ltd.

Dynamic content in static content

It is often necessary to redefine header information more often than once per page sequence

- starting a new page sequence starts a new page
- a page sequence of chapter content may require section headings to change when section breaks do not start a new page
- "dictionary heads" are used in a dictionary presentation of many entries
    - the top left of the left-page header indicates the first word found on the page
    - the top right of the right-page header indicates the last word found on the page
    - the formatter must be told which of the possibilities is desired for choosing the first and last entry on the page
        - is it the first or last definition that begins on the page?
        - is it the first or last definition where any part of the definition is on the page?

Dynamic content that is a candidate for inclusion in static content is captured in the flow in a marker using the `<marker>` object

- the marker's parent's total area is called a "qualifying area" for the marker
    - the marker is associated with all of the non-normal areas in the qualifying area
- the "containing page" is that page with the first of the areas that would have been rendered for the marker if the marker were rendered in place
    - marker descendants do not inherit inheritable properties of the marker or its ancestors
    - marker descendants do inherit inheritable properties of where the marker is retrieved
- two markers with the same parent object must not have the same `marker-class-name=` property
    - all markers must be the first children of its parent object

Necessary to have canvas information duplicated to be retrieved

- the marker definition is not rendered on the canvas, only the retrieval
- if the static content needs to be the same as the canvas content, the content must be duplicated in the flow

Preference is afforded to the parents of nested markers based on their position in the area tree hierarchy

- areas higher in the tree (pre-order traversal) are considered preferred over areas lower in the tree
- higher preference is given to a page than to the page that precedes it

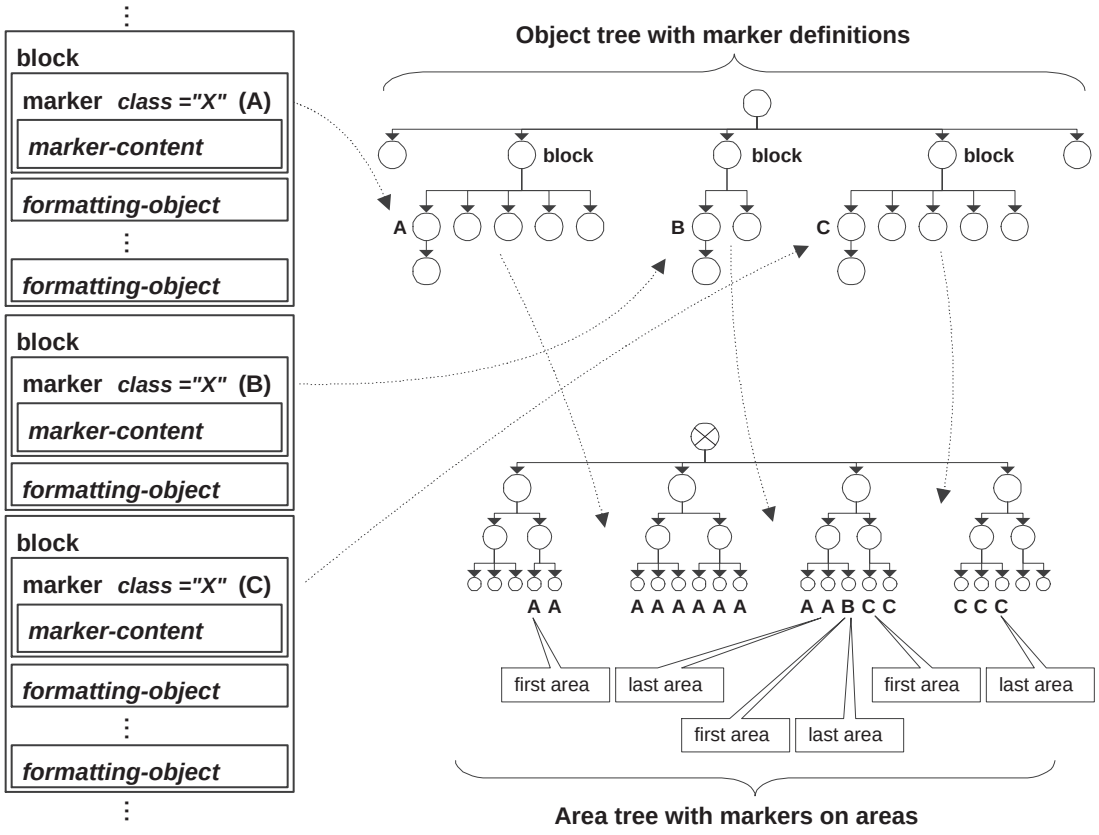
Copyright © Crane Softwrights Ltd.

Dynamic content in static content (cont.)

Areas in area tree have associated markers for each marker class

- the retrieval algorithm notes the first, last or any areas of a marker on a page
- a page may validly not contain any first or last area for a given marker class

Consider this figure that shows four pages generated in the area tree



- the first page has only one first or last area associated with any marker
  - "A" is the first and last marker whose first area is on the page
- the second page doesn't have the first or last area associated with any marker
  - "A" is the only marker with any area on the page
- the third page has a number of such areas
  - "A" is the first marker where any of its areas is on the page
  - "B" is the first marker whose first area is on the page
  - "B" is the last marker whose last area is on the page
  - "C" is the last marker whose first area is on the page
- the fourth page has only one first or last area associated with any marker
  - "C" is the first and last marker whose last area or any area is on the page

 Copyright © Crane Softwrights Ltd.

Dynamic content in static content (cont.)

Only static content may contain `<retrieve-marker/>` constructs

- references the marker named by the `retrieve-class-name=` property
  - the marker's content is added to the static content in place of the retrieval construct

Dynamic content inherits properties ancestral to the referencing `<retrieve-marker>` construct

- inherits properties as if the formatting objects of the marker content existed at the retrieval point

Can scope the search for a marker to be retrieved by using `retrieve-boundary=` property

- looking only on the page
- looking either on the page or on an earlier page in the same page sequence (default)
- looking either on the page or on any earlier page in the document

Can ask for the areas of a particular marker based on the marker's position by using `retrieve-position=` property

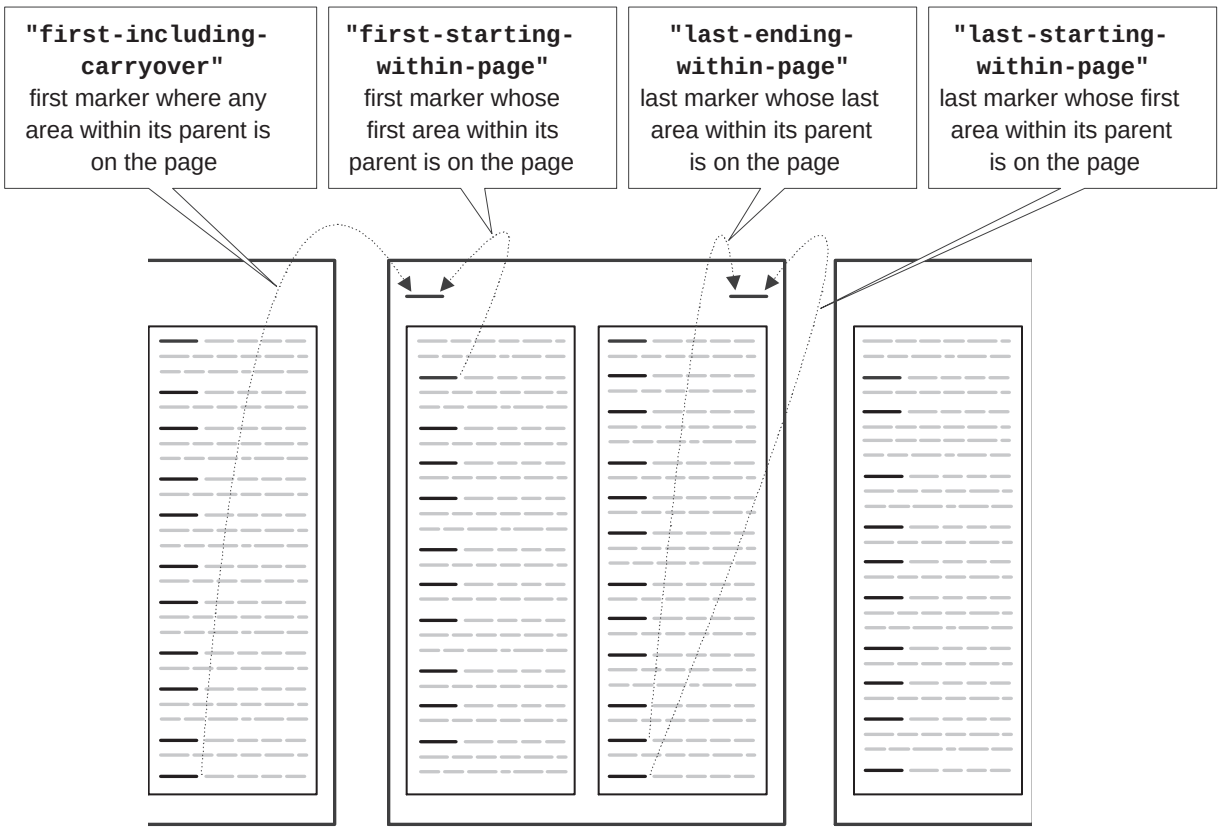
- first marker where any area within its parent is in the scope
- first marker whose first area within its parent is in the scope (default)
- last marker whose last area within its parent is in the scope
- last marker whose first area within its parent is in the scope

Copyright © Crane Softwrights Ltd.

Dynamic content in static content (cont.)

Chapter 8 - Flows, static content and page geometry sequencing
Section 2 - Content definition

Consider the production of "dictionary heads" where every entry in the dictionary includes a `<marker>` child and the header reflects the first and last entry on the page for navigation purposes:



Recall the identification of first and last qualifying areas for a given marker shown on page 260.

Copyright © Crane Softwrights Ltd.

Dynamic content in static content (cont.)

- Only table headers and footers may contain `<retrieve-table-marker>` constructs
  - alternatively, a header or footer itself may be replaced by a marker retrieval
    - this isn't true for perimeter regions

Can scope the search for a marker to be retrieved by using `retrieve-boundary-within-table=`

- looking within the table fragment with the marker on the given page
- looking to the start of the table (default)
- looking within all table fragments on the given page

Can ask for the areas of a particular marker based on the marker's position by using `retrieve-position-within-table=` property

- these are the same concepts and effects as happens with page marker retrievals
  - see Dynamic content in static content (page 262)
  - the table fragments are considered in the same way the pages are depicted
- first marker where any area within its parent is in the scope
- first marker whose first area within its parent is in the scope (default)
- last marker whose last area within its parent is in the scope
- last marker whose first area within its parent is in the scope

Copyright © Crane Softwrights Ltd.

<marker> Object

Purpose:

- the replacement formatting object content for a marker retrieved in static content
- associated with all of the areas generated by its parent formatting object

Content:

- (6.13.5) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- content validity dependent on the context of the corresponding <retrieve-marker> object the <marker> object replaces

Required property:
marker-class-name=(7.25.1;481)

 Copyright © Crane Softwrights Ltd.

<marker> Object (cont.)

An excerpt from the samp/marker.fo sample:

```
<page-sequence master-reference="frame">
  <static-content flow-name="frame-before">
    <block text-align="end" ...>
      <retrieve-marker retrieve-class-name="section"/>
    </block>
  </static-content>
  <static-content flow-name="frame-after">
    <block text-align="end" font-style="italic" font-size="12pt">
      <retrieve-marker retrieve-class-name="continued"
                  retrieve-position="last-starting-within-page"/>
    </block>
  </static-content>
  ...
<flow flow-name="frame-body">
  ...
  <block>
    <marker marker-class-name="section">Section One - 1.</marker>
    <marker marker-class-name="continued">(continued...)</marker>
    <block>1. Section One</block>
    <block space-before="1em">This is a test</block>
    ...
    <block space-before="1em">This is a test</block>
  </block>
  <block>
    <marker marker-class-name="continued"></marker>
  </block>

  <block space-before="2em">
    <marker marker-class-name="section">Section Two - 2.</marker>
    <marker marker-class-name="continued">(continued...)</marker>
    <block>2. Section Two</block>
    <block space-before="1em">This is a test</block>
    ...
    <block space-before="1em">This is a test</block>
  </block>
  <block>
    <marker marker-class-name="continued"></marker>
  </block>
```


Copyright © Crane Softwrights Ltd.

Of note:

- a surrounding block wraps all the blocks of the section for a section heading
    - the marker is the first child of the surrounding block, thus associating the marker with all of the contained blocks
    - the arrangement of the information in the marker can be (and typically would be) different than that used in the body of the page
- the continuing marker is always retrieved but is defined to be empty at the end of the section

 Copyright © Crane Softwrights Ltd.

<retrieve-marker> Object

Purpose:

- an inline-level placeholder replaced with the formatting objects of the indicated marker
- only allowed as a descendant of static content

Content:

- (6.13.6) EMPTY

Required property:
retrieve-class-name=(7.25.3;493)

Optional properties:
retrieve-boundary=(7.25.5;492) retrieve-position=(7.25.4;493)

Properties of note

- retrieve-position= indicates the marker desired by its qualifying areas (default is the first marker whose first qualifying area is on the page) as illustrated on page 260 and page 262
- retrieve-boundary= indicates limiting the search for a marker within the page, to the beginning of the page sequence (default), or to the beginning of the document

Copyright © Crane Softwrights Ltd.

<retrieve-marker> Object (cont.)

An excerpt from the samp/marker.fo sample:

```
<page-sequence master-reference="frame">
  <static-content flow-name="frame-before">
    <block text-align="end" font-weight="bold"
           color="silver" font-size="12pt">
      <retrieve-marker retrieve-class-name="section"/>
    </block>
  </static-content>
  <static-content flow-name="frame-after">
    <block text-align="end" font-style="italic" font-size="12pt">
      <retrieve-marker retrieve-class-name="continued"
                       retrieve-position="last-starting-within-page"/>
    </block>
  </static-content>
  ...
<flow flow-name="frame-body">
  ...
  <block>
    <marker marker-class-name="section">Section One - 1.</marker>
    <marker marker-class-name="continued">(continued...)</marker>
    <block>1. Section One</block>
    <block space-before="1em">This is a test</block>
    ...
    <block space-before="1em">This is a test</block>
  </block>
  <block>
    <marker marker-class-name="continued"></marker>
  </block>

  <block space-before="2em">
    <marker marker-class-name="section">Section Two - 2.</marker>
    <marker marker-class-name="continued">(continued...)</marker>
    <block>2. Section Two</block>
    <block space-before="1em">This is a test</block>
    ...
    <block space-before="1em">This is a test</block>
  </block>
  <block>
    <marker marker-class-name="continued"></marker>
  </block>
```

 Copyright © Crane Softwrights Ltd.

<retrieve-table-marker> Object

Purpose:

- an inline-level placeholder replaced with the formatting objects of the indicated marker
- only allowed as a replacement for or descendant of table headers and table footers

Content:

- (6.13.7) EMPTY

Required property:
retrieve-class-name=(7.25.3;493)

Optional properties:
retrieve-boundary-within-table=(7.25.2;492) retrieve-position-within-table=(7.25.6;493)

Properties of note

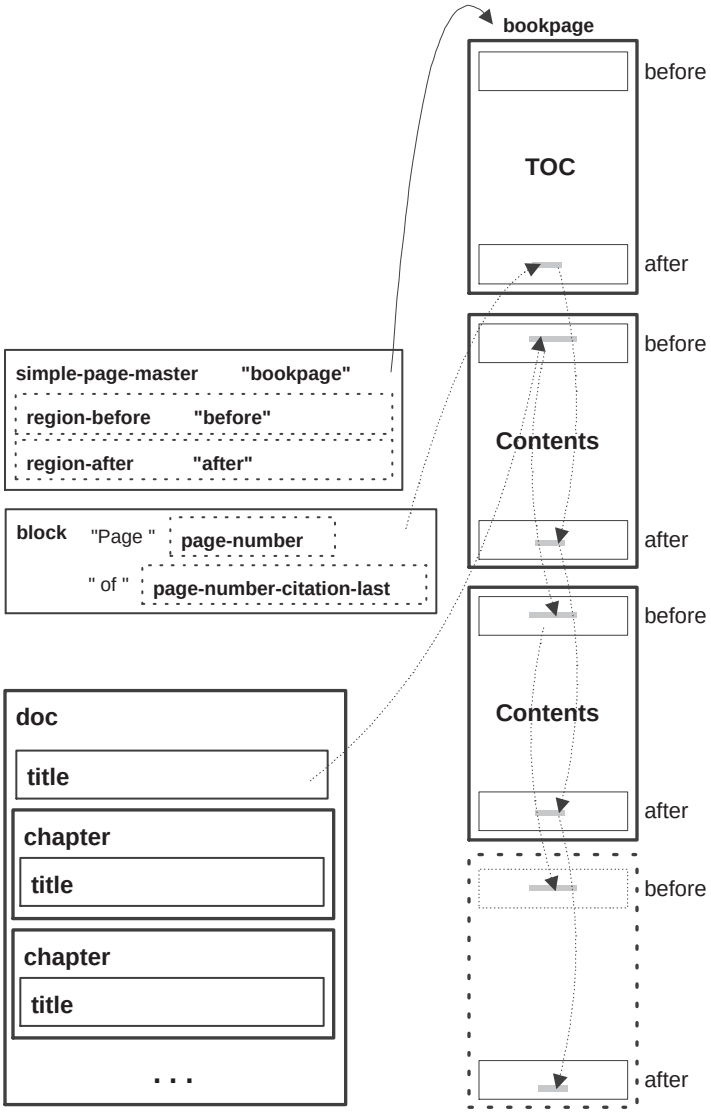
- retrieve-position-within-table= indicates the marker desired by its qualifying areas
- retrieve-boundary-within-table= indicates limiting the search for a marker within the table fragment, to the beginning of the table (default), or to only those on the page

Copyright © Crane Softwrights Ltd.

Planning a simple page sequence specification

Planning ahead can make page sequence specifying an easier task

- consider planning a book with a table of contents followed by content in separate page sequences each using the same page geometry for before and after regions
- a single-sided presentation with the page number at the bottom of all pages
    - static content indicating the current and last page numbers
- different formatting of headers for table of contents and for chapter contents
    - document title in the content page sequences but not in the TOC page sequence
        - the TOC page sequence has a different definition of static content than the content page sequences



 Copyright © Crane Softwrights Ltd.

Changing the page geometry based on authored content

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

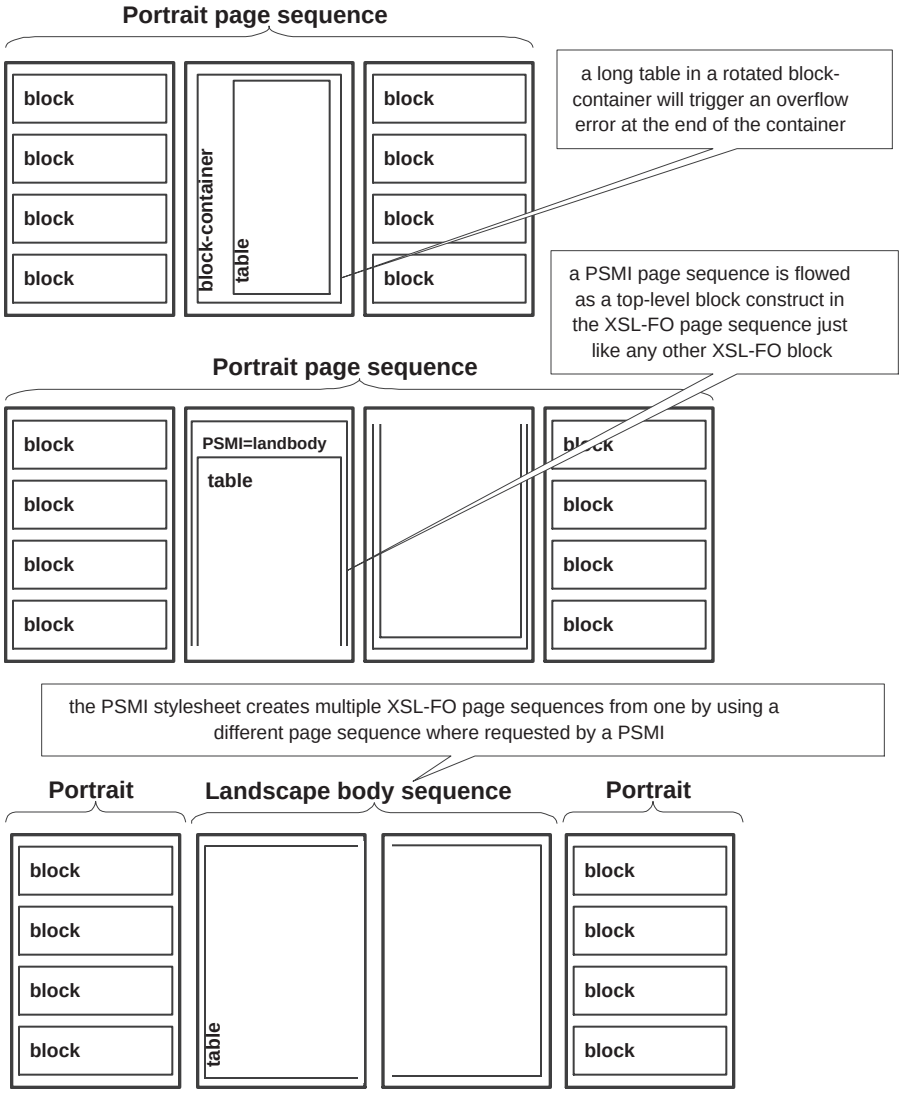
- public resource freely available from Crane Softwrights Ltd.
  - `http://www.CraneSoftwrights.com/links/res-pfux.htm`
- one-element vocabulary designed to express the semantic of this two-step intermediate process
- an XSLT stylesheet to implement the semantic for any XSL-FO+PSMI sequence

Copyright © Crane Softwrights Ltd.

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

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

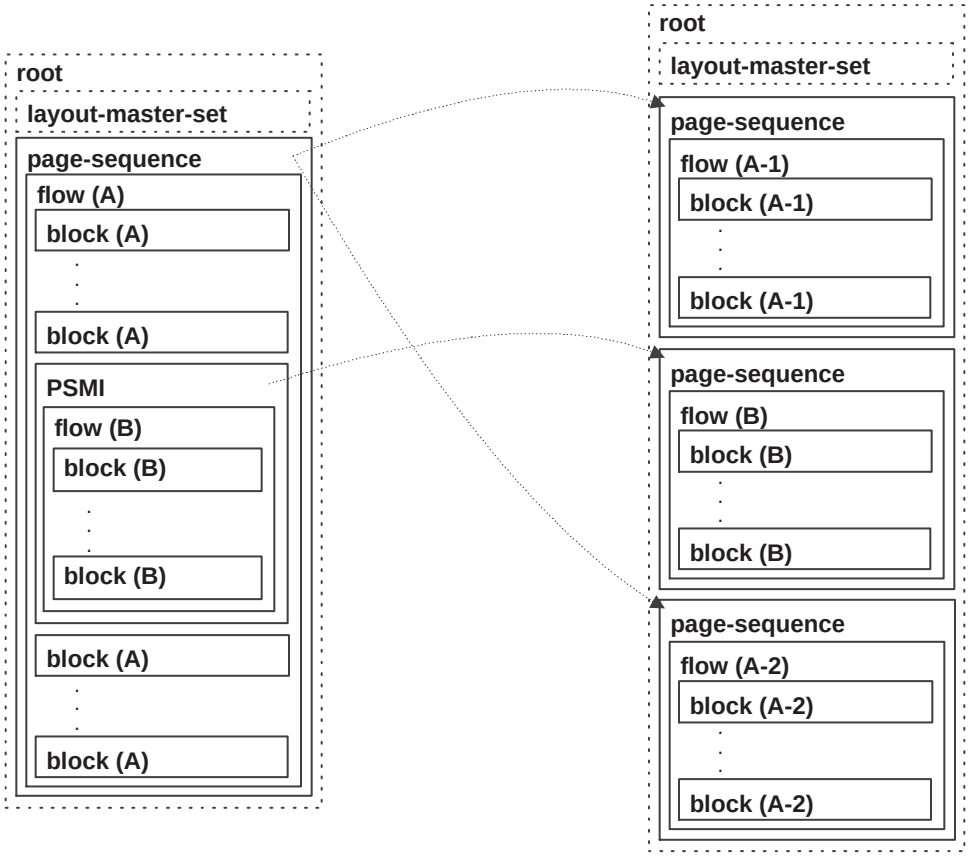


Copyright © Crane Softwrights Ltd.

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

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

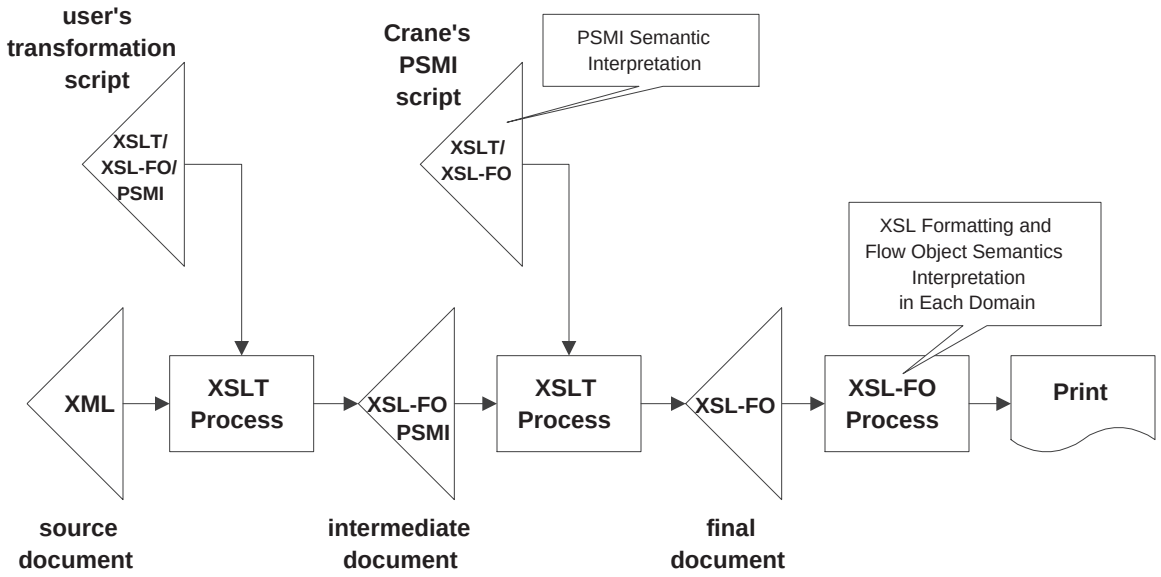


Copyright © Crane Softwrights Ltd.

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

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



Copyright © Crane Softwrights Ltd.

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

An example of flow implementing the change in page geometry:

```
    <simple-page-master master-name="frame-portraitbody" ...>
       <region-body region-name="frame-body" .../>
       <region-before region-name="frame-before" .../>
       <region-after region-name="frame-after" .../>
    </simple-page-master>
    <simple-page-master master-name="frame-landbody" ...>
       <region-body region-name="frame-body"
                    reference-orientation="90" .../>
       <region-before region-name="frame-before" .../>
       <region-after region-name="frame-after" .../>
    </simple-page-master>
...
<page-sequence master-reference="frame-portraitbody">
 <flow flow-name="frame-body">
  <block>Portrait information</block>
  ...
  <block>Next is landscaped</block>
  <psmi:page-sequence master-reference="frame-landbody"
   xmlns:psmi="http://www.CraneSoftwrights.com/resources/psmi">
   <flow flow-name="frame-body">
    <table>
     <table-body>
      <table-row>
       <table-cell border="solid">
        <block>This is a test</block>
        <block>This is a test</block>
        ...
        <block>This is a test</block>
        <block>This is a test</block>
       </table-cell>
      </table-row>
     </table-body>
    </table>
   </flow>
  </psmi:page-sequence>
  <block>Back to portrait</block>
  ...
 </flow>
```


Copyright © Crane Softwrights Ltd.

Changing the page geometry based on a pattern of geometries

Every different page geometry must be in a separate `<simple-page-master>` construct

- each master is uniquely named
- not all regions utilized in the page sequence need be defined on every page geometry

A `<page-sequence>` defines the flow for a sequence of pages

- can create a new page sequence for a change in static content
- can create a new page sequence to start the flow on a new page
- the page sequence utilizes page geometries referenced by the `master-reference=` property
- can force a sequence to create a blank page at end if necessary
    - so the page sequence automatically accommodates the first page number of the following page sequence (default)
        - the following page sequence may be forced to start on an odd or even page number thereby requiring a filler page between the end of the given page sequence
    - so the total number of pages in the page sequence is an even or odd count
    - so the last of the page sequence is an even or odd page number

The simplest case is not to sequence the pages

- a `<page-sequence>` that points to a `<simple-page-master>` repeats that one page for the entire sequence

 Copyright © Crane Softwrights Ltd.

Changing the page geometry based on a pattern of geometries (cont.)

A `<page-sequence-master>` defines an ordering of sub-sequences of page geometries

- a `<page-sequence>` that points to a `<page-sequence-master>` obtains each page geometry from the specified ordering
- the need for new pages in the flow takes the next page geometry from the next ordered sub-sequence in the master
  - each `<page-sequence-master>` must specify at least one sub-sequence of geometries
- the formatter may signal an error if it exhausts available sub-sequences
  - could otherwise just repeat the last sub-sequence
  - to prevent possible error a robust stylesheet should provide an infinitely repeatable sub-sequence as the last sub-sequence
- each master is uniquely named and is also unique from the page geometry names

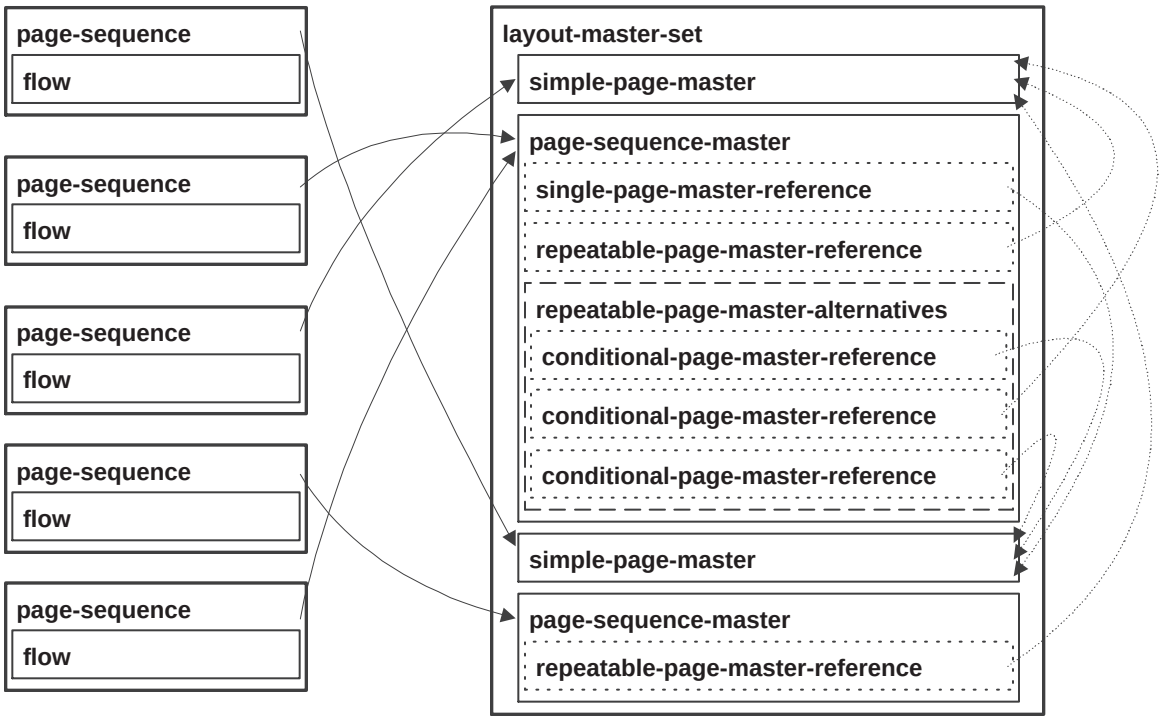
A single required `<layout-master-set>` specifies all masters

- both single-page masters and page-sequence masters
- there is no semantic order to the set of layout masters

Copyright © Crane Softwrights Ltd.

Overview of page sequence specifications

A complex (and contrived) page sequence could be as complex as depicted in this diagram:



Note there are only two specifications of actual page geometry, while many and various sequencing patterns of using that geometry are defined and utilized by the five page sequence specifications

- all subsequences are composed only of references to page geometries

 Copyright © Crane Softwrights Ltd.

Page geometry sub-sequences

A sub-sequence defines some or all of the page geometries used in a page sequence

- the first page of flow comes from the first qualifying geometry of the first sub-sequence of the page sequence
- when one sub-sequence is exhausted, the next page uses the first qualifying geometry from the next sub-sequence

A sub-sequence can specify the use of only one geometry

- use `<single-page-master-reference/>` to use one geometry once
- use `<repeatable-page-master-reference/>` to use one geometry more than once
    - can be unbounded (default) or bounded to a maximum repetition

A sub-sequence can specify the choice of one of a set of geometries

- use `<repeatable-page-master-alternatives>` to collect the set of geometries from which to choose
- use `<conditional-page-master-reference/>` to specify the criteria of a choice
- each criterion of a choice is tested in the document order of choices specified in the set
    - the first choice where all criteria test true specifies the geometry used for the page of the flow
- the set of choices is repeated indefinitely by default
    - can be bounded to a maximum repetition

Copyright © Crane Softwrights Ltd.

Page geometry sub-sequences (cont.)

The choice criteria is made up of three sub-conditions

- all three criteria must be true for the choice to be used
- an unspecified criterion is considered "`any`" which is true for each test
  - one need only specify the criteria desired for uniqueness
- all criteria values are unambiguous
  - it doesn't matter which order the criteria are specified
  - each page tests true for exactly one value for each criterion

A criterion can be based on the parity of the page number

- use `odd-or-even=` to test the criterion
- "`any`" (initial value) tests true for all pages
- "`odd`" tests true of the page number is odd
- "`even`" tests true if the page number is even

A criterion can be based on the position of the page within the page sequence

- use `page-position=` to test the criterion
- "`any`" (initial value) tests true for all pages
- "`first`" tests true if the page is the first in the page sequence
- "`last`" tests true if the page is the last in the page sequence
- "`rest`" tests true if the page is neither the first nor the last in the page sequence
- note that a one-page page sequence will test true for both values "`first`" and "`last`"
  - requires the stylesheet writer to decide which of the two possible geometries is desired for a one-page page-sequence
  - the tests in the alternatives must be ordered such that the desired geometry is selected for the one-page page sequence situation before the undesired geometry would be selected

A criterion can be based on the page being generated by the flow or not

- use `blank-or-not-blank=` to test the criterion
- "`any`" (initial value) tests true for all pages
- "`not-blank`" tests true if the page contains flow
- "`blank`" tests true if the page does not contain flow because of the page being generated to meet sequence conditions (i.e. a "forced" page)
  - e.g. ensuring the last page of a chapter is on an even page number

 Copyright © Crane Softwrights Ltd.

Forced blank pages

A forced blank page is a page not containing any paginated flow, but created by the formatter in response to page sequencing requirements

- referred to as a "blank page" in a page sequence
  - tested in `<conditional-page-master-reference>` using `blank-or-not-blank=`
- if no page master is supplied for a blank page, a page with no content is rendered
- always rendered as the last page in a page sequence

Two ways a blank page can be triggered by sequencing

- the page parity of the last page in the given page sequence
  - `force-page-count=`
  - all values except `no-force` can require a blank page to be created
  - note this is an extended conformance property and may not be available in all implementations
- the page parity of the first page of the following page sequence
  - a value of `force-page-count=="auto"` on the given page sequence will cause a final blank page to be created based on the parity of the page number of the first page of the following page sequence
  - `initial-page-number=` includes "`auto-odd`" and "`auto-even`"
  - the following page sequence value is tested for odd or even (including a specified value of "1") forcing the given page sequence to end on even or odd respectively
  - it is common that a blank page appears "unexpectedly" when the conditions of the following page sequence are the actual trigger
  - note this is a basic conformance property and must be available in all implementations

Four ways a blank page can be triggered by breaking to a new page

- using `break-before="odd-page"` or `break-after="odd-page"` when in the middle of an odd page
- using `break-before="even-page"` or `break-after="even-page"` when in the middle of an even page
- it is important to remember that using two `break-before="page"` or `break-after="page"` blocks in a row does not constitute a blank page for the purposes of geometry testing
  - it is a "`not-blank`" page since there is an empty area on the page from the first of the two blocks

Copyright © Crane Softwrights Ltd.

Forced blank pages (cont.)

Forcing every second page to be blank in the flow is different

- cannot be triggered by stylesheet injection of breaks
  - no knowledge of the formatted page boundaries during transformation
- cannot be tested by `blank-or-not-blank=` because there is candidate flow for the even pages

It is acceptable to sequence a page geometry not containing any region for the flow

- the formatter always obtains "the next" page in a sequence of patterned pages when the flow overflows a page boundary
- a page is populated by all of the targeted content for the regions on the page
  - all static content for regions on the page is always formatted
  - if no region on the page is for the flow, none of the flow is consumed
  - the formatter is then finished with the page and goes to the next geometry in the sequence without changing its position in the flowed content

Also useful for pages of visible static content

- e.g. a ruled "Notes by the reader" page on the left hand side opposite content on the right hand side
- label the `<region-body>` on the un-flowed page for use by static content
- define the entire page body content in the static content for the flow

 Copyright © Crane Softwrights Ltd.

<page-sequence-master> Object

Chapter 8 - Flows, static content and page geometry sequencing
Section 4 - Page geometry sequencing

Purpose:

- the definition and name of a particular sequence of using page-masters

Content:

- (6.4.8) (single-page-master-reference|repeatable-page-master-reference|repeatable-page-master-alternatives)+
- Child objects (alphabetical):
  - <repeatable-page-master-alternatives>(6.4.11;288)
  - <repeatable-page-master-reference>(6.4.10;287)
  - <single-page-master-reference>(6.4.9;286)
- Referring object:
  - <layout-master-set>(6.4.7;63)

Required property:
master-name=(7.27.8;481)

Copyright © Crane Softwrights Ltd.

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

Excerpts from a draft XSLT stylesheet for producing training material:

```
<layout-master-set>
  <simple-page-master master-name="frame-left" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-left"/>
  </simple-page-master>
  <simple-page-master master-name="frame-right" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-right"/>
  </simple-page-master>
  <page-sequence-master master-name="frames">
    <repeatable-page-master-alternatives
      maximum-repeats="no-limit">
      <conditional-page-master-reference
        master-reference="frame-right" odd-or-even="odd"/>
      <conditional-page-master-reference
        master-reference="frame-left" odd-or-even="even"/>
    </repeatable-page-master-alternatives>
  </page-sequence-master>
</layout-master-set>
...
<page-sequence master-reference="frames" ...>
```


Copyright © Crane Softwrights Ltd.

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

The page sequence used in this example prepares to alternately use differently-named page masters

- each page master has the same name for the before region and a different name for the after region
- the page sequence defines the static content to be rendered for every possible named region in all simple page masters triggered by the page sequence master
- only that static content for the regions that are used on a given page are rendered
    - any defined static content for regions that are not on the page is ignored

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

<single-page-master-reference> Object

Purpose:

- the specification of the single use of a page-master within a sequence of page-masters

Content:

- (6.4.9) EMPTY
- Referring object:
  - <page-sequence-master>(6.4.8;283)

Required property:
master-reference=(7.27.9;482)

Copyright © Crane Softwrights Ltd.

<repeatable-page-master-reference> Object

Purpose:

- the specification of the repeated use of a page-master within a sequence of page-masters

Content:

- (6.4.10) EMPTY
- Referring object:
  - <page-sequence-master>(6.4.8;283)

Required property:
master-reference=(7.27.9;482)

Optional property:
maximum-repeats=(7.27.10;482)

Copyright © Crane Softwrights Ltd.

<repeatable-page-master-alternatives> Object

Purpose:

- the collection of candidate page-master references from which the first one in document order is to be used based on successful status conditions detected by the formatter

Content:

- (6.4.11) (`conditional-page-master-reference`+)
- Child object:
  - `<conditional-page-master-reference>`(6.4.12;290)
- Referring object:
  - `<page-sequence-master>`(6.4.8;283)

Optional property:
`maximum-repeats=`(7.27.10;482)

Of note:

- child `<conditional-page-master-reference>` objects are tested in document order of the stylesheet
- stylesheet writer must order the children to ensure the first child that tests true for all conditions refers to the desired page geometry
- special consideration for a one-page page sequence:
  - a page-sequence of only one page tests true for both the first page of the sequence and the last page of the sequence
  - stylesheet writer decides how to treat a one-page page sequence and orders that choice in document order before the undesirable choice

 Copyright © Crane Softwrights Ltd.

<repeatable-page-master-alternatives> Object (cont.)

Excerpts from a draft XSLT stylesheet for this training material document:

```
<layout-master-set>
  <simple-page-master master-name="frame-left" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-left"/>
  </simple-page-master>
  <simple-page-master master-name="frame-right" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-right"/>
  </simple-page-master>
  <page-sequence-master master-name="frames">
    <repeatable-page-master-alternatives
      maximum-repeats="no-limit">
      <conditional-page-master-reference
        master-reference="frame-right" odd-or-even="odd"/>
      <conditional-page-master-reference
        master-reference="frame-left" odd-or-even="even"/>
    </repeatable-page-master-alternatives>
  </page-sequence-master>
</layout-master-set>
...
<page-sequence master-reference="frames" ...>
```


Copyright © Crane Softwrights Ltd.

<conditional-page-master-reference> Object

Purpose:

- a page-master choice available to the formatter when selecting from a collection of candidate page-masters

Content:

- (6.4.12) EMPTY
- Referring object:
  - <repeatable-page-master-alternatives>(6.4.11;288)

Required property:
master-reference=(7.27.9;482)

Optional properties:
blank-or-not-blank=(7.27.1;453) page-position=(7.27.14;488)
odd-or-even=(7.27.12;484)

 Copyright © Crane Softwrights Ltd.

<conditional-page-master-reference> Object (cont.)

Excerpts from a draft XSLT stylesheet for training material:

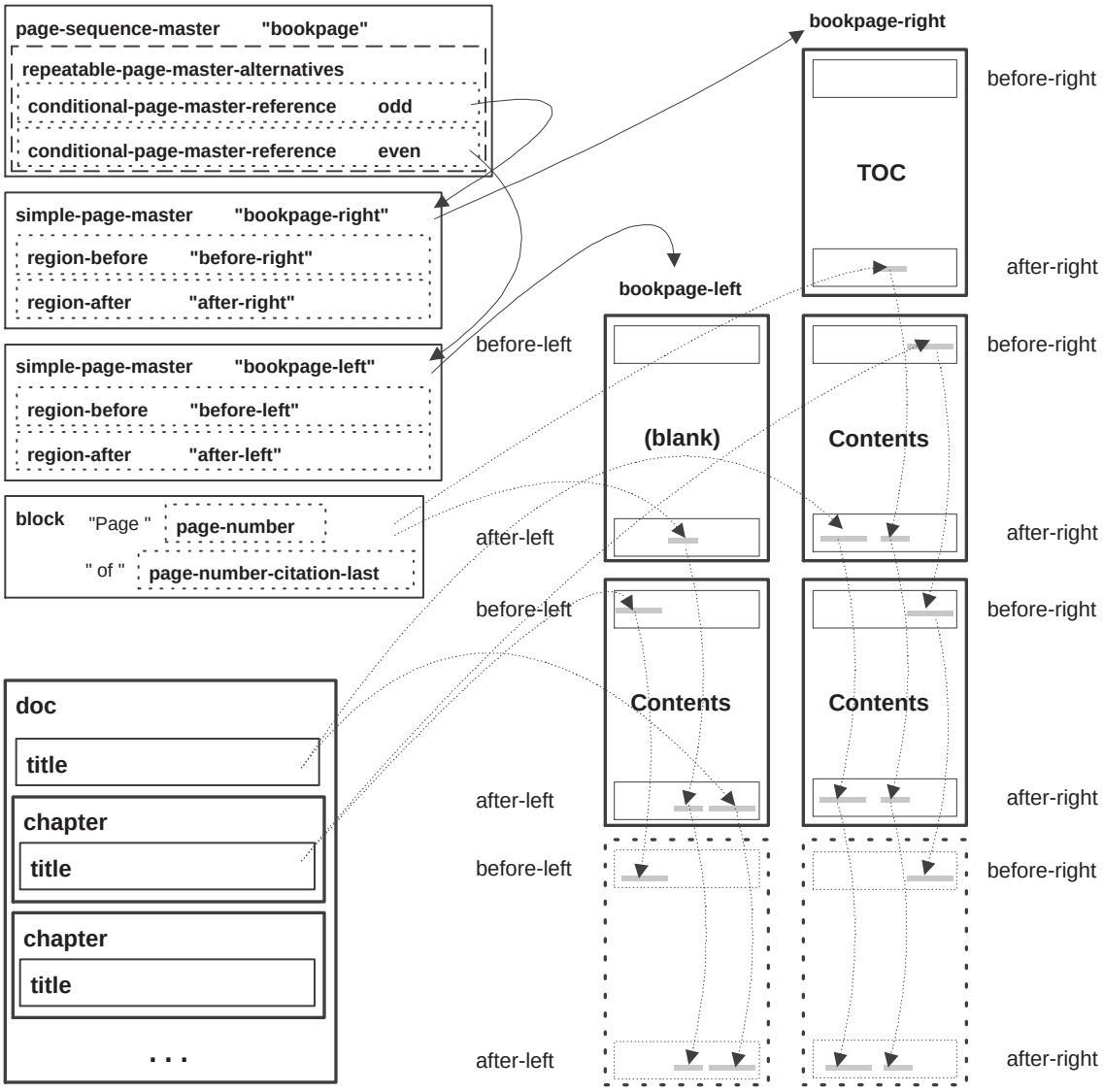
```
<layout-master-set>
  <simple-page-master master-name="frame-left" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-left"/>
  </simple-page-master>
  <simple-page-master master-name="frame-right" ...>
    <region-body region-name="pages-body" .../>
    <region-before extent=".3in" region-name="pages-before"/>
    <region-after extent=".3in" region-name="pages-after-right"/>
  </simple-page-master>
  <page-sequence-master master-name="frames">
    <repeatable-page-master-alternatives
      maximum-repeats="no-limit">
      <conditional-page-master-reference
        master-reference="frame-right" odd-or-even="odd"/>
      <conditional-page-master-reference
        master-reference="frame-left" odd-or-even="even"/>
    </repeatable-page-master-alternatives>
  </page-sequence-master>
</layout-master-set>
...
<page-sequence master-reference="frames" ...>
```


Copyright © Crane Softwrights Ltd.

Planning a more complex page sequence specification

Planning ahead helps more when considering more complex requirements

- consider planning a book with a table of contents and the contents each starting on the right-hand page
- different formatting of headers and footers for table of contents and for contents
  - table of contents with only the page number centered in the footer and an even number of pages in total count (to ensure the content starts on an odd page)
  - other pages with document title at bottom left of odd pages and bottom right of even pages, and chapter title at top right of odd pages and top left of even pages



Copyright © Crane Softwrights Ltd.

Chapter 9 - Bookmarks and indexes

- Introduction - Additional navigation aids
- Section 1 - Bookmarks
- Section 2 - Indexes

Outcomes:

- understand basic concepts of bookmark and index formatting objects

Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

Additional navigation aids (cont.)

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

Copyright © Crane Softwrights Ltd.

Bookmarks

Chapter 9 - Bookmarks and indexes
Section 1 - Bookmarks

A very simple bookmark structure can be defined as a property of the entire publication

- placed as an optional child of the `<root>` element
- a hierarchical tree with only one kind of branch
- an initial state for each entry indicates whether or not the entry appears when the bookmark pane is first opened
- a single bookmark can point into the given publication or point to an external location
- no counting limits imposed by XSL-FO (though there may be limits by the reader software)
    - there are no depth restrictions on the nested levels of bookmarks
    - there are no count restrictions on the bookmarks at any given level

`<bookmark-tree>`

- the root of all of the bookmarks
- nothing is presented for the root of the bookmark tree
- must have at least one bookmark child and may have any number more

`<bookmark>`

- a branch of the bookmark tree
- represents a visible entry in the bookmarks presented to the reader
- may have any number of bookmark children

`<bookmark-title>`

- a leaf of the bookmark tree showing the title of the bookmark
- this is the value used by the reader for navigation
- very limited formatting control
    - only color, bold and italic

 Copyright © Crane Softwrights Ltd.

Bookmarks (cont.)

The `starting-state=` relates to the visibility of the child bookmarks, not the bookmark to which the property is applied

- it is thus impossible for all bookmarks to be invisible when starting up
- the default visibility is "`show`", making it visible

There are many ways one may wish to initialize the visibility of the bookmarks

- which bookmarks do you want visible when the reader first opens the bookmark pane?
- just the major sections:

```
- - Prelude
  - Overview
  + Introduction
  + Introducing XSL-FO
  + The context of XSL-FO
  ...
```

- the publication title and the major sections:

```
- + Practical Formatting Using XSL-FO
    - Prelude
    - Overview
    + Introduction
    + Introducing XSL-FO
    + The context of XSL-FO
  ...
```

- just the publication title:

```
- + Practical Formatting Using XSL-FO
```

- fully expanded or expanded to an arbitrary level

Copyright © Crane Softwrights Ltd.

<bookmark-tree> Object

Purpose:

- the root of the bookmark tree
- has no visible text

Content:

- 1 (6.11.1) (bookmark+)
- Child object:
  - 1 <bookmark>(6.11.2;300)
- Referring object:
  - <root>(6.4.2;61)

No properties are defined for this formatting object.

Properties of interest:

- very limited formatting features

 Copyright © Crane Softwrights Ltd.

<bookmark-tree> Object (cont.)

An example bookmark tree for this training material

- only top-levels are initially visible
- all layers below are initially invisible

```
<bookmark-tree>
 <bookmark internal-destination="index">
  <bookmark-title>Prelude</bookmark-title>
 </bookmark>
 <bookmark internal-destination="start">
  <bookmark-title>Overview</bookmark-title>
 </bookmark>
 <bookmark starting-state="hide" internal-destination="desc">
  <bookmark-title>Introduction</bookmark-title>
  <bookmark internal-destination="desc">
   <bookmark-title>Practical Formatting Using XSL-FO</bookmark-title>
  </bookmark>
 </bookmark>
 <bookmark starting-state="hide" internal-destination="intro-fo">
  <bookmark-title>Introducing XSL-FO</bookmark-title>
  <bookmark starting-state="hide" internal-destination="intro-fo-f">
   <bookmark-title>Introduction</bookmark-title>
   <bookmark internal-destination="intro-fo-f">
    <bookmark-title>Contrasting browsed vs. paginated presentations
</bookmark-title>
   </bookmark>
  </bookmark>
 </bookmark>
 <bookmark starting-state="hide" internal-destination="ctxt-fo">
  <bookmark-title>The context of XSL-FO</bookmark-title>
  <bookmark starting-state="hide" internal-destination="xml-family">
   <bookmark-title>Introduction</bookmark-title>
   <bookmark internal-destination="xml-family">
    <bookmark-title>Overview</bookmark-title>
   </bookmark>
  </bookmark>
  ...
```


Copyright © Crane Softwrights Ltd.

<bookmark> Object

Chapter 9 - Bookmarks and indexes
Section 1 - Bookmarks

Purpose:

- a branch in the bookmark tree
- represents a visible entry in the bookmarks presented to the reader

Content:

- (6.11.2) (bookmark-title,bookmark*)
- Child objects (alphabetical):
  - <bookmark>(6.11.2;300)
  - <bookmark-title>(6.11.3;301)
- Referring objects:
  - <bookmark-tree>(6.11.1;298)
  - <bookmark>(6.11.2;300)

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:
external-destination=(7.23.6;469) starting-state=(7.23.10;498)
internal-destination=(7.23.8;476)

Property of interest:

- the starting-state= visibility applies to the child bookmarks, not the given bookmark

Copyright © Crane Softwrights Ltd.

<bookmark-title> Object

Chapter 9 - Bookmarks and indexes
Section 1 - Bookmarks

Purpose:

- a leaf in the bookmark tree
- represents a visible text of a bookmark presented to the reader

Content:

- 1 (6.11.3) (#PCDATA)
- Referring object:
  - 1 <bookmark>(6.11.2;300)

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:
color=(7.18.1;465) font-weight=(7.9.9;472)
font-style=(7.9.7;472)

Shorthand influencing the above properties:
font=(7.31.13;470)

Properties of interest:

- very limited formatting features

Copyright © Crane Softwrights Ltd.

Resolving indexing page numbers

Chapter 9 - Bookmarks and indexes
Section 2 - Indexes

An arbitrary set of multiple page number citations is unsuitable for use in indexing

- every `<page-number-citation/>` is always visible so cannot be used
- any given index entry may have multiple citations where there is unnecessary duplication or detail
  - when two citations end up on the same page, the page number is repeated
  - when citations end up on two or more adjacent pages, all page numbers are shown instead of a page range

`<index-page-citation-list>` groups together sets of index page number citations

- very flexible and powerful presentation options for resolving and presenting index page numbers
- no semantics for the presentation of the entry, only the page numbers
- only the pages formatted up to this element participate in the indexing

The presentation of the entire index is still the stylesheet's responsibility

- sort order
  - are particular letters of the alphabet considered synonyms for grouping under the first letter in the index?
  - when sorting in another language, are the letters starting English words leading, interleaved with, or following the letters starting the other language
    - e.g. interleaved for Latin-based languages (French, German, etc.)
    - e.g. English letters before Japanese letters, with Japanese letter synonyms
    - e.g. Russian letters before English letters
- grouping of secondary index entries under primary index entries
- presentation (indentation and appearance)

 Copyright © Crane Softwrights Ltd.

Preparing the content for indexing

The content must be seeded with anchors to cite with index citations

- two kinds of index citations:
  - object citations
    - only citing the descendants of a single page
    - may be more than one page of descendants
    - a point citation could cite an empty wrapper
      - should use `keep-with-next=` property to keep the wrapper on the same page as the item
  - range citations
    - citing a consecutive sequence of pages in a range
    - this range is typically but not necessarily the same as the resolved range in the final index
    - two adjacent or overlapping range citations can end up as either separate ranges or a combined range in the end result

An index citation has one or two components

- `index-key=` (mandatory; unique value for each index entry)
  - any value can be used but should be a meaningful and meaningfully-unique value
  - e.g. "*`primary-value-here`*"
    - could represent a primary key value
  - e.g. "*`primary-value-here`*`
`*`secondary-value-here`*"
    - could represent a secondary key value in the context of a primary key value
    - this relies on the premise that the presence of "`
`" in any actual content value is extremely unlikely due to line-end-sequence normalization to `
`
- `index-class=` (optional annotation)
  - this annotates the key with a user-defined characteristic but does not modify the key or make the key unique
  - adjacent page number entries of different index classes remain distinct in the result set of page numbers
  - e.g. distinguishing ranges "`iv-xxi, 1-7`" from "`iv-7`"
    - the first uses two different classes for distinction, the second does not

Copyright © Crane Softwrights Ltd.

Preparing the content for indexing (cont.)

Most formatting objects can be an object citation

- the page numbers of all of the areas of the cited formatting object
- may result in more than one page number
- using an empty block or inline area suffices to make it a single page number
    - use keep properties to ensure the empty block is on the same page as the information being indexed

A pair of formatting objects is used to define the start and end of a range citation

- `<index-range-begin/>`
    - marks the start limit of areas defining the pages of the index range
    - useful as a range only when it has an `id=` property
- `<index-range-end/>`
    - marks the ending limit of areas defining the pages of the index range
    - must point to the matching range start by using `ref-id=`

 Copyright © Crane Softwrights Ltd.

<index-range-begin> Object

Purpose:

- mark the beginning of a range of pages to be cited in the index

Content:

- (6.10.4) EMPTY

Optional properties:

id=(7.30.8;475)
index-key=(7.24.2;475)
index-class=(7.24.1;475)

Copyright © Crane Softwrights Ltd.

<index-range-begin> Object (cont.)

Example from indexing.fo sample:

```
<index-range-begin id="a1" index-key="phrase1"/>
<block space-before=".5em" keep-with-next="always"
       index-key="phrase1&#xd;orig">
Section 1.10.32 of "de Finibus Bonorum et Malorum"...
</block>
<block space-before=".5em">
"Sed ut perspiciatis unde omnis...
</block>
<block space-before=".5em" keep-with-next="always"
       index-key="phrase1&#xd;xlate">
1914 translation by H. Rackham
</block>
<block space-before=".5em">
"But I must explain to you how...
</block>
<index-range-end ref-id="a1"/>
<index-range-begin id="a2" index-key="phrase2"/>
<block space-before=".5em" keep-with-next="always"
       index-key="phrase2&#xd;orig">
Section 1.10.33 of "de Finibus Bonorum et Malorum"...
</block>
<block space-before=".5em">
"At vero eos et accusamus et iusto...
</block>
<block space-before=".5em" keep-with-next="always"
       index-key="phrase2&#xd;xlate">
1914 translation by H. Rackham
</block>
<block space-before=".5em">
"On the other hand, we denounce...
</block>
<index-range-end ref-id="ap2"/>
```


Copyright © Crane Softwrights Ltd.

<index-range-end> Object

Purpose:

- mark the end of a range of pages to be cited in the index

Content:

- (6.10.5) EMPTY

Required property:
ref-id=(7.30.13;491)

Copyright © Crane Softwrights Ltd.

Assembling indexing citations

Simplest is to have only a single set of index page number citations

- one key value per `<index-page-citation-list>` entry
- no distinguishing index classes
- `<index-key-reference>` finds all of the pages for a key
    - `ref-index-key=` has the key value

One can include different key values in a single set of page numbers

- multiple `<index-key-reference>` children of one `<index-page-citation-list>`
- each can be formatted the same or differently
    - font weight
    - font style
    - `<index-page-number-prefix>`
        - content preceding each page number cited
    - `<index-page-number-suffix>`
        - content following each page number cited

Sample from the XSL-FO specification section 6.10:

```
<index-page-citation-list>
  <index-key-reference ref-index-key="Eiffel Tower;;;preferred"
                       font-weight="bold"/><!--XB-->
  <index-key-reference ref-index-key="Eiffel Tower;;;figure"
                       font-style="italic"/><!--XI-->
    <index-page-number-prefix>
      <inline>[</inline>
    </index-page-number-prefix>
    <index-page-number-suffix>
      <inline>]</inline>
    </index-page-number-suffix>
  </index-key-reference>
  <index-key-reference ref-index-key="Eiffel Tower;;"/><!--XX-->
</index-page-citation-list>
```

- a collection of three sets of page numbers
- the first ("`XB`") is shown in bold to indicate it is the preferred link
- the second ("`XI`") is in italics and annotated with square brackets to indicate a page with a figure
- all other key references ("`XX`") for the Eiffel Tower are plain

Copyright © Crane Softwrights Ltd.

`<index-page-citation-list>` Object

Purpose:

- to render a consolidated and massaged sequence of index page number citations

Content:

- 1 (6.10.7) (`index-page-citation-list-separator?`, `index-page-citation-range-separator?`,`index-key-reference+`)
- Child objects (alphabetical):
  - 1 `<index-key-reference>`(6.10.6;313)
  - 1 `<index-page-citation-list-separator>`(6.10.8;311)
  - 1 `<index-page-citation-range-separator>`(6.10.9;312)

Optional properties:

1 `merge-pages-across-index-key-references=`(7.24.6;482) 1 `merge-sequential-page-numbers=`(7.24.5;483)
1 `merge-ranges-across-index-key-references=`(7.24.4;483)

Properties of interest:

- use `merge-sequential-page-numbers=` to create ranges from consecutive page number citations, or keep them separate
  - "`merge`"(default) - a range is created for three or more consecutive page numbers
  - "`leave-separate`" - pages cited are listed separately
  - acts within pages from each `<index-key-reference>`
  - acts within pages across all `<index-key-reference>` when `merge-ranges-across-index-key-references=` is "`merge`"
- use `merge-ranges-across-index-key-references=` to ignore key reference distinctions and put all pages into ranges, or respect distinctions and preclude pages from different references from being in the same page range
  - "`merge`" (default)
  - "`leave-separate`"
- use `merge-pages-across-index-key-references=` to remove duplicate page citations from different references, or retain all page citations
  - "`merge`" (default)
  - "`leave-separate`"

Copyright © Crane Softwrights Ltd.

`<index-page-citation-list>` Object (cont.)

Example from `indexing.fo` sample:

```
01 <block>Index</block>
02 <block space-before="1em">Phrase 1 -
03   <index-page-citation-list>
04     <index-key-reference ref-index-key="phrase1"/>
05   </index-page-citation-list>
06 </block>
07 <block start-indent="1cm">original -
08   <index-page-citation-list>
09     <index-key-reference ref-index-key="phrase1&#xd;orig"/>
10   </index-page-citation-list>
11 </block>
12 <block start-indent="1cm">translated  -
13   <index-page-citation-list>
14     <index-key-reference ref-index-key="phrase1&#xd;xlate"/>
15   </index-page-citation-list>
16 </block>
17 <block space-before="1em">Phrase 2 -
18   <index-page-citation-list>
19     <index-key-reference ref-index-key="phrase2"/>
20   </index-page-citation-list>
21 </block>
22 <block start-indent="1cm">original -
23   <index-page-citation-list>
24     <index-key-reference ref-index-key="phrase2&#xd;orig"/>
25   </index-page-citation-list>
26 </block>
27 <block start-indent="1cm">translated  -
28   <index-page-citation-list>
29     <index-key-reference ref-index-key="phrase2&#xd;xlate"/>
30   </index-page-citation-list>
31 </block>
```

 Copyright © Crane Softwrights Ltd.

<index-page-citation-list-separator> Object

Purpose:

- defines the sequence of text used to separate cited pages and ranges in a massaged list of page numbers

Content:

- (6.10.8) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - <index-page-citation-list>(6.10.7;309)

No properties are defined for this formatting object.

Default used when not specified by the stylesheet is the two character sequence ", "

- comma followed by a regular, breaking space
- any specified value is not a global and has scope only within the one <index-page-citation-list>

Copyright © Crane Softwrights Ltd.

<index-page-citation-range-separator> Object

Purpose:

- defines the sequence of text used to separate the first and last number in a single page range in a massaged list of page numbers

Content:

- 1 (6.10.9) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - 1 <index-page-citation-list>(6.10.7;309)

No properties are defined for this formatting object.

Default used when not specified by the stylesheet is the one character "–"

- en dash
- any specified value is not a global and has scope only within the one <index-page-citation-list>

 Copyright © Crane Softwrights Ltd.

`<index-key-reference>` Object

Purpose:

- retrieve all page numbers on which there are areas from formatting objects with a given key

Content:

- 1 (6.10.6) (`index-page-number-prefix`?,`index-page-number-suffix`?)
- Child objects (alphabetical):
  - 1 `<index-page-number-prefix>`(6.10.2;314)
  - 1 `<index-page-number-suffix>`(6.10.3;315)
- Referring object:
  - 1 `<index-page-citation-list>`(6.10.7;309)

Required property:

1 `ref-index-key=`(7.24.7;491)

Optional property:

1 `page-number-treatment=`(7.24.3;488)

See `<index-page-citation-list>` Object (page 310) for an example

See Assembling indexing citations (page 308) for an example

Properties of interest:

- use `page-number-treatment=` to make the cited page number an active hyperlink to the citation using "`link`"
  - default is that the page number is not linked

Copyright © Crane Softwrights Ltd.

<index-page-number-prefix> Object

Purpose:

- defines a sequence of formatting objects to prefix cited pages placed in a massaged list of page numbers

Content:

- 1 (6.10.2) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - 1 <index-key-reference>(6.10.6;313)

No properties are defined for this formatting object.

 Copyright © Crane Softwrights Ltd.

`<index-page-number-suffix>` Object

Purpose:

- defines a sequence of formatting objects to prefix cited pages placed in a massaged list of page numbers

Content:

- 1 (6.10.3) (#PCDATA|%inline;)*
- Child object:
  - %inline;(6.2;72)
- Referring object:
  - 1 <index-key-reference>(6.10.6;313)

No properties are defined for this formatting object.

Copyright © Crane Softwrights Ltd.

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

- Introduction - Spacing and arrangement constraint definition
- Section 1 - Break conditions
- Section 2 - Widows and orphan control
- Section 3 - Keep conditions
- Section 4 - Control of the spacing between areas
- Section 5 - Border specifications
- Section 6 - Displaying backgrounds

Outcomes:

- understand basic concepts of spacing and stacking

 Copyright © Crane Softwrights Ltd.

Spacing and arrangement constraint definition

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

Copyright © Crane Softwrights Ltd.

Spacing and arrangement constraint definition (cont.)

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

 Copyright © Crane Softwrights Ltd.

Breaks

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 1 - Break conditions

One often needs to arbitrarily continue flowing information at the start of the next line, column or page

- if no change to static content, one need not use a new page sequence to begin on a new page
- may need to continue at the top of an even page number or odd page number
    - may introduce a blank page that can be accommodated in the page sequence master

Breaking at the line level can be done with an empty `<block>` object

- adds a block of zero lines to the flow, interrupting the block-progression-direction
- `<block/>` behaves somewhat like the HTML `<br>` element
- multiple breaks in a row do not create multiple lines as in HTML
    - the areas created by the blocks do not have any dimensions
- can be nested inside of a block of lines without being considered an error
- not appropriate for typical program listings or other monospace presentations
    - see discussion of preservation of white space in Preserving white space (page 110)
- `<block><leader/></block>` creates an empty line
    - spaces in the block would get collapsed by default property values
    - the default pattern for a leader is constructed of spaces that are not collapsed

Breaking at the column or page level can be done with the `break-before=` and `break-after=` properties on a block

- the block may be empty
- can specify to the next column
- can specify to the next page
    - regardless of page parity
    - to the next even or odd page

Specifying a break does *not* mean forcing a break

- the property specifies a condition to be met, not an action to be performed
- if the information already satisfies the break requirement, nothing is added to the rendering of the flow

Copyright © Crane Softwrights Ltd.

Breaks (cont.)

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 1 - Break conditions

Conditionally breaking lines

- sometimes necessary to display content as a whole, but provide break points for the flowing algorithm to accommodate narrow margins
  - e.g. displaying long URL or URI addresses in narrow-width table cells
  - `http://www.mycompany.com/homePageEnglish.htm`
- hyphenation could introduce arbitrary dashes that materially affect the displayed result
  - `http://www.mycompany.com/homePage-English.htm`
- using a zero-width space (``) will be invisible if no break needs to happen
- provides a place to break the line if there is no room for the entire line
  - `http://www.mycompany.com/homePageEnglish.htm`

    `http://www.mycompany.com/homePageEnglish.htm`

Copyright © Crane Softwrights Ltd.

Widows and orphans

The natural breaking of blocks of lines may result in an unpleasant appearance

- seeing only one line of a block at the bottom or top of a page may leave the impression there is only one line of information
  - having two or more lines can leave the better impression there is more than just those lines in the block on the other side of the page break
- the effect is magnified when the block is justified
  - the last line of a justified block is typically not justified
  - when positioned at the top of a page, the unjustified last line may look more like a heading

Widows are the lines at the end of a block overflowing to the start of a page

- can specify the minimum number of line-areas allowed in the first area of the page generated by the last lines of a block
- initial value of `widows=` is "2"

Orphans are the lines at the start of a block overflowing at the end of a page

- can specify the minimum number of line-areas allowed in the last area of the page generated by the first lines of a block
- initial value of `orphans=` is "2"

Breaking happens to the next logical reference area

- column break if multiple columns in the page
- a page break if at the end of the last column

Copyright © Crane Softwrights Ltd.

Widows and orphans (cont.)

The entire block is moved to the next reference area if neither condition can be satisfied

- widows and orphans are not tested until there are widow lines to be accommodated
- the entire block is moved when there are not enough orphan lines remaining after accommodating the minimum number of widow lines

The `samp/widows.fo` file illustrates the setting of different values:

Test 1	W=0,O=0.	**Test 2**	W=4,O=0;	**Test 3**	Block:
Block:		Block:	W=4,O=0;		W=4,O=4;
W=0,O=0;		W=4,O=0;	W=4,O=0;		W=4,O=4;
W=0,O=0;			W=4,O=0.		W=4,O=4;
W=0,O=0;					W=4,O=4;
W=0,O=0;					W=4,O=4.

Note the following regarding the three examples of the six-line block wrapping:

- in each test the values of widows and orphans ("W=" and "O=") are different to illustrate the behavior for a six-line long block that wraps at the end of each column
- in Test 1, the widows specification is 0 which is less than or equal to the actual one widow line in the block
  - the block is left untouched and shows what would result when widows and orphans are not defined
  - remember when not specified the values are defined as "2", not "0"
- in Test 2, the widows specification is 4 which is greater than what would be only one widow line in the block as in Test 1
  - three orphan lines are moved to the next column to satisfy the widow count
  - the remaining orphan count of 2 does satisfy the specified orphan count of 0
- in Test 3, the widows specification is 4 which is greater than what would be only one widow line in the block as in Test 1
  - three orphan lines are first moved to the next column to satisfy the widow count, reducing the orphan count to two, as in Test 2
  - the remaining orphan count of 2 does not, however, satisfy the specified orphan count of 4, so all remaining orphans are then moved to the next column as well

 Copyright © Crane Softwrights Ltd.

Keeps

Can prevent information from separating over line, column or page contexts

- line-oriented keeps for inline-level constructs only
  - use *`keep`*`.within-line=` for the line context
- column- and page-oriented keeps for block-level constructs
  - use *`keep`*`.within-column=` for the column context
  - use *`keep`*`.within-page=` for the page context

Overriding widows and orphans with the `keep-together=` property

- keeping descendants together
- adding a context break to the flow before an area whose entire content will not fit within the `keep-together=` component context

Injecting context breaks with the `keep-with-previous=` and `keep-with-next=` properties

- keeping siblings together
- adding a context break to the flow before an area when
  - the following area has a `keep-with-previous=` property component and both areas will not fit within the component context
  - the area has a `keep-with-next=` property component and both areas will not fit within the component context

Relative strengths of keeps can be specified to ensure "more important" areas are kept together when "less important" areas cannot be kept together

- e.g. safety publishing where a list of steps is attempted to be kept together
  - there may be warnings found in the list of steps
  - if the steps cannot be kept together, the more important blocks associated with the warning are to be kept together

`keep-together=` is inherited and is applied to ***all*** areas of influence by the formatting object

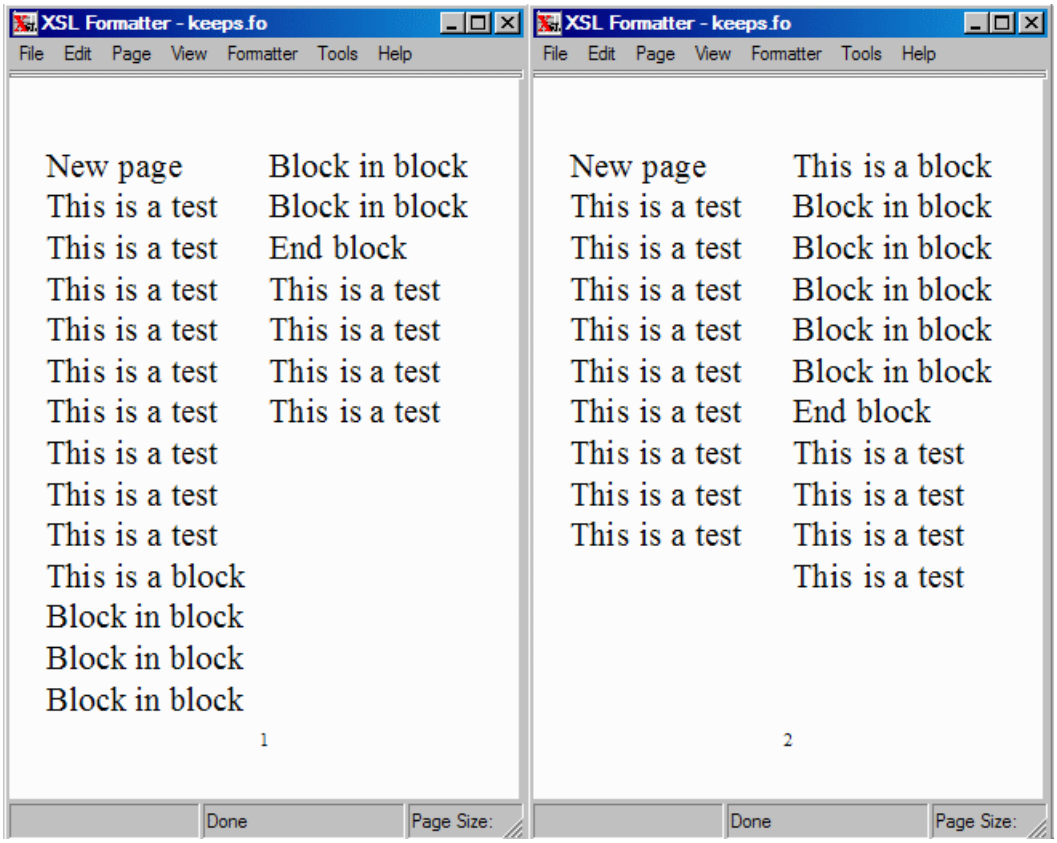
- the `.within-line` compound component of `keep-together=` is implicitly specified if not explicitly specified as otherwise
- inline areas in the block will inherit the `.within-line` component and apply it to the line-areas returned, usually causing undesirable results
- text needs `keep-together="auto"` to undo the setting by the ancestor
  - alternatively, it is easier to just use `.within-page=` or `.within-column=` on the ancestor

Copyright © Crane Softwrights Ltd.

Examples of keeps

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 3 - Keep conditions

Consider the following "before and after" of not using keeps and then using keeps:



Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Examples of keeps (cont.)

The following markup from `samp/keeps.fo` illustrates controlling the keep for descendants:

```
<layout-master-set>
  ...
    <region-body region-name="frame-body" column-count="2" .../>
  ...
</layout-master-set>
 ...
<flow flow-name="frame-body" font-size="40pt">
  <block break-before="page">New page</block>
  <block>This is a test</block>
  ...
  <block>This is a test</block>
  <block>This is a block
    <block>Block in block</block>
    ...
    <block>Block in block</block>
    End block</block>
  <block>This is a test</block>
  ...
  <block>This is a test</block>
  <block break-before="page">New page</block>
  <block>This is a test</block>
  ...
  <block>This is a test</block>
  <block keep-together.within-column="always">This is a block
    <block>Block in block</block>
    ...
    <block>Block in block</block>
    End block</block>
  <block>This is a test</block>
  ...
  <block>This is a test</block>
</flow>
```

Of note:

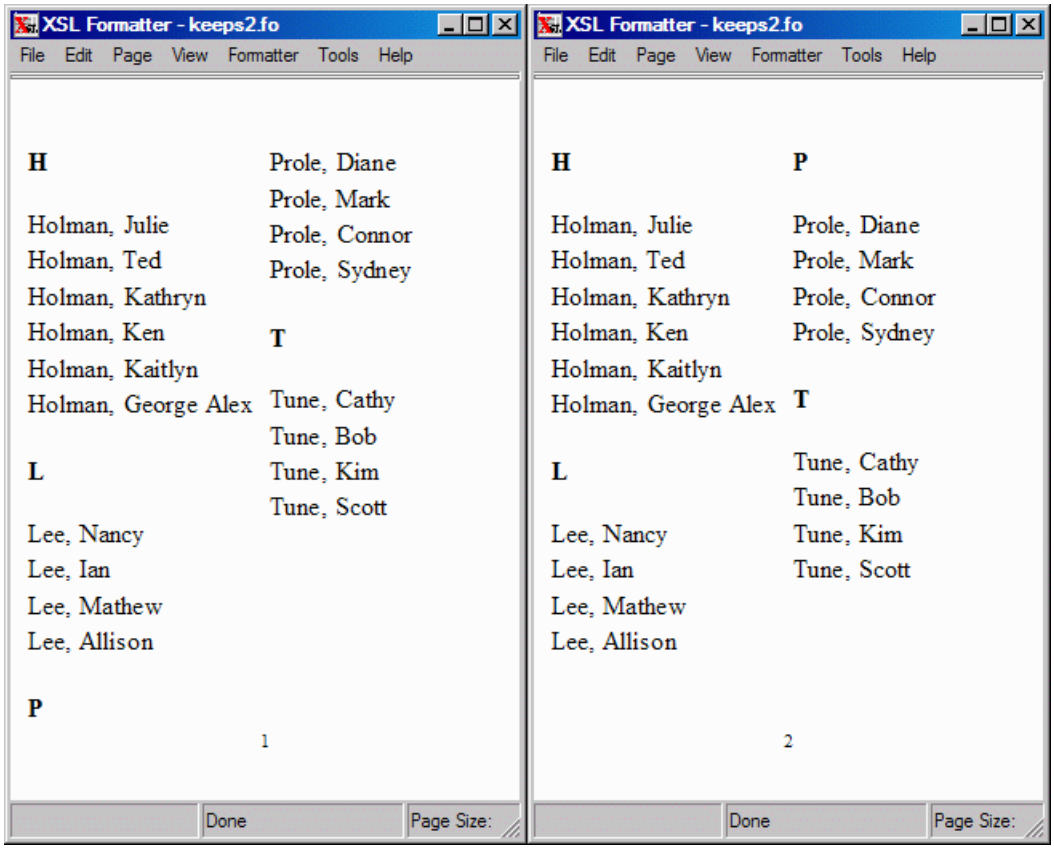
- the lines reading "Block in block" spread over the column break on page 1
- these lines are kept together within the column on page 2

Copyright © Crane Softwrights Ltd.

Examples of keeps (cont.)

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 3 - Keep conditions

Consider the following "before and after" of not using keeps of siblings and then using keeps:



Note how the space at the head of each reference area is conditional and is, therefore, discarded.

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Examples of keeps (cont.)

The following markup from `samp/keeps2.fo` illustrates controlling the keep for siblings:

```
01   <block space-before="1.5cm" space-after="1.3cm"
02          font-weight="bold">L</block>
03   <block space-before=".2em">Lee, Nancy</block>
04   <block space-before=".2em">Lee, Ian</block>
05   <block space-before=".2em">Lee, Mathew</block>
06   <block space-before=".2em">Lee, Allison</block>
07
08   <block space-before="1.5cm" space-after="1.3cm"
09          font-weight="bold">P</block>
10   <block space-before=".2em">Prole, Diane</block>
11   <block space-before=".2em">Prole, Mark</block>
12   <block space-before=".2em">Prole, Connor</block>
13   <block space-before=".2em">Prole, Sydney</block>
14 ...
15   <block space-before="1.5cm" space-after="1.3cm"
16          keep-with-next.within-column="always"
17          font-weight="bold">L</block>
18   <block space-before=".2em">Lee, Nancy</block>
19   <block space-before=".2em">Lee, Ian</block>
20   <block space-before=".2em">Lee, Mathew</block>
21   <block space-before=".2em">Lee, Allison</block>
22
23   <block space-before="1.5cm" space-after="1.3cm"
24          keep-with-next.within-column="always"
25          font-weight="bold">P</block>
26   <block space-before=".2em">Prole, Diane</block>
27   <block space-before=".2em">Prole, Mark</block>
28   <block space-before=".2em">Prole, Connor</block>
29   <block space-before=".2em">Prole, Sydney</block>
```

Of note:

- on page 1 the heading for the third group of names is in a different column than the names
- on page 2 the heading for the third group of names is kept together with the names

Copyright © Crane Softwrights Ltd.

Keep strength

The strength of the keep is specified in the property's value

- "`auto`" indicates no keep condition imposed
- "*`integer-number`*" indicates a relative keep strength that can be ignored
  - ignored without overflow condition if the keep area doesn't fit
- "`always`" indicates a keep that cannot be ignored
  - implies an overflow condition when if the keep area doesn't fit
- consider the situation where the pieces of a warning notice have to be kept together within a set of blocks that would be nice to keep together but may not fit on a page
  - the keep of the warning components would have a higher value than the keep of the set of blocks

A keep is attempted to be fit first within what remains within the context

- if a keep doesn't fit what remains, it begins within the next context (e.g. next column or next page)
- a keep is ignored if the amount being kept doesn't fit an entire context
  - when ignored, the keep's content is flowed as if there was no keep specified
  - does not begin at the start of the next context, but after the last flowed area
- the keeps of next higher strength within the flow of the keep being ignored are attempted to be kept together
  - any keeps of lower strength are ignored as that strength has already been addressed
- the keep of the next higher strength is then ignored if the length of that keep's flow cannot fit within the context
  - that keep's content is flowed as if there was no keep specified
- a keep of "`always`" is never ignored and if the areas do not fit in the context, an overflow condition is triggered

The size of the flow for a keep doesn't impact on the choice of page geometry

- a page-level keep will be tested on the next page geometry chosen by page sequencing
- a smaller page geometry is not simply ignored if it cannot accommodate the information in the keep and a subsequent page happens to fit the content
- the choice of page geometry is solely based on page sequencing and never on flow

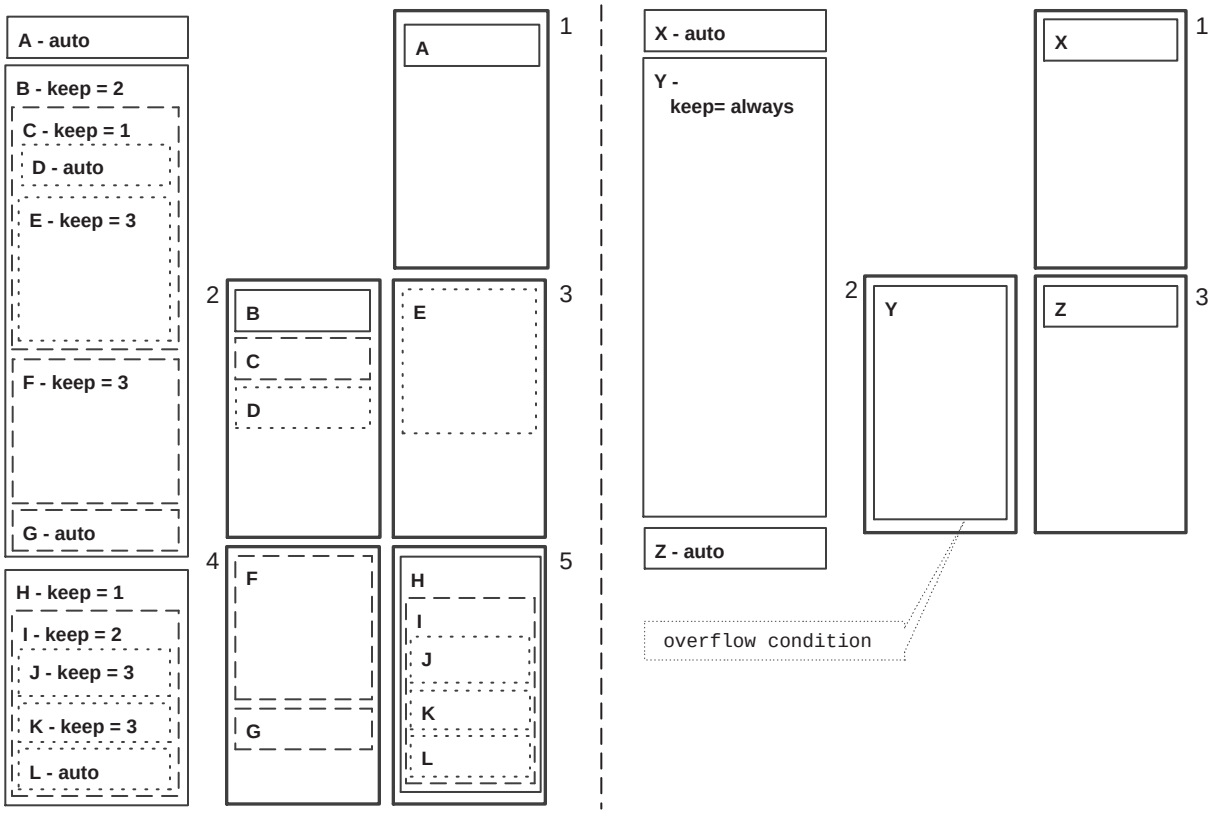
Explicit break conditions are stronger than any keep conditions

- a keep is ignored if a break is specified

 Copyright © Crane Softwrights Ltd.

Keep strength (cont.)

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 3 - Keep conditions



Of note:

- for each of the two examples, the blocks on the left depict the blocks in the flow, while the pages on the right depict where the blocks end up on the pages
- the pages are depicted with the odd page numbers on the right and the even page numbers on the left as is typical in left-to-right writing systems

Copyright © Crane Softwrights Ltd.

Keep strength (cont.)

In this example, each labeled block includes an area with that label plus the areas of the labeled blocks found therein

Using a numeric value allows the keep to be ignored

- A has no keep and flows as usual
- B has `keep="2"` and contains C, F, and G
    - all of B is longer than what remains on the first page, so the second page is used
    - because it doesn't all fit on the second page, the B keep is "broken" and so ignored
        - its content is flowed on the second page as if no keep was specified for B
- C has `keep="1"` and contains D and E
    - though all of C would fit on the next page, the keep strength specified is less than what has already been rejected, so the keep has no force and is ignored as if no keep was specified for C
- D has no keep and flows as usual
- E has `keep="3"` and doesn't fit on the rest of the second page
    - the keep strength is higher than what has been rejected, so it is respected
    - content is placed on the third page
- F has `keep="3"` and doesn't fit on the second page
    - the keep strength is higher than what has been rejected, so it is respected
    - content is placed on the fourth page
- G has no keep and flows as usual
- H has `keep="1"` and doesn't fit on the rest of the fourth page
    - it hasn't been part of any keep that has been rejected, and it fits in its entirety on a page, so it is moved to the next page
- I through L do not break over a page so their keep values are not considered

Using "`always`" does not allow the keep to be ignored

- X has no keep and flows as usual
- Y doesn't fit on the remainder of the first page,
    - because of the "`always`" it begins on the second page
    - because it is longer than the second page, an overflow condition exists
        - that which doesn't fit is lost
- Z has no keep and flows as usual
    - ends up at the start of the third page because the second page is full

 Copyright © Crane Softwrights Ltd.

Spacing, conditionality and precedence

Many components of compound properties specify the width of the space or border before or after formatting objects

- `space-before=` and `space-after=` for block-level constructs
- `space-start=` and `space-end=` for inline-level constructs
- components offer fine-tuned control over behaviors
  - *`length`*`.optimum=` for the preferable amount of space
    - using the shorthand property without a sub-field sets the optimum to the shorthand value
  - *`length`*`.minimum=` and *`length`*`.maximum=` for the limits to the actual amount of space
    - not specifying the limits implies the limits do not vary from the optimum
  - *`length`*`.conditionality=` for the discarding of unwanted space
    - at the beginning or termination of reference areas
    - also for the discarding of unwanted border widths
  - *`length`*`.precedence=` for the arbitration of conflicting space
    - where abutted space specifications are not desired
- spacing specifications are conditions to be met, not actions to be performed
  - makes stylesheet writing easier by allowing redundant specifications for adjacent areas

Can prevent border, padding and spacing from being used in certain conditions

- the pagination of flow may result in space specifications leading or trailing in reference areas
  - e.g. discard the space specified before a paragraph when the space is at the top of a page break
    - the stylesheet generating the blocks for paragraphs cannot know where the page breaks will occur in order to turn off the space specification
    - the default is to discard spacing used at the beginning or termination of a reference area
- it is often easy to write the stylesheet to always specify spacing and arbitrate between two coincident space specifications
  - e.g. arbitrate between a space specified before paragraphs in a section and a space specified after the title of a section
    - it may not be desirable to have the two spaces combine to too large a space before the first paragraph

Copyright © Crane Softwrights Ltd.

Spacing, conditionality and precedence (cont.)

Conditionality dictates whether space specifications are discarded

- the value "`retain`" can be specified to change the initial value of "`discard`"
- all contiguous space specifications that can discarded that are flowed to the start or to the end of a reference area are suppressed
    - each such area's size in the tree is set to zero
- all space specifications after the first non-discarded area (either space, padding, border, or content) and before the last non-discarded area are not suppressed as a result of conditionality
    - may be suppressed as a result of precedence or optimum size

 Copyright © Crane Softwrights Ltd.

Spacing, conditionality and precedence (cont.)

Precedence dictates arbitration between adjacent space specifications

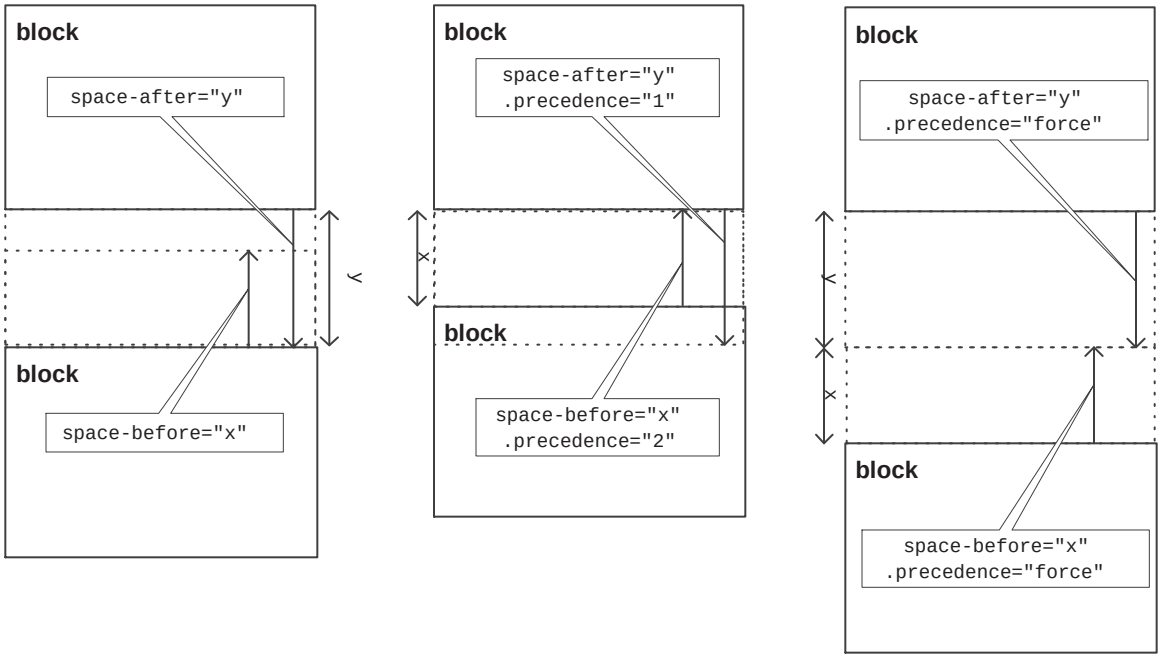
- e.g. the `space-after=` of a given block is adjacent to the `space-before=` of the following block
- an integer value (default is zero) is used for arbitration of precedence between adjacent space specifications
  - all unsuppressed areas whose precedence is less than the highest precedence integer value are then suppressed
- the value "`force`" will protect an unsuppressed area from being suppressed due to precedence

Optimum values are used for arbitration between those with equal precedence

- all unsuppressed areas whose optimum value is less than the greatest optimum value are then suppressed

Resolved minimum and maximum are derived from all remaining unsuppressed areas

- by arbitration described above all have the same (greatest) optimum value
- resolved minimum is the greatest of all minimums
- resolved maximum is the least of all maximums



Copyright © Crane Softwrights Ltd.

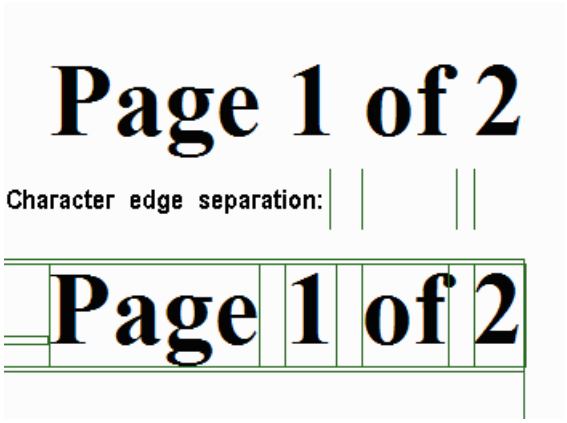
Character spacing

Can loosen or tighten the spacing between characters using `letter-spacing=`

- a positive value is in addition to the normal spacing, resulting in expanding the text
- a negative value is taken away from the normal spacing, resulting in compressing the text

Some formatting effects are unexpected due to the font:

- the apparent spacing between letters can be effected by special features of the font



Consider the above example:

- normal spacing is used which might give the impression characters are equally spaced
- the letter 2 is closer to the word "of" than the letter 1
    - this may give the impression the formatter is not doing is job properly
- the area boxes shown in the bottom half of the image indicate the inter-character spacing to be identical
- the digit 1 sits loosely inside its font box with a gap between the black of the digit and the edge of the box
    - this is necessary for numerical reports to ensure columnar digits align
- the digit 2 sits tightly inside its font box with no gap at the left of the black of the digit
- the letter "f" in this font spills over its font box into the following space area
    - this is a design aspect of the letter for the font
- the character edge separation between the black of the "1" and the black of the "o" ends up being much larger than that of the black of the "f" and the black of the "2"

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

Borders

Both block and inline areas can have the border visible using a non-zero border width

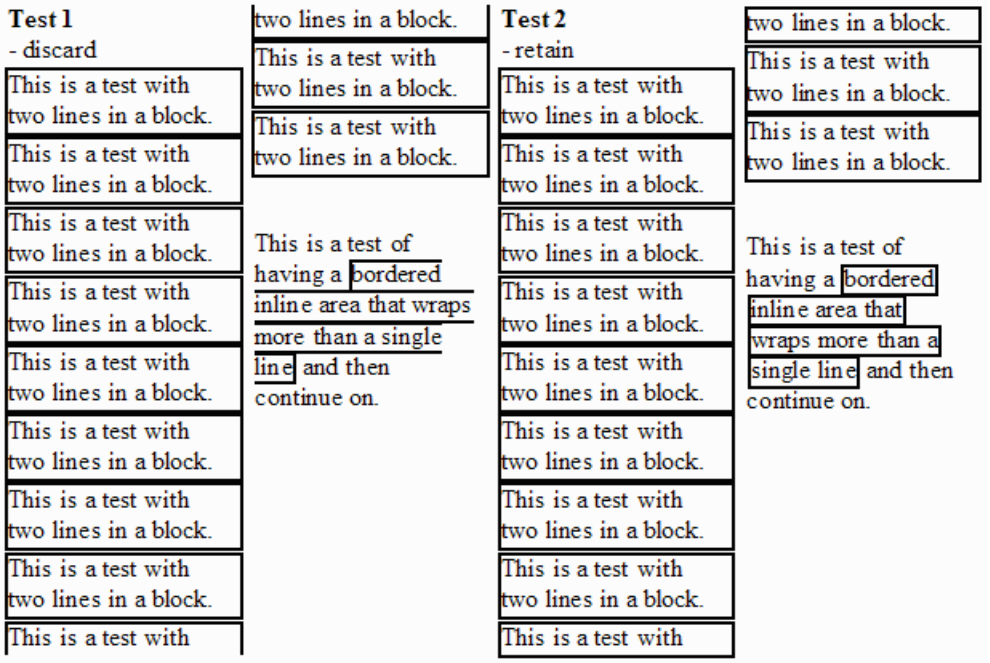
- separately specified properties on each of before, after, start, and end sides
  - `border-before-*=`, `border-after-*=`, etc.
- shorthand properties specified using `border-*=`
- `*-width=` (precedence given to wider widths in coincident table borders)
  - two computed values based on specified or defaulted component settings:
    - `*-width.`*`length`* (from resolution of sizes and precedence)
    - `*-width.conditionality=` (compound component)
      - "`discard`" or "`retain`" applies to the border of split areas
      - column and page reference areas for block areas (including table cells)
      - line reference areas for inline areas
  - a simple `*-width="`*`length`*`"` length value
    - implies `*-width.conditionality="discard"`
    - user-agent dependent values of "`thin`", "`medium`", and "`thick`"
- `*-style=` (listed in precedence when width equal for coincident table borders)
  - "`hidden`" forces `*-width="0pt"`
    - special case for table borders in that "`hidden`" has higher precedence than any other coincident border specification and the border is never seen
  - "`double`" with both lines and intervening space equal to the width
  - "`solid`"
  - "`dashed`"
  - "`dotted`"
  - "`ridge`"
  - "`outset`"
  - "`groove`"
  - "`inset`"
  - "`none`" forces `*-width="0pt"`
    - special case for table borders in that "`none`" has lower precedence than any other coincident border specification so may still be seen
- `*-color=` (precedence to construct nesting in coincident table borders of equal width and style)
  - color precedence has a different order than numeric construct precedence:
    - (highest to lowest) for cell, row, row group, column, column group, table
- region areas are fixed at a border width of 0pt and a padding of 0pt
- note that table border arbitration may be explicitly overridden by numeric precedence
  - see Table and cell borders (page 184) for details

Copyright © Crane Softwrights Ltd.

Borders (cont.)

Borders with a conditionality of "`discard`" (the default) will have borders along split reference area edges discarded

- a retained border will be drawn along the reference area's edge



Note the following about the example:

- each test includes a test of both block and inline level bordered areas
  - in Test 1 the conditional border widths are discarded by default
  - in Test 2 the conditional border widths are retained using the shorthand: `border-width.conditionality="retain"`
- the split areas have, respectively, open and closed edges along the splits in the first and second tests implementing each of "`discard`" and "`retain`" property values
- the presence of the retained border in the inline test changes the amount of text that fits on the third line

 Copyright © Crane Softwrights Ltd.

Backgrounds

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 6 - Displaying backgrounds

Properties control the background colors and images

- `background-color=`
  - either transparent to show through the parent area background (default) or a specified color
- `background-image=`
  - a URI specification to render as the background
- `background-repeat=`
  - repeatedly renders the background image in both directions, in the "x" direction (horizontal left to right according to the reference orientation), in the "y" direction (vertical top to bottom according to the reference orientation), or render only a single time
- `background-position-horizontal=`
  - position the background image in the horizontal direction (left to right according to the reference orientation)
- `background-position-vertical=`
  - position the background image in the vertical direction (top to bottom according to the reference orientation)
- `background-attachment=`
  - whether a background image scrolls in synchronization with the scrolling of an object, or if the presentation is electronic, the background image is fixed in place and the object's content scrolls over the fixed background

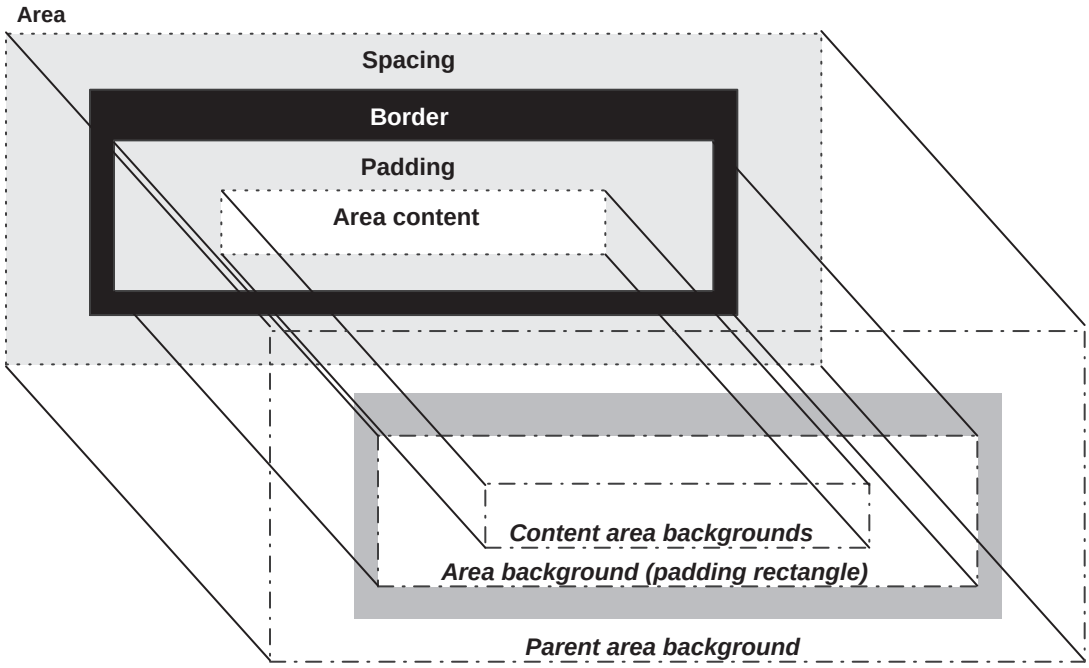
Copyright © Crane Softwrights Ltd.

Backgrounds (cont.)

Chapter 10 - Breaks, keeps, spacing, borders and backgrounds
Section 6 - Displaying backgrounds

An area's background is always within its padding rectangle

- any gaps in a border that is not solid (e.g. a dotted border) are transparent and will show through the parent area background



Precedence is to the contained areas

- a child area's background shows on top of an area's background
- an area's background shows on top of its parent's background

Copyright © Crane Softwrights Ltd.

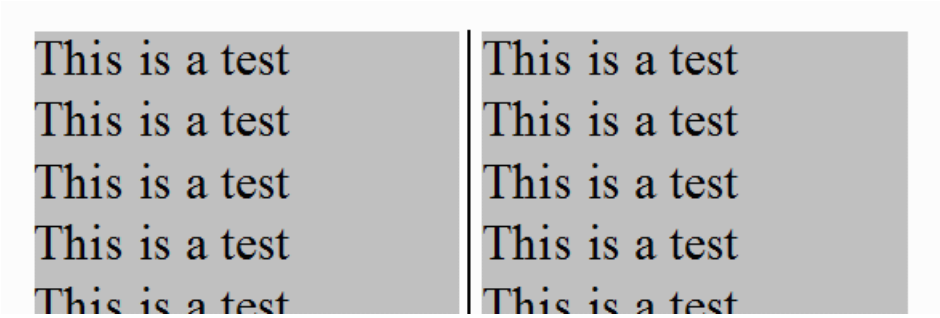
Decorating page columns using a background

Consider the need to decorate columns of a page

- no properties to decorate the normal-flow-reference-area of a column
- can create an image of the decoration for the background of the entire page
  - for a single column could center a narrow image of just the decoration
    - such as a simple vertical line
  - for multiple columns could create a page-wide graphic with strategically placed decoration
- the graphic of the decoration can be repeated for the entire page
- this scheme does not work for only a portion of the page, only the entire page

Excerpted from `samp/bkgrnd.fo`:

```
01     <region-body region-name="frame-body"
02                  column-count="2" column-gap=".5cm"
03                  background-position-horizontal="center"
04                  background-repeat="repeat-y"
05                  background-image='url("vertical-line.bmp")'/>
06     ...
07     <block background-color="silver">This is a test</block>
08     ...
```



Of note:

- the silver background is used here only to illustrate the width of the columns on the page

Copyright © Crane Softwrights Ltd.

Chapter 11 - Supplemental objects

- Introduction - Specialty constructs
- Section 1 - Specialty vocabulary
- Section 2 - Bidirectional writing support

Outcomes:

- understand basic concepts of lesser-used constructs

 Copyright © Crane Softwrights Ltd.

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:

- 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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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
- `<bidi-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

 Copyright © Crane Softwrights Ltd.

Change bars

Chapter 11 - Supplemental objects
Section 1 - Specialty vocabulary

- Change bars visually indicate content of note to the reader
  - perhaps the content indicated is different content from the content of a previous edition
  - different appearances may be used to indicate the different natures of the content
  - e.g. dashed, dotted and solid lines to the left and right of the content would indicate different aspects about the content
  - `<change-bar-begin/>`
    - the beginning of the named drawn line is aligned with the point in the flow at which this formatting object is used
    - properties govern the style, width, color, offset and placement of the line
  - `<change-bar-end/>`
    - the beginning of the named drawn line is aligned with the point in the flow at which this formatting object is used
  - `change-bar-class=` indicates the name of the line

Don't be prejudiced by the name of the construct

- the visual bars can be used for many purposes other than marking changes
- e.g. denoting sections pertaining to different responsible readers
- e.g. marking the presence of particular constructs such as definitions
- e.g. marking security levels of content

Copyright © Crane Softwrights Ltd.

<change-bar-begin> Object

Purpose:

- mark the beginning of the named drawn line shown to indicate some specific nature of the content

Content:

- (6.13.2) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)

Other required property:

change-bar-class=(7.30.1;464)

Other optional properties:

change-bar-color=(7.30.2;464)
change-bar-offset=(7.30.3;464)
change-bar-placement=(7.30.4;464)
change-bar-style=(7.30.5;464)
change-bar-width=(7.30.6;465)
z-index=(7.30.18;505)

Property of interest:

- change-bar-class= must indicate a name for which there is a paired ending formatting object

 Copyright © Crane Softwrights Ltd.

<change-bar-end> Object

Chapter 11 - Supplemental objects
Section 1 - Specialty vocabulary

Purpose:

- mark the end of the named drawn line shown to indicate some specific nature of the content

Content:

- 1 (6.13.3) EMPTY

Property sets:

- Common Accessibility Properties(7.5;426)
- Common Aural Properties(7.7;427)

Other required property:

1 change-bar-class=(7.30.1;464)

Property of interest:

- change-bar-class= must indicate a name for which there is a paired beginning formatting object

Copyright © Crane Softwrights Ltd.

Color system specification

Alternative color specifications may be needed by rendering requirement

- `<color-profile/>`
- formalized color definition provides for alternatives to built-in color granularity
- the stylesheet writer is always obliged to provide an RGB fallback color to accommodate formatters that do not support the requested color profile

Constructs out of line of the pagination need a home in an XSL-FO instance

- `<declarations>`
- a home location is required in the XSL-FO instance for referenced constructs that do not exist in any flow or repeatable construct
- for XSL-FO vocabulary purposes
    - e.g. `<color-profile>`
- may also be used for extension and other future vocabulary purposes

 Copyright © Crane Softwrights Ltd.

<color-profile> Object

Purpose:

- the declaration of a profile of candidate color values from which color specifications can be made by formatting objects

Content:

- (6.4.4) EMPTY
- Referring object:
  - <declarations>(6.4.3;348)

Required properties:
color-profile-name=(7.18.2;465) src=(7.30.16;498)

Optional property:
rendering-intent=(7.18.3;492)

Function of interest:

- src= is a URI recognized by the processor to represent the desired color system
  - same url() constraints as for image references
  - see Non-textual information (page 151)
- rgb-icc() is used for a color property value to reference the named color profile according to numeric arguments specific to the profile definition
  - note the function requires fallback RGB specification in the case the formatter does not recognize the desired color system

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

<declarations> Object

Purpose:

- a repository of global formatting object constructs including the collection of available color profiles and any collection of extended formatting objects supported by an XSL-FO processor

Content:

- 1 0 (6.4.3) (color-profile)+
- 1 1 (6.4.3) (color-profile)*
- Child object:
  - <color-profile>(6.4.4;347)
- Referring object:
  - <root>(6.4.2;61)

No properties are defined for this formatting object.

This is an optional child of <root>.

 Copyright © Crane Softwrights Ltd.

Characters and bi-directional text

Character formatting objects govern the presentation of glyphs associated with coded characters

- `<character>`
- implicitly created by the presence of text nodes (`#PCDATA`) in the XSL-FO instance
  - see Processing model of formatting (page 50)
- explicitly created using an element in the XSL-FO vocabulary
  - prevents the need to represent the character as a coded character
  - provides for specifying properties on a single character in the flow

Overriding the inherent behavior of characters as defined by Unicode

- `<bidi-override>`
- most uses of characters require no special handing
  - most Unicode characters have an intrinsic writing-direction property utilized by the formatter
    - some characters have a weak or neutral writing direction
  - the processor is supposed to respect the writing direction of all characters so that the stylesheet writer does not have to accommodate any combination used by the author of the XML document being formatted
- special formatting requirements may require behavior different than intrinsically defined
  - e.g. to protect left-to-right presentation from being influenced by the presence right-to-left characters
  - e.g. to display right-to-left characters in a left-to-right fashion

Copyright © Crane Softwrights Ltd.

<character> Object

Chapter 11 - Supplemental objects
Section 1 - Specialty vocabulary

Purpose:

- both the abstract formatting object implied by a simple text-node character in an XSL-FO instance and the concrete formatting object available to be used in place of a simple character

Content:

- (6.6.3) EMPTY

Property sets:

- Common Aural Properties(7.7;427)
- Common Border, Padding, and Background Properties(7.8;428)
- Common Font Properties(7.9;430)
- Common Hyphenation Properties(7.10;431)
- Common Margin Properties-Inline(7.12;433)
- Common Relative Position Properties(7.13;434)

Other required property:
character=(7.17.1;465)

Other optional properties:
alignment-adjust=(7.14.1;448)
alignment-baseline=(7.14.2;449)
baseline-shift=(7.14.3;453)
color=(7.18.1;465)
dominant-baseline=(7.14.5;468)
glyph-orientation-horizontal=(7.29.2;473)
glyph-orientation-vertical=(7.29.3;473)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
keep-with-next=(7.20.4;477)
keep-with-previous=(7.20.5;477)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
score-spaces=(7.30.15;495)
suppress-at-line-break=(7.17.3;499)
text-altitude=(7.29.4;500)
text-decoration=(7.17.4;500)
text-depth=(7.29.5;501)
text-shadow=(7.17.5;501)
text-transform=(7.17.6;501)
treat-as-word-space=(7.17.7;502)
visibility=(7.30.17;502)
word-spacing=(7.17.8;504)

Shorthands influencing the above properties:
font=(7.31.13;470)
page-break-after=(7.31.16;487)
page-break-before=(7.31.17;487)
vertical-align=(7.31.22;502)

Copyright © Crane Softwrights Ltd.

The importance of bidirectional text

Some international formatting requirements must accommodate mixing text of different writing directions

- the presentation of right-to-left scripts (e.g. Hebrew, Arabic, Thaana, and Syriac) may be embedded in the middle of presenting left-to-right scripts
- the stylesheet is mixing its boilerplate text with authored content of either possible direction from many different sources to produce the combined result
- the formatter is responsible for placement and interpretation of glyphs on the page based on the character code points in the XSL-FO instance
- the stylesheet writer is responsible for protecting text, where possible, from being influenced in the XSL-FO instance being formatted

Three aspects of writing direction influence the presentation of information on a line

- a line has an inline progression direction determined by the writing direction of the closest ancestral reference area
    - information is flowed onto lines in the inline progression direction, independent of the visual order of the characters in the information
- a group of adjacent characters may have semantic affinity and would thus be required to be flowed onto the lines in groups
    - groups are ordered in the inline progression direction
    - e.g. in a right-to-left progression direction there may be groups of left-to-right characters
        - the groups are rendered from the right to the left on the line, but the characters in each group are rendered from the left to the right
        - without grouping, adjacent strings of the same direction would be rendered as a whole, losing the semantic affinity in the characters
    - e.g. boilerplate information from the stylesheet is often semantically distinct from source file information, needing insulation from undue influence from the content
- characters from different scripts have inherent writing directions that may differ from the inline progression direction
    - usually necessary to respect the character direction while accommodating the inline progression direction (responsibility of the formatter)
    - may be necessary to override the inherent direction for a special effect
    - many characters are not associated with any language and are influenced by their proximity to characters from different scripts
- stylesheet writers can specify the progression direction of the information, the grouping of characters and the respect or override of the inherent Unicode direction in characters

Copyright © Crane Softwrights Ltd.

The mechanics of mixing text of different writing directions

Evolving rendering needs are being documented by Unicode and are influenced by XSL-FO

- rendering algorithm governed primarily by the Unicode Bidirectional Algorithm
  - `http://www.unicode.org/unicode/reports/tr9`
- nuances are described by XSL-FO 1.0 Recommendation - Unicode BIDI Algorithm (Section 5.8) in conjunction with the CSS definition
  - XSL-FO slightly modifies the CSS definition, such as assuming an unspecified `direction=` property is assumed to be that of the `writing-mode=` in effect
- these materials do not attempt to repeat the detailed algorithms described in the above documents, but only to give a general overview of the algorithm

There are three categories of direction strength for Unicode characters

- strong characters are from language groups with well defined left-to-right or right-to-left character progression directions
  - includes "marks" that are non-spacing invisible characters with strong direction
    - `‎` - LRM - left-to-right mark
    - `‏` - RLM - right-to-left mark
  - includes embedding and directional controls that are interpreted by the rendering algorithm
    - `‪` - LRE - left-to-right embed begin
    - `‫` - RLE - right-to-left embed begin
    - `‭` - LRO - left-to-right override begin
    - `‮` - RLO - right-to-left override begin
- weak characters are digits, currency symbols and some punctuation characters
  - includes mirroring characters whose presentation on the canvas may be different than their representation in the data
    - e.g. "(", ")", "[", "]", "<", ">", "{", "}", "«", "»", etc.
    - a mirrored character is rendered in the writing direction of adjacent text, which may require it to be flipped in presentation by the formatter
    - the stylesheet writer doesn't have any responsibility for doing the flipping
  - includes `‬` - PDF - Pop Directional Formatting - for embedding levels
- neutral characters are white space characters and some separator characters

Copyright © Crane Softwrights Ltd.

The mechanics of mixing text of different writing directions (cont.)

The embedding algorithm is based on isolating groups of sequences in "embedding levels"

- an embedded sequence has a progression direction for the members of the sequence
  - members are characters or other embedded sequences
- the stylesheet writer may need to introduce embedding levels to keep a sequence of characters together
  - the stylesheet would need to recognize or predict the semantic affinity of the group to know where to embed a new level
- using `<bidi-override>` will create an embedding level
  - the progression direction of the level is either the specified `direction=` or the current writing-mode direction if not specified
  - using a `unicode-bidi=` value of "`embed`" doesn't change the inherent writing direction of the individual characters in the embedding level
  - using a `unicode-bidi=` value of "`bidi-override`" forces the characters to render in the direction of the embedding level, ignoring the inherent properties of the characters
    - all characters are assigned the overriding direction as if they were strongly directed and their original nature is forgotten
- embedding levels are indicated in the resulting stream using LRE, RLE, LRO, RLO and PDF Unicode characters

The resulting stream of characters is grouped according to the Unicode directionality controls

- the grouping of characters crosses boundaries of embedding levels
- the act of grouping weak characters with strong characters gives direction to the weak characters
  - weak characters are influenced by their proximity to strong characters to become strongly directed themselves
  - weak characters between two strong characters of the same direction adopt that direction
  - weak characters preceding a strong character without white space interruption adopt the strong character's direction
  - weak characters following a strong character and any intervening white space adopt the strong character's direction
- the characters are rendered after all of the characters have been assigned a direction

Copyright © Crane Softwrights Ltd.

Illustration of bidirectional embedding

Examples in this book include sequences of right-to-left language text

- `&hebrew-test;` is "Hebrew test" in Hebrew
- `&arabic-test;` is "Arabic test" in Arabic

```
<!ENTITY hebrew-test  "&#x05D1;&#x05D3;&#x05D9;&#x05E7;&#x05D4;
                       &#x05E2;&#x05D1;&#x05E8;&#x05D9;&#x05EA;">
<!ENTITY arabic-test1 "&#x0625;&#x062e;&#x062a;&#x0628;&#x0627;">
<!ENTITY arabic-test2 "&#x0631; &#x0639;&#x0631;&#x0628;&#x064a;">
<!ENTITY arabic-test  "&arabic-test1;&arabic-test2;">
```

Consider a detailed example of mixing sequences of different directions in `bidimech.fo`:

- lines in the test are grouped differently to illustrate how groups are ordered in the writing direction for rendering before the characters found in the group are rendered
  - test "12" is left-to-right with embedded sequences of right-to-left text
  - test "23" is right-to-left with the identical content as test "12"
  - test "34" is almost identical to test "23" except for an introduced space after "89"
  - test "45" is right-to-left but the language text inside is in groups without overriding direction
  - test "56" is left-to-right but the language text inside is in groups that override direction
  - test "67" is right-to-left but the language text inside is in groups that override direction
  - note where the direction isn't specified, it is inferred by the writing mode
    - this is different than CSS that assumes left-to-right when not specified
- weak punctuation characters separate the strong script characters
  - the first three tests illustrate the differences in assignment of direction to weak characters
  - note the introduction of a space at the end of the "34" test changes the direction assignment to the "89" characters from the "89" characters in the "23" test
- embedding groups arrange the groups of left-to-right sequences
  - in test "34" the English text is shown to the left of the French text
  - in test "45" the French text is shown to the left of the English text
- overriding the direction results in improper presentation of language text
  - in test "56" the Hebrew and Arabic sequences are inappropriately presented
  - in test "67" the English and French sequences are inappropriately presented

 Copyright © Crane Softwrights Ltd.

Illustration of bidirectional embedding (cont.)

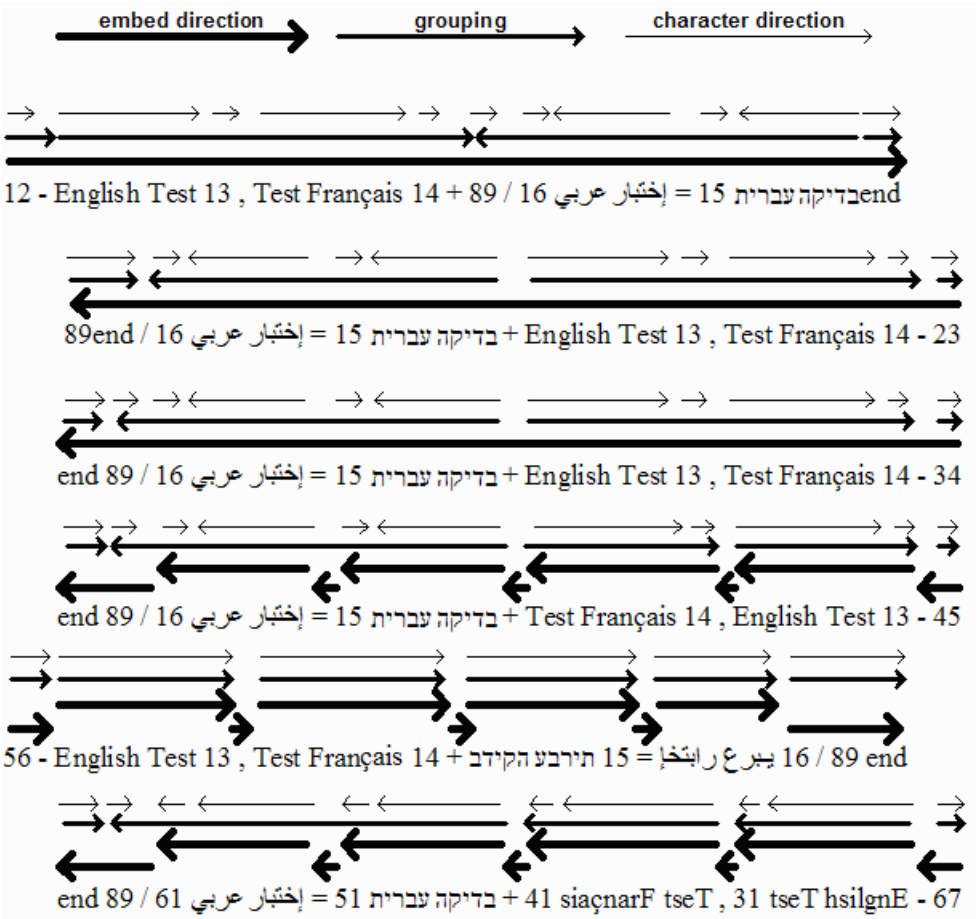
```
<block-container>
  <block>12 - English Test 13 , Test Français 14
  + &hebrew-test; 15 = &arabic-test; 16 / 89end</block>
</block-container>
<block-container writing-mode="rl-tb">
  <block>23 - English Test 13 , Test Français 14
  + &hebrew-test; 15 = &arabic-test; 16 / 89end</block>
</block-container>
<block-container writing-mode="rl-tb">
  <block>34 - English Test 13 , Test Français 14
  + &hebrew-test; 15 = &arabic-test; 16 / 89 end</block>
</block-container>
<block-container writing-mode="rl-tb">
  <block>45 - <bidi-override unicode-bidi="embed"
                            >English Test 13</bidi-override>
            , <bidi-override unicode-bidi="embed"
                            >Test Français 14</bidi-override>
            + <bidi-override unicode-bidi="embed"
                            >&hebrew-test; 15</bidi-override>
            = <bidi-override unicode-bidi="embed"
                            >&arabic-test; 16</bidi-override>
            / 89 end</block></block-container>
<block-container>
  <block>56 - <bidi-override unicode-bidi="bidi-override"
                            >English Test 13</bidi-override>
            , <bidi-override unicode-bidi="bidi-override"
                            >Test Français 14</bidi-override>
            + <bidi-override unicode-bidi="bidi-override"
                            >&hebrew-test; 15</bidi-override>
            = <bidi-override unicode-bidi="bidi-override"
                            >&arabic-test; 16</bidi-override>
            / 89 end</block></block-container>
<block-container writing-mode="rl-tb">
  <block>67 - <bidi-override unicode-bidi="bidi-override"
                            >English Test 13</bidi-override>
            , <bidi-override unicode-bidi="bidi-override"
                            >Test Français 14</bidi-override>
            + <bidi-override unicode-bidi="bidi-override"
                            >&hebrew-test; 15</bidi-override>
            = <bidi-override unicode-bidi="bidi-override"
                            >&arabic-test; 16</bidi-override>
            / 89 end</block></block-container>
```


Copyright © Crane Softwrights Ltd.

Illustration of bidirectional embedding (cont.)

This figure illustrates the on-screen interpretation of the `bidimech.fo` file:

- the line annotations are only for illustrative purposes regarding the groupings of characters and the character writing directions
- this test does not incorporate explicit use of Unicode directionality characters
  - the use of `<bidi-override>` introduces these characters into the rendering stream
- note how the grouping of characters for strength purposes goes over the bounds of embedded levels



Copyright © Crane Softwrights Ltd.

The bidirectional support challenge

The challenge for the stylesheet writer is to recognize when it is necessary to add marks or add embedding levels in the flow

- always using a single stream of character information will only work if all of the text is in the same direction as the inline progression direction
    - a stylesheet used internationally would need to respect the possibility of lines mixing weak and neutral characters with both directions of strongly directed characters
- e.g. consider mixing boilerplate text with authored text
    - authored text could be assumed to be presented to the author correctly while being entered, thus the content as entered could be formatted without special concerns
    - if English boilerplate text is to be abutted to language text of an arbitrary language, weak characters in the boilerplate text might be inadvertently influenced by the strong characters of the authored text
        - the influence could rearrange characters or flip the rendering of mirrored characters found in the boilerplate
    - use `<bidi-override>` to embed authored text at a separate embedding level than surrounding text

Be aware when formatting generated content such as table of contents entries

- mixing authored title content with stylesheet boilerplate content
    - leaders and page numbers follow the arbitrary title text
    - the content of the title is abutted to the leaders and page numbers
- wrapping the authored text in a different embedding level will protect the text characters from influencing the leaders, page numbers and other surrounding text

Copyright © Crane Softwrights Ltd.

The bidirectional support challenge (cont.)

The sample file `biditest.xml` includes titles utilizing text of different writing directions, each section with a title and a number of subsections.

```
<!DOCTYPE doc [
<!ENTITY hebrew-test  "&#x05D1;&#x05D3;&#x05D9;&#x05E7;&#x05D4;
                       &#x05E2;&#x05D1;&#x05E8;&#x05D9;&#x05EA;">
<!ENTITY arabic-test1 "&#x0625;&#x062e;&#x062a;&#x0628;&#x0627;">
<!ENTITY arabic-test2 "&#x0631; &#x0639;&#x0631;&#x0628;&#x064a;">
<!ENTITY arabic-test  "&arabic-test1;&arabic-test2;">
]>
<doc>
  <section>
    <title>English Test</title>
    <subsection>Sub 1 in English</subsection>
    ...
    <subsection>Sub 15 in English</subsection>
  </section>
  <section>
    <title>&hebrew-test;</title>
    <subsection>Sub 1 in Hebrew</subsection>
    ...
    <subsection>Sub 14 in Hebrew</subsection>
  </section>
  <section>
    <title>&arabic-test;</title>
    <subsection>Sub 1 in Arabic</subsection>
    ...
    <subsection>Sub 14 in Arabic</subsection>
  </section>
  <section>
    <title>Test Fran&#xe7;ais</title>
    <subsection>Sub 1 in French</subsection>
    ...
    <subsection>Sub 12 in French</subsection>
  </section>
```

 Copyright © Crane Softwrights Ltd.

The bidirectional support challenge (cont.)

The file `biditest.xsl` formats three tables of content, the first without protection, the second protecting the boilerplate of the entries from the arbitrary text directions of the characters obtained from the titles, and the third making the punctuation group follow in the text direction of the title by being isolated from the leader and following text.

```
01  <block space-after=".5cm">
02    This is a table of contents without using embed:
03  </block>
04  <xsl:for-each select="/doc/section">
05    <block text-align-last="justify">
06      <xsl:value-of select="position()"/>.
07      <xsl:value-of select="title"/>
08      (<xsl:value-of select="count(subsection)"/>)
09      <leader leader-pattern="dots"/>
10      <page-number-citation ref-id="{generate-id(.)}"/>
11    </block>
12  </xsl:for-each>
13  <block space-before="1.5cm" space-after=".5cm">
14    This is a table of contents using embed:
15  </block>
16  <xsl:for-each select="/doc/section">
17    <block text-align-last="justify">
18      <xsl:value-of select="position()"/>.
19      <bidi-override unicode-bidi="embed">
20        <xsl:value-of select="title"/>
21      </bidi-override>
22      (<xsl:value-of select="count(subsection)"/>)
23      <leader leader-pattern="dots"/>
24      <page-number-citation ref-id="{generate-id(.)}"/>
25    </block>
26  </xsl:for-each>
27  <block space-before="1.5cm" space-after=".5cm">
28    This is a table of contents using two embeds:
29  </block>
30  <xsl:for-each select="/doc/section">
31    <block text-align-last="justify">
32      <xsl:value-of select="position()"/>.
33      <bidi-override unicode-bidi="embed">
34        <xsl:value-of select="title"/>
35        <bidi-override unicode-bidi="embed">
36        (<xsl:value-of select="count(subsection)"/>)
37        </bidi-override>
38      </bidi-override>
39      <leader leader-pattern="dots"/>
40      <page-number-citation ref-id="{generate-id(.)}"/>
41    </block>
42  </xsl:for-each>
```


Copyright © Crane Softwrights Ltd.

The bidirectional support challenge (cont.)

The formatted result illustrates the impact of including the marks for protection from the characters of the titles:

This is a table of contents without using embed:

1. English Test (15) 2
2. 14) בדיקה עברית) 3
3. 14) إختبار عربي) 4
4. Test Français (12) 5

This is a table of contents using embed:

1. English Test (15) 2
2. בדיקה עברית (14) 3
3. إختبار عربي (14) 4
4. Test Français (12) 5

This is a table of contents using two embeds:

1. English Test (15) 2
2. (14) בדיקה עברית.....3
3. (14) إختبار عربي.....4
4. Test Français (12) 5

Note the behavior when protection is not used in the top rendering:

- the right-to-left text influences the following weak characters " (14" and makes them all part of a right-to-left group
  - the "(" character is displayed as ")" because it is a mirror character being displayed right-to-left
- the "14" still renders left-to-right because the grouping isn't powerful enough to override the inherent writing direction of the digits
- the original ")" and following space is influenced by the `<leader>`and the writing direction of the line, which is left to right, so the mirroring character doesn't change

 Copyright © Crane Softwrights Ltd.

<bidi-override> Object

Chapter 11 - Supplemental objects
Section 2 - Bidirectional writing support

Purpose:

- specifies how to manage and override the inherent Unicode text direction for a sequence of characters
- this is not needed for standard processing of a mixture of writing-mode characters, as the processor will recognize the inherent text direction and act accordingly

Content:

- (6.6.2) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- may begin with any number of <marker> children

Property sets:

- Common Aural Properties(7.7;427)
- Common Font Properties(7.9;430)
- Common Relative Position Properties(7.13;434)

Other optional properties:

color=(7.18.1;465)
direction=(7.29.1;468)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
score-spaces=(7.30.15;495)
unicode-bidi=(7.29.6;502)
word-spacing=(7.17.8;504)

Shorthand influencing the above properties:
font=(7.31.13;470)

Properties of interest:

- unicode-bidi=
  - use "embed" to wrap groups of characters where the groups are to progress in the writing direction of the parent construct
    - preserves the inherent writing direction of the characters in the group
  - use "bidi-override" to force the wrapped characters to progress in the writing direction of the parent construct
    - overrides the inherent writing direction of the characters in the group
- direction=
  - use "ltr" and "rtl" for the nature of the embedded behavior
  - the default direction is that of the parent construct

Copyright © Crane Softwrights Ltd.

Chapter 12 - Interactive objects

- Introduction - Dynamic and interactive object rendering
- Section 1 - Reflecting formatting object state by appearance
- Section 2 - Interactively changing the effective flow

Outcomes:

- understand basic concepts of interactive objects

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

Dynamic and interactive object rendering (cont.)

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

 Copyright © Crane Softwrights Ltd.

Dynamically changing property values

The formatting object sub-trees sensitive to state reflection are wrapped by a `<multi-properties>` object

- collects the sets of properties and those descendent areas that are to be sensitive to operator interaction
- may specify default values for properties to be inherited by descendent areas
- each `<multi-property-set/>` object captures the properties for a given state
    - use `active-state=` to specify the state for the collection of properties
- any, some, or all of the following states can have associated properties
    - "`link`" (unvisited)
        - true when a descendant `<basic-link>` object hasn't yet been visited
    - "`hover`"
        - true when a descendant area is under the influence of some pointing device, yet without activation of the interactivity
    - "`focus`"
        - true when a descendant area is the focus of operator interactivity (e.g. the user has tabbed to the hyperlink rather than using a pointing device)
    - "`active`"
        - true when a descendant area is being activated by operator interactivity (e.g. the time while the pointing device button is being pressed before being released)
    - "`visited`"
        - true when a descendant `<basic-link>` object has already been visited

A `<multi-properties>` object is a neutral construct

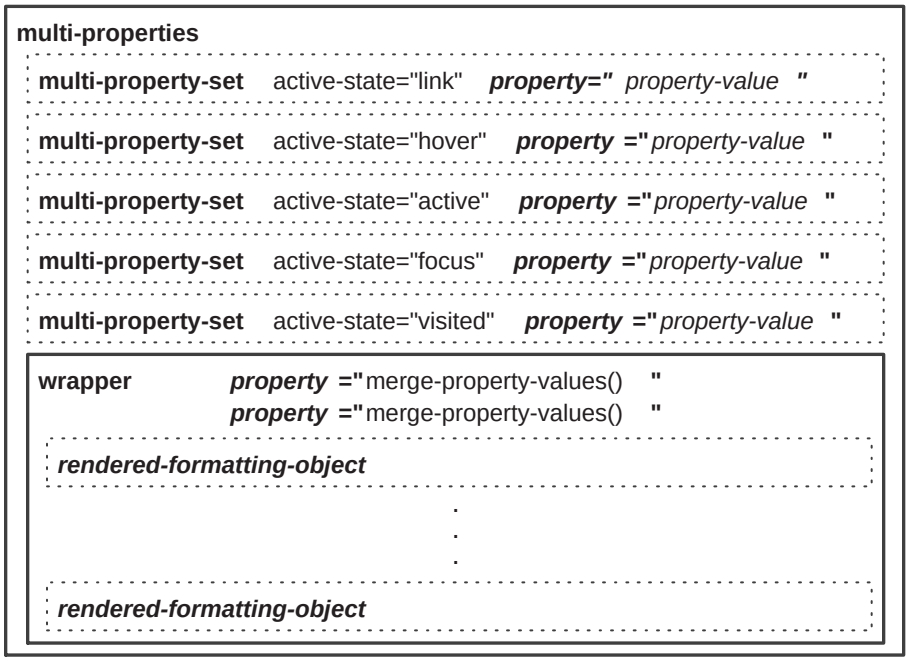
- in general can be used anywhere the formatting objects it wraps can be used

Copyright © Crane Softwrights Ltd.

Dynamically changing property values (cont.)

The objects that are to be rendered are captured in the `<wrapper>` object

- merges inherited properties with those sibling properties associated with current state by using `merge-property-values()` function
  - regular ancestor inheritance during formatting object tree refinement would miss the sibling `<multi-property-set/>` constructs
  - this function cannot be used with any other formatting object, nor with a `<wrapper>` that has any other parent object
- may specify any other inheritable properties for descendant constructs



 Copyright © Crane Softwrights Ltd.

<multi-properties> Object

Purpose:

- 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
    - supports reserved states for the <basic-link> object

Content:

- (6.9.6) (multi-property-set+,wrapper)
- Child objects (alphabetical):
    - <multi-property-set>(6.9.7;369)
    - <wrapper>(6.13.4;108)

Property sets:

- Common Accessibility Properties(7.5;426)

No other properties are defined for this formatting object.

Function of interest:

- merge-property-values() is used on the <wrapper> object that is ancestral to the formatting objects being rendered
    - merges the corresponding property from the properties specified for the active <multi-property-set> according to its state

Copyright © Crane Softwrights Ltd.

<multi-properties> Object (cont.)

Consider the need to reproduce typical browser colors for an unvisited and visited link

- blue when not yet visited
- purple when already visited

The following example is from `samp/multi1.fo`:

```
<block>This is a link:
<multi-properties>
  <multi-property-set active-state="link" color="blue"/>
  <multi-property-set active-state="visited" color="purple"/>
  <wrapper color="merge-property-values()">
    <basic-link
      external-destination="http://www.CraneSoftwrights.com"
>http://www.CraneSoftwrights.com</basic-link>
  </wrapper>
</multi-properties>
: end of block.
</block>
```

 Copyright © Crane Softwrights Ltd.

<multi-property-set> Object

Purpose:

- the set of properties associated with a single possible state of a formatting object or user interaction

Content:

- (6.9.7) EMPTY
- Referring object:
  - <multi-properties>(6.9.6;367)

Required property:
active-state=(7.23.1;448)

Optional properties:
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

Copyright © Crane Softwrights Ltd.

<multi-property-set> Object (cont.)

Continuing from samp/multi1.fo:

```
01 <block>This is a link:
02 <multi-properties>
03   <multi-property-set active-state="link" color="blue"/>
04   <multi-property-set active-state="visited" color="purple"/>
05   <wrapper color="merge-property-values()">
06     <basic-link
07       external-destination="http://www.CraneSoftwrights.com"
08 >http://www.CraneSoftwrights.com</basic-link>
09   </wrapper>
10 </multi-properties>
11 : end of block.
12 </block>
```

 Copyright © Crane Softwrights Ltd.

Dynamically changing formatting object sub-trees

The formatting objects participating in two or more renderings with dynamic changes are wrapped by a `<multi-switch>` object

- separated into individual mutually exclusive conditions
- always exactly one condition shown at a time
    - all other conditions are hidden when not shown
- operator interactivity can change which condition is shown
    - the action of doing so will hide all conditions not being shown

Individual conditions are wrapped by child `<multi-case>` objects

- one such child per condition
- named for reference purposes as the target of a toggle
- titled for display purposes when selecting from a number of cases

Changing which condition is shown is enabled by the `<multi-toggle>` object

- defines a "hot spot" on a page
- surrounds the flow object sub-trees sensitive to operator interaction
    - the essence of the operator interaction is defined by the formatter
- can have any number of toggles in a given sequence for different behaviors
    - each toggle can only change the current shown state of one other case
        - may provide a list of case names from which the operator can choose
            - the list is documented using the respective case titles
        - may cycle around sibling cases in either direction
    - can only change states amongst sibling `<multi-case>` objects
- only sensitive areas need be surrounded, not the entire content

Descendent `<multi-switch>` objects in `<multi-switch>` objects can make for very rich presentation

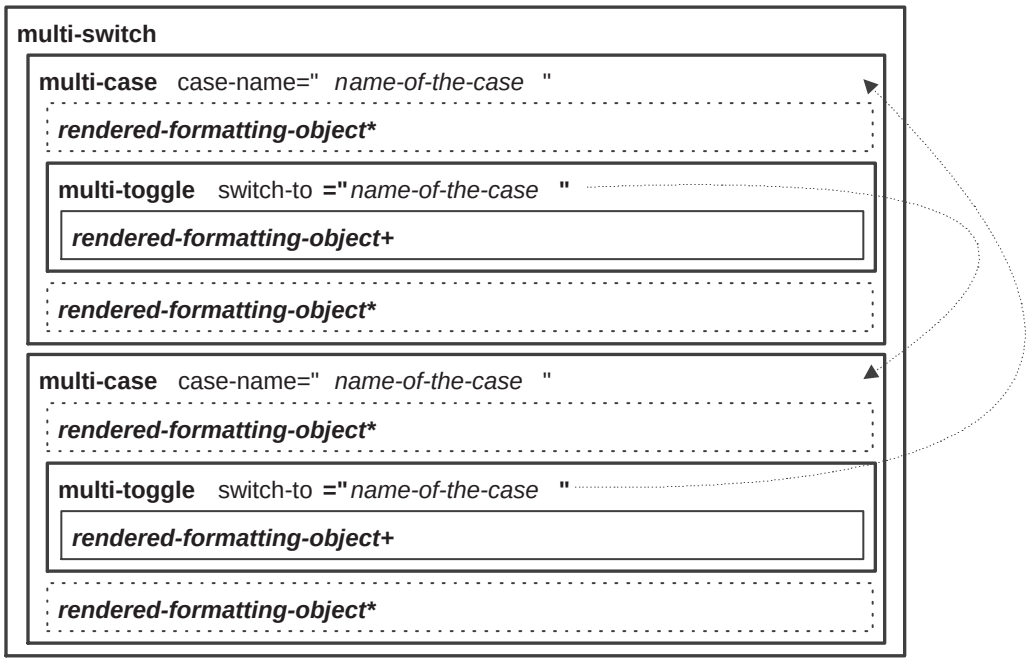
- properties can accommodate an ancestor `<multi-switch>` object being hidden
    - can restore the initial state of the descendent `<multi-switch>` object
    - the default is to preserve the active state in the descendant regardless of the behavior of the ancestor
- can be nested to any depth
- the stylesheet writer doesn't need to know programming but does need to understand basic concepts of changing state as in state machines

Copyright © Crane Softwrights Ltd.

Dynamically changing formatting object sub-trees (cont.)

The XSL-FO instance effectively implements a state machine

- initial state of which object sequence is shown
  - others are hidden
- transitions to subsequent state triggered by operator interaction
  - showing another, hiding the one
- restoration to initial state on ancestral change of state
  - hiding the ancestor, possibly resetting the descendant



 Copyright © Crane Softwrights Ltd.

<multi-switch> Object

Chapter 12 - Interactive objects
Section 2 - Interactively changing the effective flow

Purpose:

- 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

Content:

- (6.9.3) (multi-case+)
- Child object:
  - <multi-case>(6.9.4;378)

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

auto-restore=(7.23.2;450)
id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)

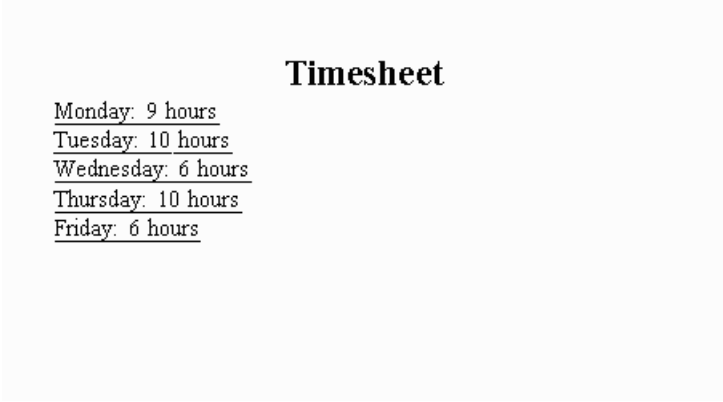
Property of interest:

- auto-restore= is used to indicate whether this <multi-switch> state is restored to its initial state when an ancestral <multi-switch> causes this <multi-switch> to be hidden

Copyright © Crane Softwrights Ltd.

<multi-switch> Object (cont.)

The following example is a mockup from samp/timesht.fo:



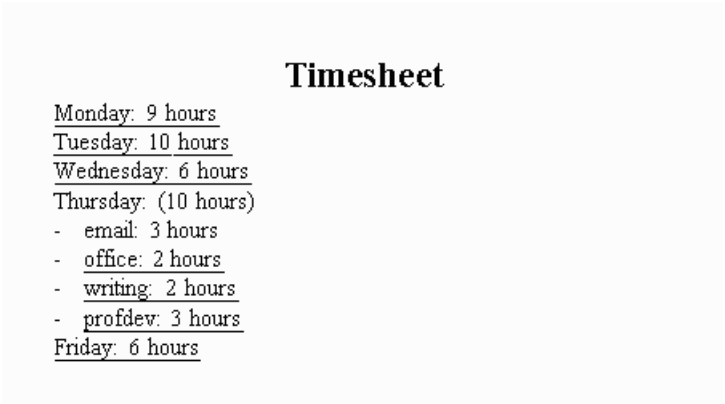
Status of presentation:

- the results are presented without any user input
- the stylesheet chooses to use underscore to indicate to the reader the hot text to click on to expand the entry

 Copyright © Crane Softwrights Ltd.

`<multi-switch>` Object (cont.)

Continuing from `samp/timesht.fo`:



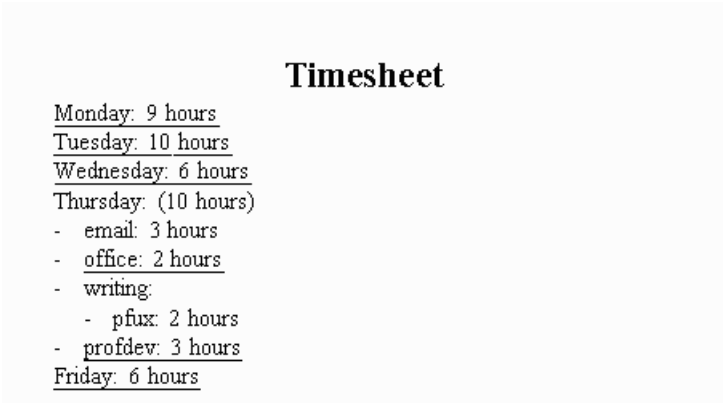
Status of presentation:

- the operator has clicked on "Thursday", thus bringing up the drop-down timesheet values
- the stylesheet again chooses to use underscore to indicate to the reader the hot text to click on to expand the entry
- the stylesheet has used indentation to indicate to the reader that "Thursday" can be clicked on to collapse the entry

Copyright © Crane Softwrights Ltd.

`<multi-switch>` Object (cont.)

Continuing from `samp/timesht.fo`:



Status of presentation:

- the operator has clicked on "writing", thus bringing up the drop-down timesheet values
- collapsing the "Thursday" again will restore the original state of "writing" to be collapsed

 Copyright © Crane Softwrights Ltd.

<multi-switch> Object (cont.)

An excerpt from samp/timesht.fo:

```
<list-block>
  ...
  <multi-switch auto-restore="true">
    <multi-case case-title="Roll-up writing"
                case-name="roll-up" starting-state="show">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block text-decoration="underline">
            <multi-toggle switch-to="roll-down"
                                  >writing: 2 hours</multi-toggle>
          </block>
        </list-item-body>
      </list-item>
    </multi-case>
    <multi-case case-title="Roll-down writing"
                case-name="roll-down" starting-state="hide">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block>
            <multi-toggle switch-to="roll-up"
                                       >writing: </multi-toggle>
          </block>
          <list-block>
            <list-item>
              <list-item-label><block>-</block></list-item-label>
              <list-item-body start-indent="body-start()">
                <block>pfux: 2 hours</block>
              </list-item-body>
            </list-item>
          </list-block>
        </list-item-body>
      </list-item>
    </multi-case>
  </multi-switch>
  ...
</list-block>
```


Copyright © Crane Softwrights Ltd.

<multi-case> Object

Purpose:

- 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
  - comprised of interaction-sensitive (using <multi-toggle>) and interaction-insensitive (using other) formatting objects

Content:

- (6.9.4) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- Referring object:
  - <multi-switch>(6.9.3;373)

Property sets:

- Common Accessibility Properties(7.5;426)

Other required properties:

case-name=(7.23.3;464)

case-title=(7.23.4;464)

Other optional properties:

id=(7.30.8;475)

index-class=(7.24.1;475)

index-key=(7.24.2;475)

starting-state=(7.23.10;498)

Copyright © Crane Softwrights Ltd.

`<multi-case>` Object (cont.)

Again from the excerpt of `samp/timesht.fo`:

```
<list-block>
  ...
  <multi-switch auto-restore="true">
    <multi-case case-title="Roll-up writing"
                case-name="roll-up" starting-state="show">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block text-decoration="underline">
            <multi-toggle switch-to="roll-down"
                                  >writing: 2 hours</multi-toggle>
          </block>
        </list-item-body>
      </list-item>
    </multi-case>
    <multi-case case-title="Roll-down writing"
                case-name="roll-down" starting-state="hide">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block>
            <multi-toggle switch-to="roll-up"
                                       >writing: </multi-toggle>
          </block>
          <list-block>
            <list-item>
              <list-item-label><block>-</block></list-item-label>
              <list-item-body start-indent="body-start()">
                <block>pfux: 2 hours</block>
              </list-item-body>
            </list-item>
          </list-block>
        </list-item-body>
      </list-item>
    </multi-case>
  </multi-switch>
  ...
</list-block>
```


Copyright © Crane Softwrights Ltd.

<multi-toggle> Object

Chapter 12 - Interactive objects
Section 2 - Interactively changing the effective flow

Purpose:

- the definition of those interaction-sensitive objects within a candidate rendered sequence of formatting objects
  - specifies which candidate sequence is to be the rendered sequence as a result of the interaction

Content:

- (6.9.5) (#PCDATA|%inline;|%block;)*
- Child objects (alphabetical):
  - %block;(6.2;71)
  - %inline;(6.2;72)
- must be a descendant of <multi-case>

Property sets:

- Common Accessibility Properties(7.5;426)

Other optional properties:

id=(7.30.8;475)
index-class=(7.24.1;475)
index-key=(7.24.2;475)
switch-to=(7.23.11;499)

Copyright © Crane Softwrights Ltd.

`<multi-toggle>` Object (cont.)

Again from the excerpt `samp/timesht.fo`:

```
<list-block>
  ...
  <multi-switch auto-restore="true">
    <multi-case case-title="Roll-up writing"
                case-name="roll-up" starting-state="show">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block text-decoration="underline">
            <multi-toggle switch-to="roll-down"
                                  >writing: 2 hours</multi-toggle>
          </block>
        </list-item-body>
      </list-item>
    </multi-case>
    <multi-case case-title="Roll-down writing"
                case-name="roll-down" starting-state="hide">
      <list-item>
        <list-item-label><block>-</block></list-item-label>
        <list-item-body start-indent="body-start()">
          <block>
            <multi-toggle switch-to="roll-up"
                                      >writing: </multi-toggle>
          </block>
          <list-block>
            <list-item>
              <list-item-label><block>-</block></list-item-label>
              <list-item-body start-indent="body-start()">
                <block>pfux: 2 hours</block>
              </list-item-body>
            </list-item>
          </list-block>
        </list-item-body>
      </list-item>
    </multi-case>
  </multi-switch>
  ...
</list-block>
```


Copyright © Crane Softwrights Ltd.

Chapter 13 - Where XSL-FO 1 falls short

- Introduction - What is missing in XSL-FO 1?
- Section 1 - Page-related formatting requirements
- Section 2 - Geometry-related formatting requirements
- Section 3 - Alignment of areas

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

Arbitrating between retrieved content

A page may have many candidate markers for retrieval where only a subset is needed

- e.g. a page in a document may have separate areas marked with "Confidential", "Secret" and "Top Secret"
- the objective is to choose from the available markers in scope only the "highest" or any particular one of those found

Retrieved content cannot be compared for algorithmic inclusion or rejection

- one would need to be able to express criteria for choosing between available markers
- the XSL-FO 1.0 expression language doesn't have comparative operators
- one future approach might be the assignment of priority to markers in a class and the retrieval of the marker in scope of a specific or highest priority

Other limitations exist on retrieving markers found on a page

- one cannot count the markers and express the number found
- one cannot work with other than the first or last marker on a page
    - the number of markers between the first and last, if any, is irrelevant

Work around is to render each retrieved content in place with opaque backgrounds

- use `<block-container>` with an absolute position and a different value of `z-index=` for each retrieved content
- utilize the identical position for each absolutely positioned container
- assign the higher index numbers to the "more important" containers
- use an opaque white background to obscure any other value present at the location
- retrieve the words desired from the marker into each container
- only the "top-most" container will be visible, obscuring the containers that are behind
- this strategy was conceived by Eliot Kimber

 Copyright © Crane Softwrights Ltd.

Arbitrating between retrieved content (cont.)

From the `marker-arb.fo` example:

```
  <static-content flow-name="frame-before">
    <block-container z-index="1" absolute-position="absolute"
                     height="12pt" width="3cm" right="0cm">
      <block background-color="white" text-align="end"
             font-family="Courier" font-size="12pt">
        <retrieve-marker retrieve-class-name="classified"
                         retrieve-boundary="page"/>
      </block>
    </block-container>
    <block-container z-index="2" absolute-position="absolute"
                     height="12pt" width="3cm" right="0cm">
      <block background-color="white" text-align="end"
             font-family="Courier" font-size="12pt">
        <retrieve-marker retrieve-class-name="secret"
                         retrieve-boundary="page"/>
      </block>
    </block-container>
    <block-container z-index="3" absolute-position="absolute"
                     height="12pt" width="3cm" right="0cm">
      <block background-color="white" text-align="end"
             font-family="Courier" font-size="12pt">
        <retrieve-marker retrieve-class-name="topsecret"
                         retrieve-boundary="page"/>
      </block>
    </block-container>
  </static-content>
...
    <block space-before="4em">
      <marker marker-class-name="classified">classified</marker>
      This is a classified test</block>
    <block space-before="4em">This is a test</block>
    <block space-before="4em">
      <marker marker-class-name="secret">secret</marker>
      This is a secret test</block>
    <block space-before="4em">This is a test</block>
    <block space-before="4em">
      <marker marker-class-name="topsecret">top secret</marker>
      This is a top secret test</block>
    <block space-before="4em">This is a test</block>
```


Copyright © Crane Softwrights Ltd.

Retrieving markers into the body

- No way to retrieve a marker anywhere except in static content
  - e.g. table caption changes with "continued..." when table overflows a page
    - conditionally putting an indication that the table continues on the next page
- XSL-FO 1.1 allows retrieval into table headers and footers
  - but not elsewhere in the body

Work-around is to put the string in the static content.

 Copyright © Crane Softwrights Ltd.

Line numbering

No way to indicate that lines on the page are to be numbered

- often necessary to count lines within a block, a page, a page sequence or the entire document
- useful during the editing practice for citing where the focus of discussion is in a document

The formatter is responsible for creating lines based on the content flowed into a block

- the stylesheet doesn't know where the formatter will break lines
- no defined area of the line is reserved for containing line numbers

Work around is for the stylesheet to produce as many blocks as required for the lines

- generate line numbers in the stylesheet for each line
  - prefix each block with a line number in a fixed width
  - pair a block the line number with the block with the line content by using a `<list-block>` construct
- may wish to turn off the wrapping of the content in the line to prevent an overflowed line from disrupting the line numbering and apparent spacing

Work around is to paint the background with line numbers

- can use a graphic with line numbers
- can extend perimeter region behind body region and flow line numbers into background
- problem that there is no guarantee the line numbers will line up because of any `space-before=` or `space-after=` or other formatting objects of arbitrary size

Copyright © Crane Softwrights Ltd.

Footnotes

Page-based footnote citations

- it is often desired to restart the numbering of footnotes on each page that has footnotes
- only the formatter knows which pages on which content will be placed
- the stylesheet is responsible for all footnote citations
- the stylesheet can never know when to restart the numbering of footnotes

Column-wide footnote rendering

- a number of users ask for footnotes to grow from the bottom of a column
- this is available as a vendor extension in a number of products

Inline footnote rendering

- a number of users ask for footnote bodies to be flowed inline in the blocks of the footnote reference area
- many short footnotes as block-level constructs will leave an exorbitant amount of white-space
- many short footnotes as inline-level constructs would make for very compact and efficient use of the footnote reference area
- this is being considered as a vendor extension in some products

Multiple references to the same footnote

- cannot cite the same footnote with two references expecting the footnote to be on the same page as both references
- using an empty footnote body for the second and subsequent footnotes will trigger unnecessary `xsl-footnote-separator` sub-regions when there are no other footnotes that would need such a region on the page

 Copyright © Crane Softwrights Ltd.

Determining the current formatted location

A common request is to know the current running position on a page when transforming

- wanting to know the page number
- wanting to know the formatted location on the page

The two-phase arms-length architecture of XSL does not permit detection of the current formatted position

- XSLT produces a standalone transformation result before an XSL-FO engine can act on it
- cannot predetermine the location of the formatted position when doing the transformation
    - different implementations can have different interpretations based on built-in defaults

Work-around is to include in the XSL-FO instance as many contingencies as possible

- the XSL-FO engine can only work with the information found within the expression of formatting intent

Copyright © Crane Softwrights Ltd.

Choosing page geometry based on page content

It is not possible in native XSL-FO to alternate page geometries based on content in the middle of a page sequence

- may wish to select an alternative page geometry based on the formatted result
  - e.g. use an alternative reference orientation for a table that is "too wide"
- may wish to select an alternative page geometry (e.g. with wider margins) to avoid breaking a keep condition

A workaround is the freely available PSMI semantic (see page 272)

- not a formatter-managed solution so some nuances cannot be addressed
  - e.g. cannot choose which page master to use based on the retrieved dimensions of a rendered graphic
  - e.g. cannot choose based on the rendered width of a table
- viable workaround only when the stylesheet can make the decision to use an alternative geometry

 Copyright © Crane Softwrights Ltd.

Aligning flows and marks

It is a common need to align parallel streams of content in actual or virtual columns on the page

- e.g. line numbers to the edge of a block of lines
- e.g. marginalia associated with content found in the middle of a block of lines
- e.g. aligned blocks or table rows across face-to-face pages

1.0 No support of change bars

1.1 Change bars supported

A work around may be to use tables with invisible borders

- turn on border edges to mimic change bars
- problem at all times because one is unable to judge in the transformation the extent of the formatted result determined by the formatter
- hidden rows align content in sister columns

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

<xsl:attribute-set> instruction

<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

 Copyright © Crane Softwrights Ltd.

Simpler list and footnote structures in 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

Copyright © Crane Softwrights Ltd.

Common errors writing expressions

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

 Copyright © Crane Softwrights Ltd.

Common errors writing expressions (cont.)

Consider the situation of arithmetic calculations with lengths:

```
<xsl:template name="test">
<xsl:variable name="num" select="100"/>
<xsl:variable name="medium-font" select="'10pt'"/>

<block space-before="10pt div 2">
First test: <xsl:value-of select="$num div 2"/>
</block>

<block space-before="{$medium-font} div 2">
Second test
</block>

<block space-before="{$medium-font div 2}">
Third test
</block>

<block>
  <xsl:attribute name="space-before">
    <xsl:value-of select="$medium-font div 2"/>
  </xsl:attribute>
  Fourth test
</block>

<block>
  <xsl:attribute name="space-before">
    <xsl:value-of select="$medium-font"/> div 2<xsl:text/>
  </xsl:attribute>
  Fifth test
</block>
</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

Copyright © Crane Softwrights Ltd.

Annex B - XSL-FO expressions

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

 Copyright © Crane Softwrights Ltd.

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.

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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            ::= '"' [^"]* '"'
                          | "'" [^']* "'"
```


Copyright © Crane Softwrights Ltd.

```
[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)
```


Copyright © Crane Softwrights Ltd.

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"`

Copyright © Crane Softwrights Ltd.

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()`

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Functions summarized by name

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

Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

Annex C - XSL-FO object summary

- Introduction - Objects
- Section 1 - Formatting objects

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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.

 Copyright © Crane Softwrights Ltd.

Objects summarized by name

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

`<basic-link>`(6.9.2;161) (visual: extended; aural: extended)
- (#PCDATA|%inline;|%block;)*

`<bidi-override>`(6.6.2;361) (visual: extended; aural: basic)
- (#PCDATA|%inline;|%block;)*

`<block>`(6.5.2;109) (visual: basic; aural: basic)
- (#PCDATA|%inline;|%block;)*

`<block-container>`(6.5.3;227) (visual: extended; aural: basic)
- (%block;)+

`<bookmark>`(6.11.2;300) (visual: extended; aural: extended)
- (bookmark-title,bookmark*)

`<bookmark-title>`(6.11.3;301) (visual: extended; aural: extended)
- (#PCDATA)

`<bookmark-tree>`(6.11.1;298) (visual: extended; aural: extended)
- (bookmark+)

`<change-bar-begin>`(6.13.2;344) (visual: extended; aural: extended)
- EMPTY

`<change-bar-end>`(6.13.3;345) (visual: extended; aural: extended)
- EMPTY

`<character>`(6.6.3;350) (visual: basic; aural: basic)
- EMPTY

`<color-profile>`(6.4.4;347) (visual: extended; aural: N/A)
- EMPTY

`<conditional-page-master-reference>`(6.4.12;290) (visual: extended; aural: extended)
- EMPTY

`<declarations>`(6.4.3;348) (visual: basic; aural: basic)
- (color-profile)*

`<external-graphic>`(6.6.5;152) (visual: basic; aural: basic)
- EMPTY

`<float>`(6.12.2;214) (visual: extended; aural: extended)
- (%block;)+

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

`<flow-assignment>`(6.4.23;247) (visual: extended; aural: extended)
- (flow-source-list,flow-target-list)

Copyright © Crane Softwrights Ltd.

- <flow-map>(6.4.22;246) (visual: extended; aural: extended)
  - (flow-assignment+)
- <flow-name-specifier>(6.4.25;249) (visual: extended; aural: extended)
  - EMPTY
- <flow-source-list>(6.4.24;248) (visual: extended; aural: extended)
  - (flow-name-specifier+)
- <flow-target-list>(6.4.26;250) (visual: extended; aural: extended)
  - (region-name-specifier+)
- <folio-prefix>(6.6.13;256) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*
- <folio-suffix>(6.6.14;258) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*

<footnote>(6.12.3;219) (visual: extended; aural: extended)
  - (inline,footnote-body)

<footnote-body>(6.12.4;222) (visual: extended; aural: extended)
  - (%block;)+

- <index-key-reference>(6.10.6;313) (visual: extended; aural: extended)
  - (index-page-number-prefix?,index-page-number-suffix?)
- <index-page-citation-list>(6.10.7;309) (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;311) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*
- <index-page-citation-range-separator>(6.10.9;312) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*
- <index-page-number-prefix>(6.10.2;314) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*
- <index-page-number-suffix>(6.10.3;315) (visual: extended; aural: extended)
  - (#PCDATA|%inline;)*
- <index-range-begin>(6.10.4;305) (visual: extended; aural: extended)
  - EMPTY
- <index-range-end>(6.10.5;307) (visual: extended; aural: extended)
  - EMPTY

<initial-property-set>(6.6.4;111) (visual: extended; aural: basic)
  - EMPTY

<inline>(6.6.7;112) (visual: basic; aural: basic)
  - (#PCDATA|%inline;|%block;)*

 Copyright © Crane Softwrights Ltd.

`<inline-container>`(6.6.8;228) (visual: extended; aural: extended)

- (`%block;`)+

`<instream-foreign-object>`(6.6.6;155) (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;169) (visual: basic; aural: basic)

- (`#PCDATA`|`%inline;`)*

`<list-block>`(6.8.2;140) (visual: basic; aural: basic)

- (`list-item`+)

`<list-item>`(6.8.3;142) (visual: basic; aural: basic)

- (`list-item-label`,`list-item-body`)

`<list-item-body>`(6.8.4;146) (visual: basic; aural: basic)

- (`%block;`)+

`<list-item-label>`(6.8.5;144) (visual: extended; aural: basic)

- (`%block;`)+

`<marker>`(6.13.5;264) (visual: extended; aural: extended)

- (`#PCDATA`|`%inline;`|`%block;`)*

`<multi-case>`(6.9.4;378) (visual: basic; aural: basic)

- (`#PCDATA`|`%inline;`|`%block;`)*

`<multi-properties>`(6.9.6;367) (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;369) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)

- `EMPTY`

`<multi-switch>`(6.9.3;373) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)

- (`multi-case`+)

`<multi-toggle>`(6.9.5;380) (visual: extended, need not be implemented for extended conformance for non-interactive media; aural: extended)

- (`#PCDATA`|`%inline;`|`%block;`)*

`<page-number>`(6.6.10;255) (visual: basic; aural: extended)

- `EMPTY`

`<page-number-citation>`(6.6.11;113) (visual: extended; aural: extended)

- `EMPTY`

`<page-number-citation-last>`(6.6.12;114) (visual: extended; aural: extended)

- `EMPTY`

Copyright © Crane Softwrights Ltd.

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

`<page-sequence-master>`(6.4.8;283) (visual: basic; aural: basic)
- (`single-page-master-reference`|`repeatable-page-master-reference`|`repeatable-page-master-alternatives`)+

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

`<region-after>`(6.4.16;239) (visual: extended; aural: extended)
- EMPTY

`<region-before>`(6.4.15;238) (visual: extended; aural: extended)
- EMPTY

`<region-body>`(6.4.14;121) (visual: basic; aural: basic)
- EMPTY

`<region-end>`(6.4.18;242) (visual: extended; aural: extended)
- EMPTY

1 `<region-name-specifier>`(6.4.27;251) (visual: extended; aural: extended)
- EMPTY

`<region-start>`(6.4.17;241) (visual: extended; aural: extended)
- EMPTY

`<repeatable-page-master-alternatives>`(6.4.11;288) (visual: extended; aural: extended)
- (`conditional-page-master-reference`+)

`<repeatable-page-master-reference>`(6.4.10;287) (visual: basic; aural: basic)
- EMPTY

`<retrieve-marker>`(6.13.6;267) (visual: extended; aural: extended)
- EMPTY

1 `<retrieve-table-marker>`(6.13.7;269) (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`)+)

1 `<scaling-value-citation>`(6.6.15;159) (visual: extended; aural: extended)
- EMPTY

`<simple-page-master>`(6.4.13;119) (visual: basic; aural: basic)
- (`region-body`+,`region-before`?,`region-after`?,`region-start`?,`region-end`?)

`<single-page-master-reference>`(6.4.9;286) (visual: basic; aural: basic)
- EMPTY

`<static-content>`(6.4.20;253) (visual: extended; aural: extended)
- (`%block;`)+

 Copyright © Crane Softwrights Ltd.

`<table>`(6.7.3;192) (visual: basic; aural: basic)
- (`table-column*`,`table-header?`,`table-footer?`,`table-body+`)

`<table-and-caption>`(6.7.2;187) (visual: basic; aural: basic)
- (`table-caption?`,`table`)

`<table-body>`(6.7.8;199) (visual: basic; aural: basic)
- (`table-row+`|`table-cell+`)

`<table-caption>`(6.7.5;190) (visual: extended; aural: extended)
- (`%block;`)+

`<table-cell>`(6.7.10;203) (visual: basic; aural: basic)
- (`%block;`)+

`<table-column>`(6.7.4;194) (visual: basic; aural: basic)
- `EMPTY`

`<table-footer>`(6.7.7;198) (visual: extended; aural: extended)
- (`table-row+`|`table-cell+`)

`<table-header>`(6.7.6;196) (visual: basic; aural: basic)
- (`table-row+`|`table-cell+`)

`<table-row>`(6.7.9;201) (visual: basic; aural: basic)
- (`table-cell+`)

`<title>`(6.4.21;124) (visual: extended; aural: extended)
- (`#PCDATA`|`%inline;`)*

`<wrapper>`(6.13.4;108) (visual: basic; aural: basic)
- (`#PCDATA`|`%inline;`|`%block;`)*

Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

- <root>(6.4.2;61)
  - <layout-master-set>(6.4.7;63)
    - <simple-page-master>(6.4.13;119)
      - <region-body/>(6.4.14;121)
      - <region-before/>(6.4.15;238)
      - <region-after/>(6.4.16;239)
      - <region-start/>(6.4.17;241)
      - <region-end/>(6.4.18;242)
    - …
    - <page-sequence-master>(6.4.8;283)
      - <single-page-master-reference/>(6.4.9;286)
      - …
      - <repeatable-page-master-reference/>(6.4.10;287)
      - …
      - <repeatable-page-master-alternatives>(6.4.11;288)
        - <conditional-page-master-reference/>(6.4.12;290)
    - …
    - ¦ <flow-map>(6.4.22;246)
      - ¦ <flow-assignment>(6.4.23;247)
        - ¦ <flow-source-list>(6.4.24;248)
          - ¦ <flow-name-specifier/>(6.4.25;249)
        - ¦ <flow-target-list>(6.4.26;250)
          - ¦ <region-name-specifier/>(6.4.27;251)
  - <declarations>(6.4.3;348)
    - <color-profile/>(6.4.4;347)
  - ¦ <bookmark-tree>(6.11.1;298)
    - ¦ <bookmark>(6.11.2;300)
      - ¦ <bookmark-title>(6.11.3;301)
      - ¦ <bookmark>(6.11.2;300)
      - …
  - ¦ <page-sequence-wrapper>(6.4.6;67)
    - <page-sequence>(6.4.5;65)
    - ¦ <page-sequence-wrapper>(6.4.6;67)
    - …
  - <page-sequence>(6.4.5;65)
    - <title>(6.4.21;124)
    - ¦ <folio-prefix>(6.6.13;256)
      - %inline;
    - ¦ <folio-suffix>(6.6.14;258)
      - %inline;
    - <static-content>(6.4.20;253)
      - <retrieve-marker/>(6.13.6;267)
      - …
      - <page-number/>(6.6.10;255)
    - <flow>(6.4.19;68)
      - <block-container>(6.5.3;227)

 Copyright © Crane Softwrights Ltd.

- %block;
- …
- <float>(6.12.2;214)
  - %block;
- …
- <table-and-caption>(6.7.2;187)
  - <table-caption>(6.7.5;190)
  - <table>(6.7.3;192)
    - <table-column/>(6.7.4;194)
    - <table-header>(6.7.6;196)
      - <retrieve-table-marker/>(6.13.7;269)
    - <table-footer>(6.7.7;198)
      - <retrieve-table-marker/>(6.13.7;269)
    - <table-body>(6.7.8;199)
      - <table-row>(6.7.9;201)
      - <table-cell>(6.7.10;203)
        - %block;
- …
- <list-block>(6.8.2;140)
  - <list-item>(6.8.3;142)
    - <list-item-label>(6.8.5;144)
      - %block;
    - <list-item-body>(6.8.4;146)
      - %block;
- …
- <wrapper>(6.13.4;108)
  - %block;
- …

Copyright © Crane Softwrights Ltd.

- <block>(6.5.2;109)
  - <marker>(6.13.5;264)
  - <initial-property-set/>(6.6.4;111)
  - <inline-container>(6.6.8;228)
    - %block;
  - …
  - <basic-link>(6.9.2;161)
  - …
  - <bidi-override>(6.6.2;361)
  - …
  - ¦ <change-bar-begin/>(6.13.2;344)
  - …
  - ¦ <change-bar-end/>(6.13.3;345)
  - …
  - <character/>(6.6.3;350)
  - …
  - <external-graphic/>(6.6.5;152)
  - …
  - <footnote>(6.12.3;219)
    - <footnote-body>(6.12.4;222)
      - %block;
  - …
  - ¦ <index-page-citation-list>(6.10.7;309)
    - ¦ <index-page-citation-list-separator>(6.10.8;311)
      - %inline;
    - ¦ <index-page-citation-range-separator>(6.10.9;312)
      - %inline;
    - ¦ <index-key-reference>(6.10.6;313)
      - ¦ <index-page-number-prefix>(6.10.2;314)
        - %inline;
      - ¦ <index-page-number-suffix>(6.10.3;315)
        - %inline;
  - …
  - ¦ <index-range-begin/>(6.10.4;305)
  - …
  - ¦ <index-range-end/>(6.10.5;307)
  - …
  - <inline>(6.6.7;112)
  - …
  - <instream-foreign-object>(6.6.6;155)
  - …
  - <leader>(6.6.9;169)
  - …
  - <page-number-citation/>(6.6.11;113)
  - …
  - ¦ <page-number-citation-last/>(6.6.12;114)

 Copyright © Crane Softwrights Ltd.

- …
- 1 <scaling-value-citation/>(6.6.15;159)
- …
- <wrapper>(6.13.4;108)
    - %inline;
- …
- <multi-switch>(6.9.3;373)
    - <multi-case>(6.9.4;378)
        - <multi-toggle>(6.9.5;380)
- …
- <multi-properties>(6.9.6;367)
    - <multi-property-set/>(6.9.7;369)

Copyright © Crane Softwrights Ltd.

Objects summarized by type

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

Block-level Formatting Objects(6.5)

- `<block>`(6.5.2;109)
- `<block-container>`(6.5.3;227)

Declarations and Pagination and Layout Formatting Objects(6.4)

- `<color-profile>`(6.4.4;347)
- `<conditional-page-master-reference>`(6.4.12;290)
- `<declarations>`(6.4.3;348)
- `<flow>`(6.4.19;68)
- 1 `<flow-assignment>`(6.4.23;247)
- 1 `<flow-map>`(6.4.22;246)
- 1 `<flow-name-specifier>`(6.4.25;249)
- 1 `<flow-source-list>`(6.4.24;248)
- 1 `<flow-target-list>`(6.4.26;250)
- `<layout-master-set>`(6.4.7;63)
- `<page-sequence>`(6.4.5;65)
- `<page-sequence-master>`(6.4.8;283)
- 1 `<page-sequence-wrapper>`(6.4.6;67)
- `<region-after>`(6.4.16;239)
- `<region-before>`(6.4.15;238)
- `<region-body>`(6.4.14;121)
- `<region-end>`(6.4.18;242)
- 1 `<region-name-specifier>`(6.4.27;251)
- `<region-start>`(6.4.17;241)
- `<repeatable-page-master-alternatives>`(6.4.11;288)
- `<repeatable-page-master-reference>`(6.4.10;287)
- `<root>`(6.4.2;61)
- `<simple-page-master>`(6.4.13;119)
- `<single-page-master-reference>`(6.4.9;286)
- `<static-content>`(6.4.20;253)
- `<title>`(6.4.21;124)

Dynamic Effects: Link and Multi Formatting Objects(6.9)

- `<basic-link>`(6.9.2;161)
- `<multi-case>`(6.9.4;378)
- `<multi-properties>`(6.9.6;367)
- `<multi-property-set>`(6.9.7;369)
- `<multi-switch>`(6.9.3;373)
- `<multi-toggle>`(6.9.5;380)

Copyright © Crane Softwrights Ltd.

Formatting Objects for Bookmarks(6.11)

- ¶ <bookmark>(6.11.2;300)
- ¶ <bookmark-title>(6.11.3;301)
- ¶ <bookmark-tree>(6.11.1;298)

Formatting Objects for Indexing(6.10)

- ¶ <index-key-reference>(6.10.6;313)
- ¶ <index-page-citation-list>(6.10.7;309)
- ¶ <index-page-citation-list-separator>(6.10.8;311)
- ¶ <index-page-citation-range-separator>(6.10.9;312)
- ¶ <index-page-number-prefix>(6.10.2;314)
- ¶ <index-page-number-suffix>(6.10.3;315)
- ¶ <index-range-begin>(6.10.4;305)
- ¶ <index-range-end>(6.10.5;307)

Formatting Objects for Lists(6.8)

- <list-block>(6.8.2;140)
- <list-item>(6.8.3;142)
- <list-item-body>(6.8.4;146)
- <list-item-label>(6.8.5;144)

Formatting Objects for Tables(6.7)

- <table>(6.7.3;192)
- <table-and-caption>(6.7.2;187)
- <table-body>(6.7.8;199)
- <table-caption>(6.7.5;190)
- <table-cell>(6.7.10;203)
- <table-column>(6.7.4;194)
- <table-footer>(6.7.7;198)
- <table-header>(6.7.6;196)
- <table-row>(6.7.9;201)

Inline-level Formatting Objects(6.6)

- <bidi-override>(6.6.2;361)
- <character>(6.6.3;350)
- <external-graphic>(6.6.5;152)
- ¶ <folio-prefix>(6.6.13;256)
- ¶ <folio-suffix>(6.6.14;258)
- <initial-property-set>(6.6.4;111)
- <inline>(6.6.7;112)
- <inline-container>(6.6.8;228)
- <instream-foreign-object>(6.6.6;155)
- <leader>(6.6.9;169)
- <page-number>(6.6.10;255)
- <page-number-citation>(6.6.11;113)
- ¶ <page-number-citation-last>(6.6.12;114)
- ¶ <scaling-value-citation>(6.6.15;159)

Copyright © Crane Softwrights Ltd.

Other Formatting Objects(6.13)

- 1.1 `<change-bar-begin>`(6.13.2;344)
- 1.1 `<change-bar-end>`(6.13.3;345)
- `<marker>`(6.13.5;264)
- `<retrieve-marker>`(6.13.6;267)
- 1.1 `<retrieve-table-marker>`(6.13.7;269)
- `<wrapper>`(6.13.4;108)

Out-of-Line Formatting Objects(6.12)

- `<float>`(6.12.2;214)
- `<footnote>`(6.12.3;219)
- `<footnote-body>`(6.12.4;222)

 Copyright © Crane Softwrights Ltd.

Annex D - XSL-FO property summaries

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

Copyright © Crane Softwrights Ltd.

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.

 Copyright © Crane Softwrights Ltd.

Common Absolute Position Properties

Properties (7.6):

`absolute-position=`(7.6.1;448)
`bottom=`(7.6.4;463)
`left=`(7.6.5;479)
`right=`(7.6.3;493)
`top=`(7.6.2;502)

Referring object:

`<block-container>`(6.5.3;227)

Shorthand influencing the above properties:

`position=`(7.31.20;490)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Common Accessibility Properties

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

Properties (7.5):

`role=`(7.5.2;494)
`source-document=`(7.5.1;495)

Referring objects:

`<basic-link>`(6.9.2;161)
`<block>`(6.5.2;109)
- `<bookmark>`(6.11.2;300)
- `<bookmark-title>`(6.11.3;301)
- `<change-bar-begin>`(6.13.2;344)
- `<change-bar-end>`(6.13.3;345)

`<external-graphic>`(6.6.5;152)
`<footnote>`(6.12.3;219)
`<footnote-body>`(6.12.4;222)
`<initial-property-set>`(6.6.4;111)
`<inline>`(6.6.7;112)
`<instream-foreign-object>`(6.6.6;155)
`<leader>`(6.6.9;169)
`<list-block>`(6.8.2;140)
`<list-item>`(6.8.3;142)
`<list-item-body>`(6.8.4;146)
`<list-item-label>`(6.8.5;144)
`<multi-case>`(6.9.4;378)
`<multi-properties>`(6.9.6;367)
`<multi-switch>`(6.9.3;373)
`<multi-toggle>`(6.9.5;380)
`<page-number>`(6.6.10;255)
`<page-number-citation>`(6.6.11;113)
- `<page-number-citation-last>`(6.6.12;114)

`<root>`(6.4.2;61)
- `<scaling-value-citation>`(6.6.15;159)

`<table>`(6.7.3;192)
`<table-and-caption>`(6.7.2;187)
`<table-body>`(6.7.8;199)
`<table-caption>`(6.7.5;190)
`<table-cell>`(6.7.10;203)
`<table-footer>`(6.7.7;198)
`<table-header>`(6.7.6;196)
`<table-row>`(6.7.9;201)
`<title>`(6.4.21;124)

Copyright © Crane Softwrights Ltd.

Common Aural Properties

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

Properties (7.7):

azimuth=(7.7.1;450)
cue-after=(7.7.2;467)
cue-before=(7.7.3;467)
elevation=(7.7.4;468)
pause-after=(7.7.5;488)
pause-before=(7.7.6;489)
pitch=(7.7.7;489)
pitch-range=(7.7.8;489)
play-during=(7.7.9;490)
richness=(7.7.10;493)
speak=(7.7.11;497)
speak-header=(7.7.12;497)
speak-numeral=(7.7.13;497)
speak-punctuation=(7.7.14;497)
speech-rate=(7.7.15;498)
stress=(7.7.16;499)
voice-family=(7.7.17;503)
volume=(7.7.18;503)

Referring objects:

<basic-link>(6.9.2;161)
<bidi-override>(6.6.2;361)
<block>(6.5.2;109)
1 <change-bar-begin>(6.13.2;344)
1 <change-bar-end>(6.13.3;345)
<character>(6.6.3;350)
<external-graphic>(6.6.5;152)
<initial-property-set>(6.6.4;111)
<inline>(6.6.7;112)
<instream-foreign-object>(6.6.6;155)
<leader>(6.6.9;169)
<list-block>(6.8.2;140)
<list-item>(6.8.3;142)
<page-number>(6.6.10;255)
<page-number-citation>(6.6.11;113)
1 <page-number-citation-last>(6.6.12;114)
1 <scaling-value-citation>(6.6.15;159)
<table>(6.7.3;192)
<table-and-caption>(6.7.2;187)
<table-body>(6.7.8;199)
<table-caption>(6.7.5;190)
<table-cell>(6.7.10;203)
<table-footer>(6.7.7;198)
<table-header>(6.7.6;196)
<table-row>(6.7.9;201)
<title>(6.4.21;124)

Shorthands influencing the above properties:

cue=(7.31.12;467)
pause=(7.31.19;488)

Copyright © Crane Softwrights Ltd.

Common Border, Padding, and Background Properties

Properties (7.8):

`background-attachment=`(7.8.1;450)
`background-color=`(7.8.2;451)
`background-image=`(7.8.3;451)
`background-position-horizontal=`(7.8.5;452)
`background-position-vertical=`(7.8.6;452)
`background-repeat=`(7.8.4;452)
`border-after-color=`(7.8.10;453)
`border-after-style=`(7.8.11;454)
`border-after-width=`(7.8.12;454)
`border-before-color=`(7.8.7;455)
`border-before-style=`(7.8.8;455)
`border-before-width=`(7.8.9;455)
`border-bottom-color=`(7.8.22;456)
`border-bottom-style=`(7.8.23;456)
`border-bottom-width=`(7.8.24;456)
`border-end-color=`(7.8.16;457)
`border-end-style=`(7.8.17;457)
`border-end-width=`(7.8.18;458)
`border-left-color=`(7.8.25;458)
`border-left-style=`(7.8.26;458)
`border-left-width=`(7.8.27;459)
`border-right-color=`(7.8.28;459)
`border-right-style=`(7.8.29;459)
`border-right-width=`(7.8.30;460)
`border-start-color=`(7.8.13;460)
`border-start-style=`(7.8.14;461)
`border-start-width=`(7.8.15;461)
`border-top-color=`(7.8.19;462)
`border-top-style=`(7.8.20;462)
`border-top-width=`(7.8.21;462)
`padding-after=`(7.8.32;484)
`padding-before=`(7.8.31;485)
`padding-bottom=`(7.8.36;485)
`padding-end=`(7.8.34;485)
`padding-left=`(7.8.37;486)
`padding-right=`(7.8.38;486)
`padding-start=`(7.8.33;486)
`padding-top=`(7.8.35;487)

Referring objects:

`<basic-link>`(6.9.2;161)
`<block>`(6.5.2;109)
`<block-container>`(6.5.3;227)
`<character>`(6.6.3;350)
`<external-graphic>`(6.6.5;152)
`<initial-property-set>`(6.6.4;111)
`<inline>`(6.6.7;112)
`<inline-container>`(6.6.8;228)
`<instream-foreign-object>`(6.6.6;155)
`<leader>`(6.6.9;169)
`<list-block>`(6.8.2;140)
`<list-item>`(6.8.3;142)
`<page-number>`(6.6.10;255)
`<page-number-citation>`(6.6.11;113)
`<page-number-citation-last>`(6.6.12;114)
`<region-after>`(6.4.16;239)
`<region-before>`(6.4.15;238)
`<region-body>`(6.4.14;121)
`<region-end>`(6.4.18;242)
`<region-start>`(6.4.17;241)
`<scaling-value-citation>`(6.6.15;159)
`<table>`(6.7.3;192)
`<table-and-caption>`(6.7.2;187)
`<table-body>`(6.7.8;199)
`<table-caption>`(6.7.5;190)
`<table-cell>`(6.7.10;203)
`<table-column>`(6.7.4;194)
`<table-footer>`(6.7.7;198)
`<table-header>`(6.7.6;196)
`<table-row>`(6.7.9;201)
`<title>`(6.4.21;124)

Shorthands influencing the above properties:

`background=`(7.31.1;450)
`background-position=`(7.31.2;451)
`border=`(7.31.3;453)
`border-right=`(7.31.7;459)
`border-style=`(7.31.8;461)
`border-top=`(7.31.10;462)

Copyright © Crane Softwrights Ltd.

`border-bottom=`(7.31.4;456)
`border-color=`(7.31.5;457)
`border-left=`(7.31.6;458)
`border-width=`(7.31.11;463)
`padding=`(7.31.15;484)

Copyright © Crane Softwrights Ltd.

Common Font Properties

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

Properties (7.9):

`font-family=`(7.9.2;471)
`font-selection-strategy=`(7.9.3;471)
`font-size=`(7.9.4;471)
`font-size-adjust=`(7.9.6;471)
`font-stretch=`(7.9.5;472)
`font-style=`(7.9.7;472)
`font-variant=`(7.9.8;472)
`font-weight=`(7.9.9;472)

Referring objects:

`<bidi-override>`(6.6.2;361)
`<block>`(6.5.2;109)
`<character>`(6.6.3;350)
`<initial-property-set>`(6.6.4;111)
`<inline>`(6.6.7;112)
`<leader>`(6.6.9;169)
`<page-number>`(6.6.10;255)
`<page-number-citation>`(6.6.11;113)
`<page-number-citation-last>`(6.6.12;114)
`<scaling-value-citation>`(6.6.15;159)
`<title>`(6.4.21;124)

Shorthand influencing the above properties:

`font=`(7.31.13;470)

Copyright © Crane Softwrights Ltd.

Common Hyphenation Properties

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

Properties (7.10):

`country=`(7.10.1;467)
`hyphenate=`(7.10.4;474)
`hyphenation-character=`(7.10.5;474)
`hyphenation-push-character-count=`(7.10.6;474)
`hyphenation-remain-character-count=`(7.10.7;475)
`language=`(7.10.2;478)
`script=`(7.10.3;495)

Referring objects:

`<block>`(6.5.2;109)
`<character>`(6.6.3;350)

Shorthand influencing the above properties:

1 `xml:lang=`(7.31.24;505)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Common Margin Properties-Block

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

Properties (7.11):

`end-indent=`(7.11.8;469)
`margin-bottom=`(7.11.2;480)
`margin-left=`(7.11.3;481)
`margin-right=`(7.11.4;481)
`margin-top=`(7.11.1;481)
`space-after=`(7.11.6;496)
`space-before=`(7.11.5;496)
`start-indent=`(7.11.7;498)

Referring objects:

`<block>`(6.5.2;109)
`<block-container>`(6.5.3;227)
`<list-block>`(6.8.2;140)
`<list-item>`(6.8.3;142)
`<region-body>`(6.4.14;121)
`<simple-page-master>`(6.4.13;119)
`<table>`(6.7.3;192)
`<table-and-caption>`(6.7.2;187)

Shorthand influencing the above properties:

`margin=`(7.31.14;480)

Copyright © Crane Softwrights Ltd.

Common Margin Properties-Inline

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

Properties (7.12):

`space-end=`(7.12.5;496)
`space-start=`(7.12.6;496)

Referring objects:

`<basic-link>`(6.9.2;161)
`<character>`(6.6.3;350)
`<external-graphic>`(6.6.5;152)
`<inline>`(6.6.7;112)
`<inline-container>`(6.6.8;228)
`<instream-foreign-object>`(6.6.6;155)
`<leader>`(6.6.9;169)
`<page-number>`(6.6.10;255)
`<page-number-citation>`(6.6.11;113)
`<page-number-citation-last>`(6.6.12;114)
`<scaling-value-citation>`(6.6.15;159)
`<title>`(6.4.21;124)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

Common Relative Position Properties

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

Property (7.13):

`relative-position=`(7.13.5;492)

Referring objects:

`<basic-link>`(6.9.2;161)
`<bidi-override>`(6.6.2;361)
`<block>`(6.5.2;109)
`<character>`(6.6.3;350)
`<external-graphic>`(6.6.5;152)
`<initial-property-set>`(6.6.4;111)
`<inline>`(6.6.7;112)
`<inline-container>`(6.6.8;228)
`<instream-foreign-object>`(6.6.6;155)
`<leader>`(6.6.9;169)
`<list-block>`(6.8.2;140)
`<list-item>`(6.8.3;142)
`<page-number>`(6.6.10;255)
`<page-number-citation>`(6.6.11;113)
- `<page-number-citation-last>`(6.6.12;114)
- `<scaling-value-citation>`(6.6.15;159)

`<table>`(6.7.3;192)
`<table-and-caption>`(6.7.2;187)
`<table-body>`(6.7.8;199)
`<table-caption>`(6.7.5;190)
`<table-cell>`(6.7.10;203)
`<table-footer>`(6.7.7;198)
`<table-header>`(6.7.6;196)
`<table-row>`(6.7.9;201)

Shorthand influencing the above property:

`position=`(7.31.20;490)

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

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.

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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 "∂".

<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>

 Copyright © Crane Softwrights Ltd.

- 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 '0pt'.)

<space>

Copyright © Crane Softwrights Ltd.

- 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.

 Copyright © Crane Softwrights Ltd.

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`)

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.

Copyright © Crane Softwrights Ltd.

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/`)

 Copyright © Crane Softwrights Ltd.

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`)

Copyright © Crane Softwrights Ltd.

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

 Copyright © Crane Softwrights Ltd.

Inherited properties

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

The following traits are inherited from ancestral property specifications:

- `allowed-height-scale=`(7.15.1;449)
- `allowed-width-scale=`(7.15.2;449)

`auto-restore=`(7.23.2;450)
`azimuth=`(7.7.1;450)
`border-collapse=`(7.28.3;457)
`border-separation=`(7.28.5;460)
`border-spacing=`(7.31.9;460)
`caption-side=`(7.28.7;463)

- `change-bar-color=`(7.30.2;464)
- `change-bar-offset=`(7.30.3;464)
- `change-bar-placement=`(7.30.4;464)
- `change-bar-style=`(7.30.5;464)
- `change-bar-width=`(7.30.6;465)

`color=`(7.18.1;465)
`country=`(7.10.1;467)
`direction=`(7.29.1;468)
`display-align=`(7.14.4;468)
`elevation=`(7.7.4;468)
`empty-cells=`(7.28.10;469)
`end-indent=`(7.11.8;469)
`font=`(7.31.13;470)
`font-family=`(7.9.2;471)
`font-selection-strategy=`(7.9.3;471)
`font-size=`(7.9.4;471)
`font-size-adjust=`(7.9.6;471)
`font-stretch=`(7.9.5;472)
`font-style=`(7.9.7;472)
`font-variant=`(7.9.8;472)
`font-weight=`(7.9.9;472)
`glyph-orientation-horizontal=`(7.29.2;473)
`glyph-orientation-vertical=`(7.29.3;473)
`hyphenate=`(7.10.4;474)
`hyphenation-character=`(7.10.5;474)
`hyphenation-keep=`(7.16.1;474)
`hyphenation-ladder-count=`(7.16.2;474)
`hyphenation-push-character-count=`(7.10.6;474)
`hyphenation-remain-character-count=`(7.10.7;475)

- `intrinsic-scale-value=`(7.30.9;476)

`intrusion-displace=`(7.19.3;477)
`keep-together=`(7.20.3;477)
`language=`(7.10.2;478)

Copyright © Crane Softwrights Ltd.

last-line-end-indent=(7.16.3;478)
leader-alignment=(7.22.1;478)
leader-length=(7.22.4;478)
leader-pattern=(7.22.2;478)
leader-pattern-width=(7.22.3;479)
letter-spacing=(7.17.2;479)
line-height=(7.16.4;479)
line-height-shift-adjustment=(7.16.5;480)
line-stacking-strategy=(7.16.6;480)
linefeed-treatment=(7.16.7;480)
merge-pages-across-index-key-references=(7.24.6;482)
merge-ranges-across-index-key-references=(7.24.4;483)
merge-sequential-page-numbers=(7.24.5;483)
orphans=(7.20.6;484)
page-break-inside=(7.31.18;487)
page-number-treatment=(7.24.3;488)
pitch=(7.7.7;489)
pitch-range=(7.7.8;489)
provisional-distance-between-starts=(7.30.12;490)
provisional-label-separation=(7.30.11;491)
relative-align=(7.14.6;492)
richness=(7.7.10;493)
rule-style=(7.22.5;494)
rule-thickness=(7.22.6;494)
scale-option=(7.30.14;494)
score-spaces=(7.30.15;495)
script=(7.10.3;495)
speak=(7.7.11;497)
speak-header=(7.7.12;497)
speak-numeral=(7.7.13;497)
speak-punctuation=(7.7.14;497)
speech-rate=(7.7.15;498)
start-indent=(7.11.7;498)
stress=(7.7.16;499)
text-align=(7.16.9;500)
text-align-last=(7.16.10;500)
text-indent=(7.16.11;501)
text-transform=(7.17.6;501)
visibility=(7.30.17;502)
voice-family=(7.7.17;503)
volume=(7.7.18;503)
white-space=(7.31.23;503)
white-space-collapse=(7.16.12;503)
white-space-treatment=(7.16.8;504)
widows=(7.20.7;504)
word-spacing=(7.17.8;504)

 Copyright © Crane Softwrights Ltd.

`wrap-option=`(7.16.13;504)
`writing-mode=`(7.29.7;504)
`xml:lang=`(7.31.24;505)

Copyright © Crane Softwrights Ltd.

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;450)
`background-position=`(7.31.2;451)
`border=`(7.31.3;453)
`border-bottom=`(7.31.4;456)
`border-color=`(7.31.5;457)
`border-left=`(7.31.6;458)
`border-right=`(7.31.7;459)
`border-spacing=`(7.31.9;460)
`border-style=`(7.31.8;461)
`border-top=`(7.31.10;462)
`border-width=`(7.31.11;463)
`cue=`(7.31.12;467)
`font=`(7.31.13;470)
`margin=`(7.31.14;480)
`padding=`(7.31.15;484)
`page-break-after=`(7.31.16;487)
`page-break-before=`(7.31.17;487)
`page-break-inside=`(7.31.18;487)
`pause=`(7.31.19;488)
`position=`(7.31.20;490)
`size=`(7.31.21;495)
`vertical-align=`(7.31.22;502)
`white-space=`(7.31.23;503)
`xml:lang=`(7.31.24;505)

 Copyright © Crane Softwrights Ltd.

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

Copyright © Crane Softwrights Ltd.

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
- Media: visual
- 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=`

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

- Value: link | visited | active | hover | focus
- Media: interactive
- 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>`

`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
- Percentages: see prose
- Media: visual
- Initial: auto

 Copyright © Crane Softwrights Ltd.

- 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 Line areas (page 100) 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
- Media: visual
- 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=`

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

- Value: [any | <percentage>]* | inherit
- Percentages: intrinsic height
- Initial: any
- Media: visual
- 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>`

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

- Value: [any | <percentage>]* | inherit
- Percentages: intrinsic width
- Initial: any
- Media: visual

Copyright © Crane Softwrights Ltd.

- an unordered white-space-separated list of percentages and possibly the token "any" to constrain the possibly 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
- Media: interactive
- 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
- Media: aural
- 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>, <bidi-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
- Percentages: allowed on 'background-position'
- Media: visual
- 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
- Media: visual
- Initial: scroll

 Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- Initial: transparent
- 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
- Media: visual
- Initial: none
- 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
- Percentages: refer to the size of the box itself
- Initial: 0% 0%
- Media: visual

Copyright © Crane Softwrights Ltd.

- 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%
- Percentages: refer to the size of the padding-rectangle
- 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: `background-position=`

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

- Value: <percentage> | <length> | top | center | bottom | inherit
- Initial: 0%
- Percentages: refer to the size of the padding-rectangle
- 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: `background-position=`

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

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

 Copyright © Crane Softwrights Ltd.

- "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
- Percentages: refers to the "line-height" of the parent area
- Media: visual
- 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=`

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

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

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

- Value: auto | <length> | <percentage> | <length-range> | inherit
- Initial: auto
- Percentages: see prose
- Media: visual
- 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>`

`border=` (7.31.3; CSS; shorthand):

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

Copyright © Crane Softwrights Ltd.

`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

 Copyright © Crane Softwrights Ltd.

- 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
- 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-before-precedence=` (7.28.2; 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
- 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
- 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-before-width=` (7.8.9; CSS; basic):

- Media: visual

Copyright © Crane Softwrights Ltd.

- 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

 Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- Initial: collapse
- object to which this property applies: `<table>`

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

- Value: [<color> | transparent]{1,4} | inherit
- Media: visual
- Initial: see individual properties
- up to four individual values for the border color can be specified in this shorthand
  - see Borders (page 335) 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

`border-end-color=` (7.8.16; 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-end-precedence=` (7.28.4; 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
- 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>`

Copyright © Crane Softwrights Ltd.

`border-end-style=` (7.8.17; 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-end-width=` (7.8.18; 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-left=` (7.31.6; 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-left-color=` (7.8.25; 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-left=`

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

- Media: visual

 Copyright © Crane Softwrights Ltd.

- 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>`
- 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

Copyright © Crane Softwrights Ltd.

- 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):

- Value: <length> <length>? | inherit
- Media: visual
- Initial: 0pt
- 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
- Media: visual
- Initial: the value of the 'color' property

 Copyright © Crane Softwrights Ltd.

- 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-start-precedence= (7.28.6; 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
- 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-start-style= (7.8.14; 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-start-width= (7.8.15; 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-style= (7.31.8; CSS; shorthand):

Copyright © Crane Softwrights Ltd.

- 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 Borders (page 335) 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
- Media: visual
- Initial: medium

Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- Initial: see individual properties
- up to four individual values for the border width can be specified in this shorthand
  - see Borders (page 335) 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
- Percentages: refer to height of containing block
- Initial: auto
- 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
- Media: visual
- Initial: auto
- 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 | inherit
- Media: visual
- 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=

Copyright © Crane Softwrights Ltd.

`caption-side=` (7.28.7; CSS; complete):
- Value: before | after | start | end | top | bottom | left | right | inherit
- Media: visual
- 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>`

`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>`

`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>`

`change-bar-offset=` (7.30.3; XSL; extended):
- Value: <length>
- Media: visual
- 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>`

`change-bar-placement=` (7.30.4; XSL; extended):
- Value: start | end | left | right | inside | outside | alternate
- Media: visual
- Initial: start
- positions the change bar relative to the column areas
- object to which this property applies: `<change-bar-begin>`

`change-bar-style=` (7.30.5; XSL; extended):
- Value: `border-style=`
- Media: visual
- Initial: none

 Copyright © Crane Softwrights Ltd.

- the style to use for the change bar, using the definitions as for borders (see Borders (page 335))
- object to which this property applies: `<change-bar-begin>`

`change-bar-width=` (7.30.6; XSL; extended):

- Value: `border-width=`
- Media: visual
- Initial: medium
- the width to use for the change bar, using the definitions as for borders (see Borders (page 335))
- object to which this property applies: `<change-bar-begin>`

`character=` (7.17.1; XSL; basic):

- Value: <character>
- Media: visual
- 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
- Media: visual
- 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
- Media: visual
- 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
- Media: visual
- Initial: depends on user agent
- see Function groupings (page 403) for the color functions available
- objects to which this property applies: `<bidi-override>`, `<block>`, `<bookmark-title>`, `<character>`, `<initial-property-set>`, `<inline>`, `<leader>`, `<title>`

`color-profile-name=` (7.18.2; XSL; extended):

Copyright © Crane Softwrights Ltd.

- Value: <name> | inherit
- Initial: N/A, value is required
- Media: visual
- object to which this property applies: `<color-profile>`

`column-count=` (7.27.2; XSL; extended):
- Value: <number> | inherit
- Initial: 1
- Media: visual
- object to which this property applies: `<region-body>`

`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
- Media: visual
- 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>`

`column-width=` (7.28.9; XSL; basic):
- Value: <length> | <percentage>
- Initial: see prose
- Percentages: refer to width of table
- Media: visual
- see Table-related formatting objects (page 181) 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
- Percentages: intrinsic height
- Media: visual
- objects to which this property applies: `<external-graphic>`, `<instream-foreign-object>`

`content-type=` (7.30.7; XSL; extended):
- Value: <string> | auto
- Initial: auto
- Media: visual
- 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>`

 Copyright © Crane Softwrights Ltd.

`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
- Percentages: intrinsic width
- Media: visual
- objects to which this property applies: `<external-graphic>`, `<instream-foreign-object>`

`country=` (7.10.1; XSL; extended):
- Value: none | <country> | inherit
- Initial: none
- Media: visual
- 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=`

`cue=` (7.31.12; CSS; shorthand):
- Value: `cue-before=` || `cue-after=` | inherit
- Initial: not defined for shorthand properties
- Media: aural
- when a single value is specified it is applied to both properties

`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>`, `<bidi-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>`, `<bidi-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=`

Copyright © Crane Softwrights Ltd.

`destination-placement-offset=` (7.23.5; XSL; extended):
- Value: <length>
- Media: interactive
- Initial: 0pt
- 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
- Media: visual
- Initial: ltr
- this property is deprecated for all formatting objects other than `<bidi-override>`
  - if used for such other objects, this overrides the direction implied by any `writing-mode=` property
- object to which this property applies: `<bidi-override>`

`display-align=` (7.14.4; XSL; extended):
- Value: auto | before | center | after | inherit
- Media: visual
- Initial: auto
- 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
- Media: visual
- 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=`

`elevation=` (7.7.4; CSS; basic):
- Value: <angle> | below | level | above | higher | lower | inherit
- Media: aural
- Initial: level

Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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>`

`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-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):

Copyright © Crane Softwrights Ltd.

- Value: empty string | <uri-specification>
- Media: interactive
- 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>`

`float=` (7.19.2; CSS; extended):

- Value: before | start | end | left | right | inside | outside | none | inherit
- Media: visual
- Initial: none
- 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>
- Media: all
- 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>`

`flow-map-reference=` (7.27.19; XSL; extended):

- Value: <name>
- Media: all
- 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>
- Media: visual
- 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>`

`flow-name-reference=` (7.27.20; XSL; extended):

- Value: <name>
- Media: all
- 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
- Media: visual
- Initial: see individual properties

Copyright © Crane Softwrights Ltd.

- 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*],]* [<family-name> | *generic-family*] | inherit
- Media: visual
- 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: `<bidi-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: `<bidi-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: `<bidi-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

Copyright © Crane Softwrights Ltd.

- 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: `<bidi-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
- Media: visual
- 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: `<bidi-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
- Media: visual
- Initial: normal
- note that "`italic`" will match "`oblique`" if no italic face is available in the font family
- objects to which this property applies: `<bidi-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
- Media: visual
- Initial: normal
- objects to which this property applies: `<bidi-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
- Media: visual
- Initial: normal

 Copyright © Crane Softwrights Ltd.

- "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: <bidi-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 | no-force | inherit
- Media: visual
- 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 442) 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-write 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

Copyright © Crane Softwrights Ltd.

- see Format tokens from XSLT 1.0 subset (page 442) 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 442) 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

 Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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>`, `<bidi-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

Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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
- Media: interactive
- 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>`

`initial-page-number=` (7.27.7; XSL; basic):

- Value: auto | auto-odd | auto-even | <number> | inherit
- Media: visual
- Initial: auto
- object to which this property applies: `<page-sequence>`

`inline-progression-dimension=` (7.15.7; CSS; basic):

- Value: auto | <length> | <percentage> | <length-range> | inherit
- Percentages: see prose
- Media: visual
- 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>`

`internal-destination=` (7.23.8; XSL; extended):

- Value: empty string | <idref>
- Media: interactive
- 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>`

 Copyright © Crane Softwrights Ltd.

`intrinsic-scale-value=` (7.30.9; XSL; extended):
- Value: <percentage> | inherit
- Initial: 100%
- Percentages: user defined
- Media: visual
- 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>`

`intrusion-displace=` (7.19.3; XSL; extended):
- Value: auto | none | line | indent | block | inherit
- Initial: auto
- Media: visual
- 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>`

`keep-together=` (7.20.3; XSL; extended):
- Value: <keep> | inherit
- Initial: .within-line=auto, .within-column=auto, .within-page=auto
- Media: visual
- 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=`

`keep-with-next=` (7.20.4; XSL; basic):
- Value: <keep> | inherit
- Initial: .within-line=auto, .within-column=auto, .within-page=auto
- Media: visual
- 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=`

`keep-with-previous=` (7.20.5; XSL; basic):
- Media: visual

Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- 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=`

`last-line-end-indent=` (7.16.3; XSL; extended):

- Value: <length> | <percentage> | inherit
- Initial: 0pt
- 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
- Media: visual
- 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>`

`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

 Copyright © Crane Softwrights Ltd.

- "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: <bidi-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 442) 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

Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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 Line areas (page 99) for more details on half-leading
- see Superscript and subscript (page 102) 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 Line areas (page 98) 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
- Initial: not defined for shorthand properties
- Percentages: refer to width of containing block
- 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

 Copyright © Crane Softwrights Ltd.

`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

Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- objects to which this property applies: `<conditional-page-master-reference>`, `<page-sequence>`, `<repeatable-page-master-reference>`, `<single-page-master-reference>`

`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>`

`merge-pages-across-index-key-references=` (7.24.6; XSL; extended):
- Value: merge | leave-separate
- Initial: merge
- Media: all

 Copyright © Crane Softwrights Ltd.

- 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>`

`merge-ranges-across-index-key-references=` (7.24.4; XSL; extended):

- Value: merge | leave-separate
- Initial: merge
- Media: all
- 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>`

`merge-sequential-page-numbers=` (7.24.5; XSL; extended):

- Value: merge | leave-separate
- Initial: merge
- Media: all
- 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
- Initial: 0pt
- 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

`min-width=` (7.15.11; CSS; complete):

- Value: <length> | <percentage> | inherit
- Initial: depends on UA
- Percentages: refer to width 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

`number-columns-repeated=` (7.28.12; XSL; basic):

- Value: <number>
- Initial: 1
- Media: visual
- 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>
- Initial: 1
- Media: visual
- 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):

Copyright © Crane Softwrights Ltd.

- Value: <number>
- Initial: 1
- Media: visual
- object to which this property applies: `<table-cell>`

`odd-or-even=` (7.27.12; XSL; extended):

- Value: odd | even | any | inherit
- Initial: any
- Media: visual
- object to which this property applies: `<conditional-page-master-reference>`

`orphans=` (7.20.6; CSS; basic):

- Value: <integer> | inherit
- Initial: 2
- Media: visual
- 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>`

`overflow=` (7.21.2; CSS; basic):

- Value: visible | hidden | scroll | error-if-overflow | repeat | auto | inherit
- Initial: auto
- Media: visual
- "`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>`

`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: 0pt
- Percentages: refer to width of containing block
- Media: visual

 Copyright © Crane Softwrights Ltd.

- *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: 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>`

`padding-bottom=` (7.8.36; 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=`

`padding-end=` (7.8.34; CSS; basic):

- Value: *padding-width* | <length-conditional> | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual

Copyright © Crane Softwrights Ltd.

- *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: 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=`

`padding-right=` (7.8.38; 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=`

`padding-start=` (7.8.33; CSS; basic):

- Value: *padding-width* | <length-conditional> | inherit
- Initial: 0pt
- Percentages: refer to width of containing block
- Media: visual

 Copyright © Crane Softwrights Ltd.

- *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

Copyright © Crane Softwrights Ltd.

- "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
- Initial: depends on user agent
- Percentages: see descriptions of 'pause-before' and 'pause-after'
- Media: aural
- a shorthand specifying, in order, pause-before= and pause-after=
  - specifying one value applies to both properties

 Copyright © Crane Softwrights Ltd.

`pause-after=` (7.7.5; CSS; basic):

- Value: <time> | <percentage> | inherit
- Initial: depends on user agent
- Percentages: see prose
- Media: aural
- specifies the pause to be observed after speaking an object's content
- objects to which this property applies: `<basic-link>`, `<bidi-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
- Percentages: see prose
- Media: aural
- specifies the pause to be observed after speaking an object's content
- objects to which this property applies: `<basic-link>`, `<bidi-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
- Media: aural
- specifies the average pitch (a frequency) of the speaking voice
- objects to which this property applies: `<basic-link>`, `<bidi-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

Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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
- Media: aural
- 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>`, `<bidi-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
- Media: visual
- Initial: static
-

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
- Media: visual
- Initial: false
- 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):

 Copyright © Crane Softwrights Ltd.

- 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 List constructs (page 137) 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 List constructs (page 137) 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
- Media: all
- objects to which this property applies: `<index-range-end>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`

`ref-index-key=` (7.24.7; XSL; extended):

- Value: <string>
- Initial: none, value required
- Media: all
- used as the search string of index keys when collecting pages for an index entry
- object to which this property applies: `<index-key-reference>`

`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>`

`region-name-reference=` (7.27.21; XSL; extended):

- Value: <name>
- Initial: none, a value is required
- Media: all

Copyright © Crane Softwrights Ltd.

- 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>`, `<bidi-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

 Copyright © Crane Softwrights Ltd.

- 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>`

`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>`, `<bidi-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

Copyright © Crane Softwrights Ltd.

- 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>`

`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

 Copyright © Crane Softwrights Ltd.

- 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-graphic>`, `<instream-foreign-object>`

`score-spaces=` (7.30.15; XSL; extended):
- Value: true | false | inherit
- Media: visual
- Initial: true
- specifies whether the text-decoration property shall be applied to spaces
- objects to which this property applies: `<bidi-override>`, `<character>`, `<initial-property-set>`, `<page-number>`, `<page-number-citation>`, `<page-number-citation-last>`, `<scaling-value-citation>`

`script=` (7.10.3; XSL; extended):
- Value: none | auto | <script> | inherit
- Media: visual
- 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>`

`show-destination=` (7.23.9; XSL; extended):
- Value: replace | new
- Media: interactive
- 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>`

`size=` (7.31.21; CSS; shorthand):
- Value: <length>{1,2} | auto | landscape | portrait | inherit
- Media: visual
- Initial: auto
- a shorthand specifying, in order, `page-width=` and `page-height=`
  - specifying one value applies to both properties

`source-document=` (7.5.1; XSL; basic):
- Value: <uri-specification> [<uri-specification>]* | none | inherit
- Media: all
- Initial: none

Copyright © Crane Softwrights Ltd.

- 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
- Percentages: N/A (Differs from margin-bottom in CSS)
- Media: visual
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<region-body>`, `<simple-page-master>`, `<table>`, `<table-and-caption>`

`space-before=` (7.11.5; XSL; basic):

- Value: <space> | inherit
- Initial: space.minimum=0pt, .optimum=0pt, .maximum=0pt, .conditionality=discard, .precedence=0
- Percentages: N/A (Differs from margin-top in CSS)
- Media: visual
- objects to which this property applies: `<block>`, `<block-container>`, `<list-block>`, `<list-item>`, `<region-body>`, `<simple-page-master>`, `<table>`, `<table-and-caption>`

`space-end=` (7.12.5; XSL; basic):

- Value: <space> | <percentage> | inherit
- Initial: space.minimum=0pt, .optimum=0pt, .maximum=0pt, .conditionality=discard, .precedence=0
- 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>`

`space-start=` (7.12.6; XSL; basic):

- Value: <space> | <percentage> | inherit
- Initial: space.minimum=0pt, .optimum=0pt, .maximum=0pt, .conditionality=discard, .precedence=0
- Percentages: refer to inline-progression-dimension of closest ancestor block-area that is not a line-area
- Media: visual

 Copyright © Crane Softwrights Ltd.

- 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
- Media: visual
- Initial: none
- objects to which this property applies: `<block>`, `<block-container>`

`speak=` (7.7.11; CSS; basic):

- Value: normal | none | spell-out | inherit
- Media: aural
- 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>`, `<bidi-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
- Media: aural
- 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>`, `<bidi-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
- Media: aural
- Initial: continuous
- specifies how numerals are spoken
- objects to which this property applies: `<basic-link>`, `<bidi-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>`

Copyright © Crane Softwrights Ltd.

`speak-punctuation=` (7.7.14; CSS; basic):
- Value: code | none | inherit
- Initial: none
- Media: aural
- specifies how punctuation is spoken
- note that "`borge`" is not an available option
- objects to which this property applies: `<basic-link>`, `<bidi-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>`

`speech-rate=` (7.7.15; CSS; basic):
- Value: <number> | x-slow | slow | medium | fast | x-fast | faster | slower | inherit
- Initial: medium
- Media: aural
- <number> specifies the speaking rate in words per minute (varying somewhat by language)
- objects to which this property applies: `<basic-link>`, `<bidi-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>`

`src=` (7.30.16; XSL; basic):
- Value: <uri-specification> | inherit
- Initial: none, value required
- Media: visual
- objects to which this property applies: `<color-profile>`, `<external-graphic>`

`start-indent=` (7.11.7; 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>`

`starting-state=` (7.23.10; XSL; extended):
- Value: show | hide
- Initial: show
- Media: interactive
- specifies the `<multi-case>` that can be initially displayed
- objects to which this property applies: `<bookmark>`, `<multi-case>`

`starts-row=` (7.28.15; XSL; extended):

 Copyright © Crane Softwrights Ltd.

- Value: true | false
- Media: visual
- Initial: false
- object to which this property applies: `<table-cell>`

`stress=` (7.7.16; CSS; basic):

- Value: <number> | inherit
- Media: aural
- Initial: 50
- specifies the height of "local peaks" in the intonation contour of a voice
- objects to which this property applies: `<basic-link>`, `<bidi-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>`

`suppress-at-line-break=` (7.17.3; XSL; extended):

- Value: auto | suppress | retain | inherit
- Media: visual
- 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>`

`switch-to=` (7.23.11; XSL; extended):

- Value: xsl-preceding | xsl-following | xsl-any | <name>[<name>]*
- Media: interactive
- Initial: xsl-any
- specifies which `<multi-case>` object this `<multi-toggle>` will switch to when evoked
- object to which this property applies: `<multi-toggle>`

`table-layout=` (7.28.16; CSS; extended):

- Value: auto | fixed | inherit
- Media: visual
- Initial: auto
- object to which this property applies: `<table>`

`table-omit-footer-at-break=` (7.28.17; XSL; extended):

- Value: true | false
- Media: visual
- 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>`

`table-omit-header-at-break=` (7.28.18; XSL; extended):

- Value: true | false
- Media: visual
- 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>`

Copyright © Crane Softwrights Ltd.

`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-stylesheet | <uri-specification>
- Media: interactive
- Initial: use-normal-stylesheet
- 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>`

 Copyright © Crane Softwrights Ltd.

`text-decoration`= (7.17.4; CSS; extended):
- Value: none | [[underline | no-underline] || [over-line | no-overline] || [line-through | no-line-through] || [blink | no-blink]] | inherit
- Initial: none
- Media: visual
- 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>`

`text-depth`= (7.29.5; XSL; extended):
- Value: use-font-metrics | <length> | <percentage> | inherit
- Initial: use-font-metrics
- Percentages: refer to font's em-height
- Media: visual
- 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>`

`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

Copyright © Crane Softwrights Ltd.

- 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=`	`dominant-baseline=`
"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"

 Copyright © Crane Softwrights Ltd.

`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>`, `<bidi-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>`, `<bidi-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-surrounding-linefeed""wrap"`
  `"pre" "preserve" "false" "preserve" "no-wrap"`
  `"nowrap" "treat-as-space""true" "ignore-if-surrounding-linefeed""no-wrap"`

`white-space-collapse`= (7.16.12; XSL; extended):

Copyright © Crane Softwrights Ltd.

- Value: false | true | inherit
- Media: visual
- 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
- Media: visual
- 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
- Media: visual
- 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
- Percentages: refer to width of containing block
- Initial: auto
- Media: visual
- 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
- Media: visual
- Initial: normal
- specifies inter-word spacing behavior in addition to the default between words
- negative values are allowed
- objects to which this property applies: `<bidi-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
- Media: visual
- 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>`

 Copyright © Crane Softwrights Ltd.

`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>`

Copyright © Crane Softwrights Ltd.

Annex E - Sample tool information

- Introduction - Sample questions for vendors

 Copyright © Crane Softwrights Ltd.

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)?

Copyright © Crane Softwrights Ltd.

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?

 Copyright © Crane Softwrights Ltd.

Sample questions for vendors (cont.)

- 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?

Copyright © Crane Softwrights Ltd.

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`

Copyright © Crane Softwrights Ltd.

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1
Copyright © Crane Softwrights Ltd.

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`

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

 Copyright © Crane Softwrights Ltd.

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

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

Seventh Edition - 2008-01-27 - ISBN 978-1-894049-19-1

Copyright © Crane Softwrights Ltd.

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)

 Copyright © Crane Softwrights Ltd.

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 [4-1-1-1] Geometric rendered areas (083) (084)
085 [4-1-2-1] Directions and writing modes (086) (087)
088 [4-1-3-1] Margins, spaces and positioning
089 [4-1-4-1] Length values for widths and spacing
090 [4-1-5-1] Area types
091 [4-1-6-1] Stacking area and rectangle relationships (092) (093) (094) (095)
096 [4-1-7-1] Allocation rectangles and alignment (097)
098 [4-1-8-1] Line areas (099) (100) (101)
102 [4-1-9-1] Superscript and subscript
103 [4-1-10-1] References in the area tree
104 [4-2-1-1] Simple block and inline objects (105) (106) (107)
108 [4-2-2-1] <wrapper> Object
109 [4-2-3-1] <block> Object
110 [4-2-4-1] Preserving white space
111 [4-2-5-1] <initial-property-set> Object
112 [4-2-6-1] <inline> Object
113 [4-2-7-1] <page-number-citation> Object
114 [4-2-8-1] <page-number-citation-last> Object
115 [4-3-1-1] Simple page layout definition (116)
117 [4-3-2-1] Spans and columns in simple page geometry (118)
119 [4-3-3-1] <simple-page-master> Object (120)
121 [4-3-4-1] <region-body> Object (122)
123 [4-3-5-1] Page sequence titling
124 [4-3-6-1] <title> Object (125)
126 [5] Generic body constructs
127 [Introduction 5-1-1] Often-used formatting constructs (128) (129) (130) (131)
132 [5-1-1-1] Aligned pairs of block-level constructs
133 [5-1-2-1] List constructs (134) (135) (136) (137)
138 [5-1-3-1] Nested list constructs (139)
140 [5-1-4-1] <list-block> Object (141)
142 [5-1-5-1] <list-item> Object (143)
144 [5-1-6-1] <list-item-label> Object (145)
146 [5-1-7-1] <list-item-body> Object (147)

Copyright © Crane Softwrights Ltd.

148 [5-1-8-1] When is a list not a list?
149 [5-2-1-1] Non-textual information (150) (151)
152 [5-2-2-1] `<external-graphic>` Object (153)
154 [5-2-3-1] `<instream-foreign-object>` Object (155) (156)
157 [5-2-4-1] Exposing the image size scaling factor
158 [5-2-5-1] `<scaling-value-citation>` Object
159 [5-3-1-1] Link requirements
160 [5-3-2-1] `<basic-link>` Object (161)
162 [5-4-1-1] Elastic and inelastic inline areas (163) (164) (165)
166 [5-4-2-1] Multiple leaders on a single line
167 [5-4-3-1] Controlling the distance between leaders
168 [5-4-4-1] `<leader>` Object (169) (170)
171 [6] Tables
172 [Introduction 6-1-1] Tabular presentation of information (173) (174) (175) (176)
177 [6-1-1-1] Aligned tuples of block-level constructs
178 [6-1-2-1] Table-related formatting objects (179) (180) (181)
182 [6-2-1-1] Table and cell borders
183 [6-2-2-1] Spanning cells
184 [6-2-3-1] Table and cell alignment
185 [6-2-4-1] `<table-and-caption>` Object (186) (187)
188 [6-2-5-1] `<table-caption>` Object (189)
190 [6-2-6-1] `<table>` Object (191)
192 [6-2-7-1] `<table-column>` Object (193)
194 [6-2-8-1] `<table-header>` Object (195)
196 [6-2-9-1] `<table-footer>` Object
197 [6-2-10-1] `<table-body>` Object (198)
199 [6-2-11-1] `<table-row>` Object (200)
201 [6-2-12-1] `<table-cell>` Object (202) (203)
204 [7] Floats, footnotes and containers
205 [Introduction 7-1-1] Out-of-line publishing constructs (206) (207) (208)
209 [7-1-1-1] Conditional reference areas and sub-regions (210)
211 [7-2-1-1] Float definition
212 [7-2-2-1] `<float>` Object
213 [7-2-3-1] The interaction of blocks and floats (214)
215 [7-3-1-1] Footnote definition (216)
217 [7-3-2-1] `<footnote>` Object (218) (219)
220 [7-3-3-1] `<footnote-body>` Object (221)
222 [7-4-1-1] Containers (223)
224 [7-4-2-1] Block containers
225 [7-4-3-1] `<block-container>` Object
226 [7-4-4-1] `<inline-container>` Object (227)
228 [8] Flows, static content and page geometry sequencing
229 [Introduction 8-1-1] Extended page model for pagination (230) (231) (232) (233) (234)
235 [8-1-1-1] Region dimensions
236 [8-1-2-1] `<region-before>` Object
237 [8-1-3-1] `<region-after>` Object (238)

 Copyright © Crane Softwrights Ltd.

239 [8-1-4-1] <region-start> Object
240 [8-1-5-1] <region-end> Object
241 [8-2-1-1] Flowed content vs. static content (242)
243 [8-2-2-1] Flow maps
244 [8-2-3-1] <flow-map> Object
245 [8-2-4-1] <flow-assignment> Object
246 [8-2-5-1] <flow-source-list> Object
247 [8-2-6-1] <flow-name-specifier> Object
248 [8-2-7-1] <flow-target-list> Object
249 [8-2-8-1] <region-name-specifier> Object
250 [8-2-9-1] Headers and footers
251 [8-2-10-1] <static-content> Object (252)
253 [8-2-11-1] <page-number> Object
254 [8-2-12-1] <folio-prefix> Object (255)
256 [8-2-13-1] <folio-suffix> Object
257 [8-2-14-1] Dynamic content in static content (258) (259) (260) (261)
262 [8-2-15-1] <marker> Object (263)
264 [8-2-16-1] <retrieve-marker> Object (265)
266 [8-2-17-1] <retrieve-table-marker> Object
267 [8-2-18-1] Planning a simple page sequence specification
268 [8-3-1-1] Changing the page geometry based on authored content (269) (270) (271) (272)
273 [8-4-1-1] Changing the page geometry based on a pattern of geometries (274)
275 [8-4-2-1] Overview of page sequence specifications
276 [8-4-3-1] Page geometry sub-sequences (277)
278 [8-4-4-1] Forced blank pages (279)
280 [8-4-5-1] <page-sequence-master> Object (281) (282)
283 [8-4-6-1] <single-page-master-reference> Object
284 [8-4-7-1] <repeatable-page-master-reference> Object
285 [8-4-8-1] <repeatable-page-master-alternatives> Object (286)
287 [8-4-9-1] <conditional-page-master-reference> Object (288)
289 [8-4-10-1] Planning a more complex page sequence specification
290 [9] Bookmarks and indexes
291 [Introduction 9-1-1] Additional navigation aids (292)
293 [9-1-1-1] Bookmarks (294)
295 [9-1-2-1] <bookmark-tree> Object (296)
297 [9-1-3-1] <bookmark> Object
298 [9-1-4-1] <bookmark-title> Object
299 [9-2-1-1] Resolving indexing page numbers
300 [9-2-2-1] Preparing the content for indexing (301)
302 [9-2-3-1] <index-range-begin> Object (303)
304 [9-2-4-1] <index-range-end> Object
305 [9-2-5-1] Assembling indexing citations
306 [9-2-6-1] <index-page-citation-list> Object (307)
308 [9-2-7-1] <index-page-citation-list-separator> Object
309 [9-2-8-1] <index-page-citation-range-separator> Object
310 [9-2-9-1] <index-key-reference> Object

Copyright © Crane Softwrights Ltd.

311 [9-2-10-1] `<index-page-number-prefix>` Object
312 [9-2-11-1] `<index-page-number-suffix>` Object
313 [10] Breaks, keeps, spacing, borders and backgrounds
314 [Introduction 10-1-1] Spacing and arrangement constraint definition (315)
316 [10-1-1-1] Breaks (317)
318 [10-2-1-1] Widows and orphans (319)
320 [10-3-1-1] Keeps
321 [10-3-2-1] Examples of keeps (322) (323) (324)
325 [10-3-3-1] Keep strength (326) (327)
328 [10-4-1-1] Spacing, conditionality and precedence (329) (330)
331 [10-4-2-1] Character spacing
332 [10-5-1-1] Borders (333)
334 [10-6-1-1] Backgrounds (335)
336 [10-6-2-1] Decorating page columns using a background
337 [11] Supplemental objects
338 [Introduction 11-1-1] Specialty constructs (339)
340 [11-1-1-1] Change bars
341 [11-1-2-1] `<change-bar-begin>` Object
342 [11-1-3-1] `<change-bar-end>` Object
343 [11-1-4-1] Color system specification
344 [11-1-5-1] `<color-profile>` Object
345 [11-1-6-1] `<declarations>` Object
346 [11-1-7-1] Characters and bi-directional text
347 [11-1-8-1] `<character>` Object
348 [11-2-1-1] The importance of bidirectional text
349 [11-2-2-1] The mechanics of mixing text of different writing directions (350)
351 [11-2-3-1] Illustration of bidirectional embedding (352) (353)
354 [11-2-4-1] The bidirectional support challenge (355) (356) (357)
358 [11-2-5-1] `<bidi-override>` Object
359 [12] Interactive objects
360 [Introduction 12-1-1] Dynamic and interactive object rendering (361)
362 [12-1-1-1] Dynamically changing property values (363)
364 [12-1-2-1] `<multi-properties>` Object (365)
366 [12-1-3-1] `<multi-property-set>` Object (367)
368 [12-2-1-1] Dynamically changing formatting object sub-trees (369)
370 [12-2-2-1] `<multi-switch>` Object (371) (372) (373) (374)
375 [12-2-3-1] `<multi-case>` Object (376)
377 [12-2-4-1] `<multi-toggle>` Object (378)
379 [13] Where XSL-FO 1 falls short
380 [Introduction 13-1-1] What is missing in XSL-FO 1?
381 [13-1-1-1] Arbitrating between retrieved content (382)
383 [13-1-2-1] Retrieving markers into the body
384 [13-1-3-1] Line numbering
385 [13-1-4-1] Footnotes
386 [13-1-5-1] Determining the current formatted location
387 [13-2-1-1] Choosing page geometry based on page content

 Copyright © Crane Softwrights Ltd.

388 [13-3-1-1] Aligning flows and marks
389 [A] Using XSLT with XSL-FO
390 [Introduction A-1-1] Using XSLT with XSL-FO
391 [A-1-1-1] `<xsl:attribute-set>` instruction
392 [A-1-2-1] Simpler list and footnote structures in XSL-FO
393 [A-2-1-1] Common errors writing expressions (394)
395 [B] XSL-FO expressions
396 [Introduction B-1-1] Expressions in XSL-FO
397 [B-1-1-1] Production summary
398 [B-2-1-1] Function summary
399 [B-2-2-1] Function groupings
400 [B-2-3-1] Functions summarized by name
401 [C] XSL-FO object summary
402 [Introduction C-1-1] Objects
403 [C-1-1-1] Objects summarized by name
404 [C-1-2-1] Prototypical hierarchy
405 [C-1-3-1] Objects summarized by type
406 [D] XSL-FO property summaries
407 [Introduction D-1-1] Properties
408 [D-1-1-1] Common Absolute Position Properties
409 [D-1-2-1] Common Accessibility Properties
410 [D-1-3-1] Common Aural Properties
411 [D-1-4-1] Common Border, Padding, and Background Properties
412 [D-1-5-1] Common Font Properties
413 [D-1-6-1] Common Hyphenation Properties
414 [D-1-7-1] Common Margin Properties-Block
415 [D-1-8-1] Common Margin Properties-Inline
416 [D-1-9-1] Common Relative Position Properties
417 [D-2-1-1] Property data types
418 [D-2-2-1] Recommendation bibliography
419 [D-2-3-1] Format tokens from XSLT 1.0 subset
420 [D-3-1-1] Inherited properties
421 [D-3-2-1] Shorthand properties
422 [D-3-3-1] Property summary
423 [E] Sample tool information
424 [Introduction E-1-1] Sample questions for vendors (425) (426)
427 [Conclusion -1-1] Where To Go From Here?
428 [Conclusion -2-1] Colophon
429 [Conclusion -3-1] Obtaining a copy of the comprehensive tutorial
430 [Postlude] Practical Formatting Using XSL-FO (Postlude)

Copyright © Crane Softwrights Ltd.

Practical Formatting Using XSL-FO

Index

A

A4 page dimensions 39, 135-136
`abs()` function **405**
  in function summaries 403
`absolute-position=` property **448**
  in set of common properties 425
  referenced 88, 226, 490
`active-state=` property **448**
  in object definition 369
  referenced 365
Adobe Acrobat 40
after edge 86
aggregation 103
alignment
  blocks 127, 132, 179
  cells 186
  tables 186
`alignment-adjust=` property **448**
  in object definition 112-114, 152, 155, 159, 161, 169, 228, 255, 350
  referenced 449, 502
`alignment-baseline=` property **449**
  in object definition 112-114, 152, 155, 159, 161, 169, 228, 255, 350
  referenced 101, 105, 153, 155, 224, 449, 502
allocation area/rectangle 90, 96-97
`allowed-height-scale=` property **449**
  in inherited property summary 443
  in object definition 152, 155
`allowed-width-scale=` property **449**
  in inherited property summary 443
  in object definition 152, 155
alphabetic numbering 442
<angle> data type **436**
  in property definition 450, 468, 473
  referenced 2, 491
Antenna House XSL Formatter 40, 54
appearance 48
Arabic 84, 87, 351, 354
area
  tree/model 52-53, 77-78, 82-84
  types 90, 94
aural media 26, 29, 59
authoring 9
`auto-restore=` property **450**
  in inherited property summary 443
  in object definition 373
  referenced 373
`azimuth=` property **450**
  in inherited property summary 443
  in set of common properties 427

B

`background=` property **450**
  in object definition 428
  in property definition 451, 453
  in shorthand definition 446
`background-attachment=` property **450**
  in set of common properties 428
  referenced 337
`background-color=` property **451**
  in set of common properties 428
  referenced 337, 451
`background-image=` property **451**
  in set of common properties 428
  referenced 337
`background-position=` property **451**
  in object definition 428
  in property definition 452
  in shorthand definition 446
`background-position-horizontal=` property **452**
  in set of common properties 428
  referenced 337, 452
`background-position-vertical=` property **452**
  in set of common properties 428
  referenced 337, 452
`background-repeat=` property **452**
  in set of common properties 428
  referenced 337, 451
backgrounds 337-339

 Copyright © Crane Softwrights Ltd.

`baseline-shift=` property **453**
in object definition 112-114, 152, 155, 159, 161, 169, 228, 255, 350
referenced 2, 98, 102, 106, 112, 449, 502
baselines 100, 105, 135, 163
`<basic-link>` object **161**
in object summaries 409, 420
in property definition 449-463, 467-470, 475-478, 480, 485-487, 489-490, 492-500, 503
referenced 2, 72, 131, 160, 365, 367, 418, 426-428, 433-434, 448, 508
before edge 86
before-float-reference-area 85, 117, 209, 211-213, 317
bi-directional writing 87, 349, 351-361
`<bidi-override>` object **361**
in object summaries 409, 421
in property definition 450, 465, 467-469, 471-473, 475-476, 479-480, 489-490, 492-493, 495, 497-499, 502-504
referenced 72, 87, 342, 349, 353, 356-357, 418, 427, 430, 434, 468
bilingual presentation 128, 174
blank page 60, 83, 233, 276, 280, 281-282
`blank-or-not-blank=` property **453**
in object definition 290
referenced 280-282
block area 90
`%block;` entity object **71**
in parent content model 68, 108-109, 112, 144, 146, 161, 190, 203, 214, 222, 227-228, 253, 264, 361, 378, 380, 409-413
referenced 73-74
`%inline;` entity object **72**
in parent content model 108-109, 112, 124, 161, 169, 256, 258, 264, 311-312, 314-315, 361, 378, 380, 409-411, 413
referenced 73-75
`<block>` object **109**
in object summaries 409, 420
in property definition 450-463, 465, 467, 469, 471-478, 480-481, 484-487, 489-490, 492-501, 503-504
referenced ?, 70-71, 81, 105, 139, 152, 155, 169, 319, 418, 426-428, 430-432, 434, 465
`<block-container>` object **227**
in object summaries 409, 420
in property definition 448, 451-463, 465, 468-469, 474-479, 481, 484-487, 491, 494, 496-498, 502, 504-505
referenced 71, 85, 87-88, 124, 169, 208, 210, 224, 384, 416, 425, 428, 432
`block-progression-dimension=` property **453**
in object definition 112, 152, 155, 190, 192, 201, 203, 227-228
referenced 225, 482
block-progression-direction 86, 91, 317
`body-start()` function **405**
in function summaries 403
referenced ?, 136-137
`<bookmark>` object **300**
in object summaries 409, 421
in parent content model 298, 300, 409
in property definition 470, 476, 494, 496, 498
referenced 295-296, 300-301, 416, 426
`<bookmark-title>` object **301**
in object summaries 409, 421
in parent content model 300, 409
in property definition 465, 472-473, 494, 496
referenced 295-296, 416, 426
`<bookmark-tree>` object **298**
in object summaries 409, 421
in parent content model 61, 412
referenced 295-296, 300, 416
`border=` property **453**
in object definition 428
in property definition 454-455, 457, 461
in shorthand definition 446
referenced 58
`border-after-color=` property **453**
in set of common properties 428
`border-after-precedence=` property **454**
in object definition 192, 194, 196, 198-199, 201, 203
referenced 455, 457, 461
`border-after-style=` property **454**
in set of common properties 428

Copyright © Crane Softwrights Ltd.

`border-after-width=` property **454**
in set of common properties 428
referenced ?, 93
`border-before-color=` property **455**
in set of common properties 428
`border-before-precedence=` property **455**
in object definition 192, 194, 196, 198-199, 201, 203
`border-before-style=` property **455**
in set of common properties 428
`border-before-width=` property **455**
in set of common properties 428
referenced ?, 93
`border-bottom=` property **456**
in object definition 429
in property definition 456-457
in shorthand definition 446
referenced 58
`border-bottom-color=` property **456**
in set of common properties 428
referenced 58
`border-bottom-style=` property **456**
in set of common properties 428
`border-bottom-width=` property **456**
in set of common properties 428
`border-collapse=` property **457**
in inherited property summary 443
in object definition 192
referenced 184, 454, 460, 469
`border-color=` property **457**
in object definition 429
in property definition 454-459, 461-462
in shorthand definition 446
referenced 58, 184
`border-end-color=` property **457**
in set of common properties 428
`border-end-precedence=` property **457**
in object definition 192, 194, 196, 198-199, 201, 203
`border-end-style=` property **457**
in set of common properties 428
`border-end-width=` property **458**
in set of common properties 428
referenced ?
`border-left=` property **458**
in object definition 429
in property definition 458-459
in shorthand definition 446
`border-left-color=` property **458**
in set of common properties 428
`border-left-style=` property **458**
in set of common properties 428
`border-left-width=` property **459**
in set of common properties 428
border-rectangle 77, 91-95, 213, 237
`border-right=` property **459**
in object definition 428
in property definition 459-460
in shorthand definition 446
`border-right-color=` property **459**
in set of common properties 428
`border-right-style=` property **459**
in set of common properties 428
`border-right-width=` property **460**
in set of common properties 428
`border-separation=` property **460**
in inherited property summary 443
in object definition 192
referenced 184, 460
`border-spacing=` property **460**
in inherited property summary 443
in object definition 192
in property definition 460
in shorthand definition 446
`border-start-color=` property **460**
in set of common properties 428
`border-start-precedence=` property **461**
in object definition 192, 194, 196, 198-199, 201, 203
`border-start-style=` property **461**
in set of common properties 428
`border-start-width=` property **461**
in set of common properties 428
referenced ?
`border-style=` property **461**
in object definition 428
in property definition 454-456, 458-462
in shorthand definition 446
referenced 58, 93, 184

 Copyright © Crane Softwrights Ltd.

`border-top=` property **462**
in object definition 428
in property definition 462-463
in shorthand definition 446
`border-top-color=` property **462**
in set of common properties 428
`border-top-style=` property **462**
in set of common properties 428
`border-top-width=` property **462**
in set of common properties 428
`border-width=` property **463**
in object definition 429
in property definition 455-461, 463
in shorthand definition 446
referenced 121, 184, 238-239, 241-242
borders 95-97, 184, 331, 335-336
precedence 184
width 93
bottom edge 86
`bottom=` property **463**
in set of common properties 425
referenced 88, 226, 502
`<br>` (HTML) 319
`break-after=` property **463**
in object definition 109, 140, 142, 187, 192, 201, 227
referenced 60, 281, 319, 487
`break-before=` property **463**
in object definition 109, 140, 142, 187, 192, 201, 227
referenced 60, 281, 319, 487
breaks 319-320, 328
browser windows 7, 13, 40

C

`caption-side=` property **463**
in inherited property summary 443
in object definition 187
referenced 180, 190
Cascading Stylesheets (CSS) 17, 19, 22, 23, 26, 45, 82, 89, 352
`case-name=` property **464**
in object definition 378
`case-title=` property **464**
in object definition 378
`ceiling()` function **405**
in function summaries 403
change pages 47
`<change-bar-begin>` object **344**
in object summaries 409, 422
in property definition 450, 464-465, 467, 469, 489-490, 493-494, 496-499, 503, 505
referenced 73, 342-343, 418, 426-427
`change-bar-class=` property **464**
in object definition 344-345
referenced 343-345
`change-bar-color=` property **464**
in inherited property summary 443
in object definition 344
`<change-bar-end>` object **345**
in object summaries 409, 422
in property definition 450, 464, 467, 469, 489-490, 493-494, 496-499, 503
referenced 73, 342-343, 418, 426-427
`change-bar-offset=` property **464**
in inherited property summary 443
in object definition 344
`change-bar-placement=` property **464**
in inherited property summary 443
in object definition 344
`change-bar-style=` property **464**
in inherited property summary 443
in object definition 344
`change-bar-width=` property **465**
in inherited property summary 443
in object definition 344
<character> data type **436**
in property definition 465, 473-474
character direction
embedding 353, 357
strength 352
`<character>` object **350**
in object summaries 409, 421
in property definition 449-463, 465, 467-469, 471-480, 485-487, 489-490, 492-493, 495-504
referenced 72, 113-114, 255, 342, 349, 418, 427-428, 430-431, 433-434
`character=` property **465**
in object definition 350
Chinese 87

Copyright © Crane Softwrights Ltd.

`clear=` property **465**
in object definition 109, 140, 187, 192, 214, 227
referenced 213, 215
`clip=` property **465**
in object definition 121, 152, 155, 227-228, 238-239, 241-242
colophon 511
color 346-347
<color> data type **436**
in property definition 451, 453-460, 462, 464-465, 501
`color=` property **465**
in inherited property summary 443
in object definition 109, 111-112, 124, 169, 301, 350, 361
referenced 55
`<color-profile>` object **347**
in object summaries 409, 420
in parent content model 348, 409
in property definition 466, 492, 498
referenced 342, 346, 416
`color-profile-name=` property **465**
in object definition 347
`column-count=` property **466**
in object definition 121
`column-gap=` property **466**
in object definition 121
`column-number=` property **466**
in object definition 194, 203
referenced 185, 466
`column-width=` property **466**
in object definition 194
referenced 181
columns (page)
balancing 117
footnotes 388
single, see normal-flow-reference-area
spanned 117-118, see also span-reference-area
width 117
columns (table)
properties 177
spanned 176-177, 185
width 174-175, 179, 181
"Common Absolute Position Properties" property set **425**
in object definition 227
"Common Accessibility Properties" property set **426**
in object definition 61, 109, 111-114, 124, 140, 142, 144, 146, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 219, 222, 255, 300-301, 344-345, 367, 373, 378, 380
"Common Aural Properties" property set **427**
in object definition 109, 111-114, 124, 140, 142, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 255, 344-345, 350, 361
"Common Border, Padding, and Background Properties" property set **428-429**
in object definition 109, 111-114, 121, 124, 140, 142, 152, 155, 159, 161, 169, 187, 190, 192, 194, 196, 198-199, 201, 203, 227-228, 238-239, 241-242, 255, 350
"Common Font Properties" property set **430**
in object definition 109, 111-114, 124, 159, 169, 255, 350, 361
"Common Hyphenation Properties" property set **431**
in object definition 109, 350
"Common Margin Properties-Block" property set **432**
in object definition 109, 119, 121, 140, 142, 187, 192, 227
"Common Margin Properties-Inline" property set **433**
in object definition 112-114, 124, 152, 155, 159, 161, 169, 228, 255, 350
"Common Relative Position Properties" property set **434**
in object definition 109, 111-114, 140, 142, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 228, 255, 350, 361
compound properties 53, 58

 Copyright © Crane Softwrights Ltd.

`<conditional-page-master-reference>` object **290**
in object summaries 409, 420
in parent content model 288, 412
in property definition 453, 482, 484, 488
referenced ?, 236, 279, 281, 288, 416
conditionality 53, 85, 89, 93, 96, 318, 331-333, 336
conformance 59
container objects 224-229
block containers 207-229, 226
inline containers 97, 105, 229
content-based formatting 45, 47
`content-height=` property **466**
in object definition 152, 155
referenced 150, 153, 155
content-rectangle 77, 91-95
`content-type=` property **466**
in object definition 152, 155
`content-width=` property **466**
in object definition 152, 155
referenced 150, 153, 155
<country> data type **436**
in property definition 467
`country=` property **467**
in inherited property summary 443
in object definition 65, 159
in set of common properties 431
referenced 505
`cue=` property **467**
in object definition 427
in property definition 467
in shorthand definition 446
`cue-after=` property **467**
in set of common properties 427
`cue-before=` property **467**
in set of common properties 427

D

data types 56, 435-438
`<declarations>` object **348**
in object summaries 409, 420
in parent content model 61, 412
referenced 51, 342, 346-347, 416
`destination-placement-offset=` property **467**
in object definition 161
device independence 26
dictionary headers 232, 259, 262
`direction=` property **468**
in inherited property summary 443
in object definition 361
referenced 352-353, 361
discarding space 88
areas, see conditionality
white-space characters, see white-space characters; collapsing
`display-align=` property **468**
in inherited property summary 443
in object definition 121, 152, 155, 203, 227-228, 238-239, 241-242
referenced 121, 153, 155, 186, 203, 238-239
Document Object Model (DOM) 24
Document Style Semantics and Specification Language (DSSSL) 17, 21, 26, 45
Document Type Definition (DTD) 27
`dominant-baseline=` property **468**
in object definition 112-114, 152, 155, 159, 161, 169, 228, 255, 350
referenced 502
dot leader 127, 164
dynamic content 80, 259-263

E

`elevation=` property **468**
in inherited property summary 443
in set of common properties 427
em (length) 89
empty page, see blank page
`empty-cells=` property **469**
in inherited property summary 443
in object definition 203
referenced 184
end edge 86
`end-indent=` property **469**
in inherited property summary 443
in set of common properties 432
referenced 92-94, 135, 137, 144, 479
endnotes 207
`ends-row=` property **469**
in object definition 203
referenced 182
ex (length) 89
Examples ZIP file 5

Copyright © Crane Softwrights Ltd.

expressions 56
errors writing 396-397
productions 400-401
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 Transformations (XSLT) 11, 17, 23, 24-25, 271, 274, 393-397
`extent=` property **469**
in object definition 238-239, 241-242
`external-destination=` property **469**
in object definition 161, 300
referenced 160, 470, 476
`<external-graphic>` object **152-153**
in object summaries 409, 421
in property definition 449-463, 465-469, 474-478, 480, 484-487, 489-490, 492-500, 503-504
referenced 72, 131, 149, 418, 426-428, 433-434

F

fallback processing 59
<family-name> data type **436**
in property definition 471
`<float>` object **214**
in object summaries 409, 422
in property definition 465, 470, 475-476
referenced 74, 124, 169, 208, 210, 213, 218, 417, 470
`float=` property **470**
in object definition 214
floats 148, 207-229
`floor()` function **405**
in function summaries 403
`<flow>` object **68**
in object summaries 409, 420
in parent content model 65, 412
in property definition 470, 475-476
referenced ?, 46, 60, 71-75, 82, 117, 244, 416
flow semantics 26, 48
`<flow-assignment>` object **247**
in object summaries 409, 420
in parent content model 246, 410
referenced 236, 248, 250, 416
`<flow-map>` object **246**
in object summaries 410, 420
in parent content model 63, 411
in property definition 470
referenced 236, 243, 245, 247, 416
`flow-map-name=` property **470**
in object definition 246
`flow-map-reference=` property **470**
in object definition 65
`flow-name=` property **470**
in object definition 68, 253
referenced 68, 243
`flow-name-reference=` property **470**
in object definition 249
`<flow-name-specifier>` object **249**
in object summaries 410, 420
in parent content model 248, 410
in property definition 470
referenced 236, 416
`<flow-source-list>` object **248**
in object summaries 410, 420
in parent content model 247, 409
referenced 236, 249, 416
`<flow-target-list>` object **250**
in object summaries 410, 420
in parent content model 247, 409
referenced 236, 251, 416
`fo2html.xsl` 30
`<folio-prefix>` object **256**
in object summaries 410, 421
in parent content model 65, 412
referenced 235, 244, 416
`<folio-suffix>` object **258**
in object summaries 410, 421
in parent content model 65, 412
referenced 235, 244, 416

 Copyright © Crane Softwrights Ltd.

`font=` property **470**
in inherited property summary 443
in object definition 109, 111-114, 124, 152, 155, 159, 161, 169, 228, 255, 301, 350, 361, 430
in property definition 471-473, 480
in shorthand definition 446
referenced 58
font size 51, 168
`font-family=` property **471**
in inherited property summary 443
in set of common properties 430
referenced 58, 471
`font-selection-strategy=` property **471**
in inherited property summary 443
in set of common properties 430
referenced 471
`font-size=` property **471**
in inherited property summary 443
in set of common properties 430
referenced 58, 471
`font-size-adjust=` property **471**
in inherited property summary 443
in set of common properties 430
`font-stretch=` property **472**
in inherited property summary 443
in set of common properties 430
referenced 471
`font-style=` property **472**
in inherited property summary 443
in object definition 301
in set of common properties 430
referenced 58, 112, 471
`font-variant=` property **472**
in inherited property summary 443
in set of common properties 430
referenced 58, 471
`font-weight=` property **472**
in inherited property summary 443
in object definition 301
in set of common properties 430
referenced 58, 112, 471
footers
page 232
table 177
`<footnote>` object **219**
in object summaries 410, 422
in property definition 475-476, 494, 496
referenced 75, 112, 124, 169, 208, 210, 213, 217-218, 222, 418, 426
`<footnote-body>` object **222**
in object summaries 410, 422
in parent content model 219, 410
in property definition 475-476, 494, 496
referenced 210, 217, 418, 426
footnote-reference-area 85, 117, 209, 211-212, 217, 317
footnotes 207-229, 388
labeling 207, 218, 388, 395
`force-page-count=` property **473**
in object definition 65
referenced 281
foreign objects 149-159
`format=` property **473**
in object definition 65, 159
referenced 65, 113-114, 159, 255
format tokens 442
formatting 23, 45, 48-49
formatting semantics 26
page numbers 60, 442
running location 389
Formatting Object Tree 51
formatting objects
block-level 70, 71
inline-level 70, 72
neutral 70, 73
out-of-line 70, 74, 85, 207-210
out-of-line inline 70, 75, 207-210
summary list 409-413
<frequency> data type **436**
in property definition 489
`from-nearest-specified-value()` function **405**
in function summaries 403
`from-page-master-region()` function **405**
in function summaries 403
`from-parent()` function **405**
in function summaries 403
referenced 56

Copyright © Crane Softwrights Ltd.

`from-table-column()` function **405**
in function summaries 403
referenced 57, 181, 194, 483
functions 57
library 56
summary list 405-406

G

general purpose XML transformations 24
`generate-id()` (XSLT) 103, 107
ghost list item 139
glyph area 90
glyph-direction 86
`glyph-orientation-horizontal=` property **473**
in inherited property summary 443
in object definition 350
`glyph-orientation-vertical=` property **473**
in inherited property summary 443
in object definition 350
graphics 97, 105, 130, 149-159
`grouping-separator=` property **473**
in object definition 65, 159
`grouping-size=` property **474**
in object definition 65, 159

H

half-leading 99
headers
page 232
table 177
Hebrew 87, 351, 354
`height=` property **474**
in object definition 112, 152, 155, 190, 192, 201, 203, 227-228
referenced 153, 155, 463
hyperlink 8, 10-11, 103, 127, 130, 160-162, 365
hyperlinks to outside information 20, 36-38
Hypertext Markup Language (HTML) 7, 9, 11, 13, 17, 19, 22, 30, 32-34, 174, 319, 363
`hyphenate=` property **474**
in inherited property summary 443
in set of common properties 431
`hyphenation-character=` property **474**
in inherited property summary 443
in set of common properties 431
`hyphenation-keep=` property **474**
in inherited property summary 443
in object definition 109
`hyphenation-ladder-count=` property **474**
in inherited property summary 443
in object definition 109
`hyphenation-push-character-count=` property **474**
in inherited property summary 443
in set of common properties 431
`hyphenation-remain-character-count=` property **475**
in inherited property summary 443
in set of common properties 431

I

<id> data type **436**
in property definition 475
`id=` property **475**
in object definition 61, 65, 67-68, 108-109, 112-114, 140, 142, 144, 146, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 214, 219, 222, 227-228, 253, 255, 305, 350, 361, 369, 373, 378, 380
referenced 54, 103, 106-107, 158, 304
ID/IDREF 103
<idref> data type **436**
in property definition 476, 491
impartation semantics 48, 52
indents 94-95, 148, 165
`index-class=` property **475**
in object definition 61, 65, 67-68, 108-109, 112-114, 140, 142, 144, 146, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 214, 219, 222, 227-228, 253, 255, 305, 350, 361, 369, 373, 378, 380
referenced 303

 Copyright © Crane Softwrights Ltd.

`index-key=` property **475**
in object definition 61, 65, 67-68, 108-109, 112-114, 140, 142, 144, 146, 152, 155, 159, 161, 169, 187, 190, 192, 196, 198-199, 201, 203, 214, 219, 222, 227-228, 253, 255, 305, 350, 361, 369, 373, 378, 380
referenced 303
`<index-key-reference>` object **313**
in object summaries 410, 421
in parent content model 309, 410
in property definition 488, 491
referenced 72, 295, 308-309, 314-315, 418
`<index-page-citation-list>` object **309**
in object summaries 410, 421
in property definition 483
referenced 295, 302, 308, 311-313, 418
`<index-page-citation-list-separator>` object **311**
in object summaries 410, 421
in parent content model 309, 410
referenced 295, 418
`<index-page-citation-range-separator>` object **312**
in object summaries 410, 421
in parent content model 309, 410
referenced 295, 418
`<index-page-number-prefix>` object **314**
in object summaries 410, 421
in parent content model 313, 410
referenced 295, 308, 418
`<index-page-number-suffix>` object **315**
in object summaries 410, 421
in parent content model 313, 410
referenced 295, 308, 418
`<index-range-begin>` object **305**
in object summaries 410, 421
in property definition 475-476
referenced 295, 304, 418
`<index-range-end>` object **307**
in object summaries 410, 421
in property definition 491
referenced 295, 304, 418
`indicate-destination=` property **476**
in object definition 161
inherited properties 57, 89, 104, 261, 366
summary list 443-445
`inherited-property-value()` function **405**
in function summaries 403
referenced 56
`initial-page-number=` property **476**
in object definition 65
referenced 281
`<initial-property-set>` object **111**
in object summaries 410, 421
in property definition 450-463, 465, 467, 469, 471-473, 479-480, 485-487, 489-490, 492-499, 501-504
referenced 81-82, 105, 109, 418, 426-428, 430, 434
`<inline>` object **112**
in object summaries 410, 421
in parent content model 219, 410
in property definition 449-463, 465, 467-469, 471-478, 480, 485-487, 489-490, 492-494, 496-499, 501, 503-504
referenced 72, 81, 106, 217, 418, 426-428, 430, 433-434
`<inline-container>` object **228**
in object summaries 411, 421
in property definition 449, 451-463, 465, 468, 474-478, 480, 484-487, 491-492, 496-497, 504-505
referenced ?, 72, 85, 87, 100, 169, 180, 210, 224, 229, 418, 428, 433-434
`inline-progression-dimension=` property **476**
in object definition 112, 152, 155, 190, 192, 203, 227-228
referenced 94, 181, 482
inline-progression-direction 86, 91, 317, 351
`<instream-foreign-object>` object **155**
in object summaries 411, 421
in property definition 449-463, 465-469, 474-478, 480, 484-487, 489-490, 492-500, 503-504
referenced 72, 131, 149, 418, 426-428, 433-434

Copyright © Crane Softwrights Ltd.

<integer> data type **436**
in property definition 436, 438, 454-455, 457, 461, 484, 504-505
referenced 491
interactive objects 363-381
`internal-destination=` property **476**
in object definition 161, 300
referenced 103, 160, 470, 476
`intrinsic-scale-value=` property **476**
in inherited property summary 443
in object definition 159
referenced 158-159
`intrusion-displace=` property **477**
in inherited property summary 443
in object definition 109, 140, 142, 187, 190, 192, 227
referenced 216

J

Japanese 87

K

<keep> data type **436**
in property definition 477-478
`keep-together=` property **477**
in inherited property summary 443
in object definition 109, 112, 140, 142, 144, 146, 161, 187, 190, 192, 201, 227-228
referenced 323, 488
`keep-with-next=` property **477**
in object definition 109, 112-114, 140, 142, 152, 155, 159, 161, 169, 187, 192, 201, 227-228, 255, 350
referenced 83, 303, 323, 487
`keep-with-previous=` property **477**
in object definition 109, 112-114, 140, 142, 152, 155, 159, 161, 169, 187, 192, 201, 227-228, 255, 350
referenced 83, 323, 487
keeps 323-330
Kleene operator 27

L

`label-end()` function **405**
in function summaries 403
referenced ?, 136-137
landscape 271-272
<language> data type **436**
in property definition 478
`language=` property **478**
in inherited property summary 443
in object definition 65, 159
in set of common properties 431
referenced 505
`last-line-end-indent=` property **478**
in inherited property summary 444
in object definition 109
referenced 105
layout
layout-based formatting 45, 47
semantics 48
`<layout-master-set>` object **63**
in object summaries 411, 420
in parent content model 61, 412
referenced ?, 46, 60, 119, 245-246, 277, 283, 416
`<leader>` object **169-170**
in object summaries 411, 421
in property definition 449-463, 465, 467-469, 471-473, 475-480, 485-487, 489-490, 492-494, 496-501, 503-504
referenced 72, 131, 163, 169, 360, 418, 426-428, 430, 433-434, 478-479
`leader-alignment=` property **478**
in inherited property summary 444
in object definition 169
`leader-length=` property **478**
in inherited property summary 444
in object definition 169
`leader-pattern=` property **478**
in inherited property summary 444
in object definition 169
referenced 170, 494
`leader-pattern-width=` property **479**
in inherited property summary 444
in object definition 169
leaders 130, 163-172, 319
leading 99
left edge 86
`left=` property **479**
in set of common properties 425
referenced 88, 226, 494

 Copyright © Crane Softwrights Ltd.

<length> data type **437**
in property definition 437-438, 448, 451-453, 460, 463-464, 466-469, 471, 474, 476, 478-479, 482-483, 488, 491, 493-495, 498, 500-502, 504
referenced 480-481, 484-487, 502
<length-bp-ip-direction> data type **436**
in property definition 460
<length-conditional> data type **437**
in property definition 454, 456, 458, 461, 484-486
<length-range> data type **437**
in property definition 453, 476, 478
lengths 89
`letter-spacing=` property **479**
in inherited property summary 444
in object definition 111, 113-114, 159, 169, 255, 350, 361
referenced 334
`letter-value=` property **479**
in object definition 65, 159
ligatures 84
line 82
area 90, 98-101
breaking 154
creating 105, 317
elastic 127, 163-166
height 168
numbering 387
`line-height=` property **479**
in inherited property summary 444
in object definition 109, 111-114, 124, 152, 155, 159, 161, 169, 228, 255, 350, 361
referenced 58, 82, 99, 101, 170, 476
`line-height-shift-adjustment=` property **480**
in inherited property summary 444
in object definition 109
referenced 98, 102
`line-stacking-strategy=` property **480**
in inherited property summary 444
in object definition 109
referenced 98, 102
`linefeed-treatment=` property **480**
in inherited property summary 444
in object definition 109
referenced 110, 503
link, see hyperlink
list formatting 132, 148, 395
nested lists 138-139
`<list-block>` object **140**
in object summaries 411, 421
in property definition 450-463, 465, 467, 469, 475-478, 481, 485-487, 489-494, 496-499, 503
referenced 71, 83, 131, 138-139, 141-142, 148, 387, 417, 426-428, 432, 434
`<list-item>` object **142**
in object summaries 411, 421
in parent content model 140, 411
in property definition 450-463, 467, 469, 475-478, 481, 485-487, 489-490, 492-494, 496-499, 503
referenced 131, 139, 143-144, 146, 417, 426-428, 432, 434
`<list-item-body>` object **146**
in object summaries 411, 421
in parent content model 142, 411
in property definition 475-477, 494, 496
referenced 131, 138, 147, 417, 426
`<list-item-label>` object **144**
in object summaries 411, 421
in parent content model 142, 411
in property definition 475-477, 494, 496
referenced 131, 145, 417, 426
lists of effective pages 47
loose-leaf publishing 47

M

main-reference-area 85, 117
`margin=` property **480**
in object definition 432
in property definition 481
in shorthand definition 446
`margin-bottom=` property **480**
in set of common properties 432
referenced ?, 88, 480
`margin-left=` property **481**
in set of common properties 432
referenced ?, 88, 480

Copyright © Crane Softwrights Ltd.

`margin-right=` property **481**
in set of common properties 432
referenced ?, 88, 480
`margin-top=` property **481**
in set of common properties 432
referenced ?, 88, 480
margins 88, 94-95
`<marker>` object **264**
in object summaries 411, 422
in property definition 481
referenced 68, 108-109, 112, 124, 140, 142, 144, 146, 161, 169, 187, 190, 192, 196, 198-199, 203, 213, 218, 227-228, 235, 259, 262, 264, 361, 418, 481, 493
`marker-class-name=` property **481**
in object definition 264
referenced 259, 493
markers 259-263, 384, 386
markup
syntax preservation 24
`master-name=` property **481**
in object definition 119, 283
referenced 115
`master-reference=` property **482**
in object definition 65, 286-287, 290
referenced 65, 276
Mathematical Markup Language (MathML) 54, 149
`max()` function **405**
in function summaries 403
`max-height=` property **482**
`max-width=` property **482**
`maximum-repeats=` property **482**
in object definition 287-288
`media-usage=` property **482**
in object definition 61
referenced 61
`merge-pages-across-index-key-references=` property **482**
in inherited property summary 444
in object definition 309
referenced 309
`merge-property-values()` function **405**
in function summaries 403
referenced 56-57, 367
`merge-ranges-across-index-key-references=` property **483**
in inherited property summary 444
in object definition 309
referenced 309
`merge-sequential-page-numbers=` property **483**
in inherited property summary 444
in object definition 309
referenced 309
`min()` function **405**
in function summaries 403
`min-height=` property **483**
`min-width=` property **483**
`<multi-case>` object **378**
in object summaries 411, 420
in parent content model 373, 411
in property definition 464, 475-476, 494, 496, 498
referenced ?, 364, 371, 380, 419, 426, 450, 464, 498-499
`<multi-properties>` object **367**
in object summaries 411, 420
in property definition 494, 496
referenced ?, 73, 104, 108, 364-365, 369, 419, 426, 448
`<multi-property-set>` object **369**
in object summaries 411, 420
in parent content model 367, 411
in property definition 448, 475-476
referenced ?, 364-367, 419
`<multi-switch>` object **373**
in object summaries 411, 420
in property definition 450, 475-476, 494, 496
referenced ?, 73, 364, 371, 373, 378, 419, 426, 450
`<multi-toggle>` object **380**
in object summaries 411, 420
in property definition 475-476, 494, 496, 499
referenced ?, 72, 364, 371, 378, 419, 426, 464, 499

 Copyright © Crane Softwrights Ltd.

N
<name> data type **437**
in property definition 464, 466, 470, 481-482, 491, 493, 499
namespaces 19, 31, 53-54, 149, 155-156
default namespace 31
navigation tools (as within a book) 12, 232, 262
normal-flow-reference-area 85, 117, 317, 339
normal/non-normal areas 91
<number> data type **437**
in property definition 436, 438, 466, 471, 474-476, 479, 482-484, 489, 493, 498-499, 503
referenced 498
`number-columns-repeated=` property **483**
in object definition 194
`number-columns-spanned=` property **483**
in object definition 194, 203
referenced 185, 466
`number-rows-spanned=` property **483**
in object definition 203
referenced 185

O
`odd-or-even=` property **484**
in object definition 290
referenced 280
optimum 55, 163
orphans 321-322
`orphans=` property **484**
in inherited property summary 444
in object definition 109
referenced 321
outdents 165
overflow 243
`overflow=` property **484**
in object definition 121, 152, 155, 227-228, 238-239, 241-242
referenced 153, 155, 243

P
`padding=` property **484**
in object definition 429
in property definition 485-487
in shorthand definition 446
referenced 121, 238-239, 241-242
padding width 93
`padding-after=` property **484**
in set of common properties 428
referenced ?
`padding-before=` property **485**
in set of common properties 428
referenced ?
`padding-bottom=` property **485**
in set of common properties 428
referenced 484
`padding-end=` property **485**
in set of common properties 428
referenced ?
`padding-left=` property **486**
in set of common properties 428
referenced 484
padding-rectangle 77, 91-95, 213, 237, 338
`padding-right=` property **486**
in set of common properties 428
referenced 484
`padding-start=` property **486**
in set of common properties 428
referenced ?
`padding-top=` property **487**
in set of common properties 428
referenced 484
page
choreography 234
count (total) 107
dimensions 7
fidelity, see rendering; page fidelity
geometry 63, 79-80, 82, 115, 233, 243, 271-275, 276-292, 278, 328, 383, 390
layout 115-116
parity 233, 280-281, 319
plan 270, 292
position 233, 280
region, see regions
titles 123
page number citations 103, 232
Page Sequence Master Interleave (PSMI) 271-275, 390

Copyright © Crane Softwrights Ltd.

`page-break-after=` property **487**
in object definition 109, 112-114, 140, 142, 152, 155, 159, 161, 169, 187, 192, 201, 227-228, 255, 350
in property definition 463, 477
in shorthand definition 446
`page-break-before=` property **487**
in object definition 109, 112-114, 140, 142, 152, 155, 159, 161, 169, 187, 192, 201, 227-228, 255, 350
in property definition 463, 477
in shorthand definition 446
`page-break-inside=` property **487**
in inherited property summary 444
in object definition 109, 112, 140, 142, 144, 146, 161, 187, 190, 192, 201, 227-228
in property definition 477
in shorthand definition 446
`page-citation-strategy=` property **488**
in object definition 114
referenced 107, 114
`page-height=` property **488**
in object definition 119
referenced 482, 488, 495
`<page-number>` object **255**
in object summaries 411, 421
in property definition 449-463, 467-469, 471-473, 475-480, 485-487, 489-490, 492-504
referenced ?, 72, 235, 244, 416, 426-428, 430, 433-434
`<page-number-citation>` object **113**
in object summaries 411, 421
in property definition 449-463, 467-469, 471-473, 475-480, 485-487, 489-504
referenced ?, 13, 72, 81, 106, 302, 418, 426-428, 430, 433-434
`<page-number-citation-last>` object **114**
in object summaries 411, 421
in property definition 449-463, 467-469, 471-473, 475-480, 485-504
referenced 72, 81, 107, 418, 426-428, 430, 433-434
`page-number-treatment=` property **488**
in inherited property summary 444
in object definition 313
referenced 313
`page-position=` property **488**
in object definition 290
referenced 280
`<page-sequence>` object **65**
in object summaries 412, 420
in parent content model 61, 67, 412
in property definition 467, 470, 473-476, 478-479, 482, 491, 505
referenced ?, 46, 60, 68, 74-75, 107, 113-114, 124, 212, 243-245, 253, 255-256, 258, 276-277, 416, 470, 482
`<page-sequence-master>` object **283**
in object summaries 412, 420
in parent content model 63, 411
in property definition 482
referenced ?, 236, 277, 286-288, 416
`<page-sequence-wrapper>` object **67**
in object summaries 412, 420
in parent content model 61, 67, 412
in property definition 475-476
referenced 46, 60, 65, 67, 416
`page-width=` property **488**
in object definition 119
referenced 482, 488, 495
pagination 12, 231-292
semantics 13, 21, 26, 48, 51
`pause=` property **488**
in object definition 427
in property definition 489
in shorthand definition 446
`pause-after=` property **488**
in set of common properties 427
referenced 488
`pause-before=` property **489**
in set of common properties 427
referenced 488
<percentage> data type **437**
in property definition 437, 448-449, 451-453, 463, 466-467, 469, 471, 474, 476-479, 482-483, 488-489, 491, 493, 496, 498, 500-504
referenced 480-481, 484-487, 502

 Copyright © Crane Softwrights Ltd.

percentages 89, 99, 102
perimeter regions 79-80, 85, 116, 209, 213, 218, 237
`pitch=` property **489**
in inherited property summary 444
in set of common properties 427
`pitch-range=` property **489**
in inherited property summary 444
in set of common properties 427
`play-during=` property **490**
in set of common properties 427
portability problems 54, 58, 84, 86, 99, 123
Portable Document Format (PDF) 13, 32-34, 40, 49
`position=` property **490**
in object definition 425, 434
in property definition 448, 492
in shorthand definition 446
post-order traversal 83
pre-order traversal 83
precedence 94, 331-333
areas 89, 318
table borders 184
`precedence=` property **490**
in object definition 238-239
referenced 79, 238-239
processing model 45, 50-52, 82
program listings 110
properties 54
compound 58
direct vs. constraint 55
inherited 57, 194, 443-445
property classes 45
shorthand 51, 58, 446
summary list 447-505
value expressions 56
`proportional-column-width()` function **406**
in function summaries 403
referenced 194
`provisional-distance-between-starts=` property **490**
in inherited property summary 444
in object definition 140
referenced 137, 140
`provisional-label-separation=` property **491**
in inherited property summary 444
in object definition 140
referenced 137, 140

Q

R

recommendations 35
`ref-id=` property **491**
in object definition 113-114, 159, 307
referenced 103, 106-107, 113-114, 158-159, 304
`ref-index-key=` property **491**
in object definition 313
referenced 308
reference area 85-87, 90, 224
summary list of reference areas 85
reference orientation 78, 87, 91, 116
`reference-orientation=` property **491**
in object definition 65, 119, 121, 227-228, 238-239, 241-242
referenced ?, 87, 94, 224, 405, 482
references 82, 103
Refined Formatting Object Tree 51
refinement 51, 54
`<region-after>` object **239**
in object summaries 412, 420
in parent content model 119, 412
in property definition 451-463, 465, 468-469, 484-487, 490-491, 505
referenced ?, 79, 116, 235, 252, 416, 428
`<region-before>` object **238**
in object summaries 412, 420
in parent content model 119, 412
in property definition 451-463, 465, 468-469, 484-487, 490-491, 505
referenced ?, 79, 116, 235, 416, 428
`<region-body>` object **121**
in object summaries 412, 420
in parent content model 119, 412
in property definition 451-463, 465-466, 468-469, 481, 484-487, 491, 496, 498, 505
referenced 79, 81, 116, 282, 416, 428, 432

Copyright © Crane Softwrights Ltd.

`<region-end>` object **242**
in object summaries 412, 420
in parent content model 119, 412
in property definition 451-463, 465, 468-469, 484-487, 491, 505
referenced 79, 116, 235, 252, 416, 428
`region-name=` property **491**
in object definition 121, 238-239, 241-242
referenced 116, 121, 238-239, 241-242
`region-name-reference=` property **491**
in object definition 251
`<region-name-specifier>` object **251**
in object summaries 412, 420
in parent content model 250, 410
in property definition 492
referenced 236, 416
`<region-start>` object **241**
in object summaries 412, 420
in parent content model 119, 412
in property definition 451-463, 465, 468-469, 484-487, 491, 505
referenced 79, 116, 235, 252, 416, 428
regions 79-80, 82, 116, 209, 232, 237, 244
`relative-align=` property **492**
in inherited property summary 444
in object definition 142, 203
referenced 135, 142, 186, 468
`relative-position=` property **492**
in set of common properties 434
referenced 490
rendering 45, 48-49
agent 52
page fidelity 40, 84
semantics 49
`rendering-intent=` property **492**
in object definition 347
RenderX XEP 40, 54
`<repeatable-page-master-alternatives>` object **288**
in object summaries 412, 420
in parent content model 283, 412
in property definition 482
referenced ?, 236, 279, 290, 416
`<repeatable-page-master-reference>` object **287**
in object summaries 412, 420
in parent content model 283, 412
in property definition 482
referenced ?, 236, 279, 416
result tree 29
`retrieve-boundary=` property **492**
in object definition 267
referenced 261, 267
`retrieve-boundary-within-table=` property **492**
in object definition 269
referenced 263, 269
`retrieve-class-name=` property **493**
in object definition 267, 269
referenced 261
`<retrieve-marker>` object **267**
in object summaries 412, 422
in property definition 492-493
referenced 73, 83, 235, 261, 264, 416, 493
`retrieve-position=` property **493**
in object definition 267
referenced ?, 261, 267
`retrieve-position-within-table=` property **493**
in object definition 269
referenced 263, 269
`<retrieve-table-marker>` object **269**
in object summaries 412, 422
in property definition 493
referenced 73, 235, 263, 417, 493
`rgb()` function **406**
in function summaries 403
`rgb-icc()` function **406**
in function summaries 403
referenced 347
`richness=` property **493**
in inherited property summary 444
in set of common properties 427
right edge 86
`right=` property **493**
in set of common properties 425
referenced 88, 226
`role=` property **494**
in set of common properties 426

 Copyright © Crane Softwrights Ltd.

roman numeral numbering 442
<root> object **61**
in object summaries 412, 420
in property definition 475-476, 482, 494, 496
referenced ?, 46, 60, 63, 65, 67, 107, 296, 298, 348, 416, 426, 496
rotating characters, see glyph-direction
round() function **406**
in function summaries 403
referenced 56
rows (table)
building strategies 182, 183, 196, 198-199
spanning 177, 185
rule-style= property **494**
in inherited property summary 444
in object definition 169
rule-thickness= property **494**
in inherited property summary 444
in object definition 169

S

Scalable Vector Graphics (SVG) 19, 54, 127, 149, 156
scale-option= property **494**
in inherited property summary 444
in object definition 159
referenced 158-159
scaling= property **494**
in object definition 152, 155
referenced 150, 153, 155
scaling-method= property **494**
in object definition 152, 155
<scaling-value-citation> object **159**
in object summaries 412, 421
in property definition 449-463, 467-469, 471-480, 485-487, 489-504
referenced 72, 131, 158, 419, 426-428, 430, 433-434
score-spaces= property **495**
in inherited property summary 444
in object definition 111, 113-114, 159, 255, 350, 361
<script> data type **437**
in property definition 495
script= property **495**
in inherited property summary 444
in set of common properties 431
semantic information processing 18
sequence
sequence constructor, see template
serialization of result tree 29
server delivery 32-34
<shape> data type **437**
in property definition 465
shift-direction 86, 102
shorthand properties 58
summary list 446
show-destination= property **495**
in object definition 161
<simple-page-master> object **119**
in object summaries 412, 420
in parent content model 63, 411
in property definition 469, 481-482, 488, 491, 496, 498, 505
referenced ?, 81, 115, 121, 238-239, 241-242, 276, 383, 416, 432
<single-page-master-reference> object **286**
in object summaries 412, 420
in parent content model 283, 412
in property definition 482
referenced ?, 236, 279, 416
size= property **495**
in object definition 119
in property definition 488
in shorthand definition 446
source-document= property **495**
in set of common properties 426
<space> data type **437**
in property definition 479, 496, 504
space-after= property **496**
in set of common properties 432
referenced ?, 85, 88, 92-93, 331, 333, 387
space-before= property **496**
in set of common properties 432
referenced ?, 53, 55, 85, 88, 92-93, 331, 333, 387, 465
space-end= property **496**
in set of common properties 433
referenced 88, 93, 331

Copyright © Crane Softwrights Ltd.

`space-start=` property **496**
in set of common properties 433
referenced 88, 93, 331
spacing 88, 89
collapsing white space, see white-space characters; collapsing
discarding space areas, see discarding space
modifying character spacing 334
`span=` property **497**
in object definition 109, 227
span-reference-area 85, 117
spanning
page columns, see span-reference-area
table columns/cells, see columns (table); spanned
table rows, see rows (table); spanning
`speak=` property **497**
in inherited property summary 444
in set of common properties 427
`speak-header=` property **497**
in inherited property summary 444
in set of common properties 427
`speak-numeral=` property **497**
in inherited property summary 444
in set of common properties 427
`speak-punctuation=` property **497**
in inherited property summary 444
in set of common properties 427
`speech-rate=` property **498**
in inherited property summary 444
in set of common properties 427
spreadsheets 149
`src=` property **498**
in object definition 152, 347
referenced 151, 347
stacking 70, 85, 91-95, 96-97, 209, 317
Standard Page Description Language (SPDL) 49
start edge 86
`start-indent=` property **498**
in inherited property summary 444
in set of common properties 432
referenced 57, 92-94, 135, 137, 146, 186, 216
`starting-state=` property **498**
in object definition 300, 378
referenced 297, 300
`starts-row=` property **498**
in object definition 203
referenced 182
static content 80, 212, 232, 243-244, 386
`<static-content>` object **253**
in object summaries 412, 420
in parent content model 65, 412
in property definition 470, 475-476
referenced 82, 212, 235, 244, 252, 416
`stress=` property **499**
in inherited property summary 444
in set of common properties 427
<string> data type **438**
in property definition 464, 466, 473, 475, 491, 494, 500
referenced 500
styling structured information 23, 27
sub-regions 211
sub-sequence 277, 279-280
subscript/superscript 102, 135, see also shift-direction
`suppress-at-line-break=` property **499**
in object definition 350
`switch-to=` property **499**
in object definition 380
`system-color()` function **406**
in function summaries 403
`system-font()` function **406**
in function summaries 403
referenced 471

T

`<table>` object **192**
in object summaries 413, 421
in parent content model 187, 413
in property definition 450-463, 465, 467, 469, 474-478, 481, 485-487, 489-490, 492-494, 496-499, 503-505
referenced ?, 71, 85, 178, 180, 184, 186, 194, 196, 198-199, 417, 426-428, 432, 434

 Copyright © Crane Softwrights Ltd.

`<table-and-caption>` object **187**
in object summaries 413, 421
in property definition 450-465, 467, 469, 475-478, 481, 485-487, 489-490, 492-494, 496-500, 503
referenced ?, 71, 178, 180, 186, 190, 192, 417, 426-428, 432, 434
`<table-body>` object **199**
in object summaries 413, 421
in parent content model 192, 413
in property definition 450-463, 467, 469, 475-476, 485-487, 489-490, 492-494, 496-499, 503
referenced ?, 178, 180, 182, 184, 186, 201, 203, 417, 426-428, 434
`<table-caption>` object **190**
in object summaries 413, 421
in parent content model 187, 413
in property definition 450-463, 467, 469, 474-477, 485-487, 489-490, 492-494, 496-499, 503-504
referenced ?, 85, 178, 180, 186, 417, 426-428, 434
`<table-cell>` object **203**
in object summaries 413, 421
in parent content model 196, 198-199, 201, 413
in property definition 450-463, 466-469, 474-476, 483-487, 489-490, 492-494, 496-499, 503-504
referenced ?, 85, 178, 182, 184, 417, 426-428, 434, 466, 483, 500
`<table-column>` object **194**
in object summaries 413, 421
in parent content model 192, 413
in property definition 451-463, 466, 483, 485-487, 503
referenced ?, 178, 181, 405, 417, 428, 466, 483
`<table-footer>` object **198**
in object summaries 413, 421
in parent content model 192, 413
in property definition 450-463, 467, 469, 475-476, 485-487, 489-490, 492-494, 496-499, 503
referenced ?, 178, 182, 184, 201, 203, 417, 426-428, 434, 499
`<table-header>` object **196**
in object summaries 413, 421
in parent content model 192, 413
in property definition 450-463, 467, 469, 475-476, 485-487, 489-490, 492-494, 496-499, 503
referenced ?, 178, 182, 184, 186, 201, 203, 417, 426-428, 434, 499
`table-layout=` property **499**
in object definition 192
referenced 181
`table-omit-footer-at-break=` property **499**
in object definition 192
`table-omit-header-at-break=` property **499**
in object definition 192
`<table-row>` object **201**
in object summaries 413, 421
in parent content model 196, 198-199, 413
in property definition 450-463, 467, 469, 474-478, 485-487, 489-490, 492-494, 496-499, 503
referenced ?, 178, 182, 184, 203, 417, 426-428, 434, 463
tables 148, 174-205
tables of content 103, 127, 164, 166, 270, 357
`target-presentation-context=` property **499**
in object definition 161
`target-processing-context=` property **500**
in object definition 161
`target-stylesheet=` property **500**
in object definition 161
template 24-25
TeX 49

Copyright © Crane Softwrights Ltd.

`text-align=` property **500**
in inherited property summary 444
in object definition 109, 152, 155, 187
referenced 105, 153-155, 164, 186-187, 190
`text-align-last=` property **500**
in inherited property summary 444
in object definition 109
referenced 105, 164
`text-altitude=` property **500**
in object definition 109, 113-114, 159, 169, 255, 350
`text-decoration=` property **500**
in object definition 111-114, 159, 255, 350
referenced 57, 112
`text-depth=` property **501**
in object definition 109, 113-114, 159, 169, 255, 350
`text-indent=` property **501**
in inherited property summary 444
in object definition 109
referenced 105, 216
`text-shadow=` property **501**
in object definition 111, 113-114, 159, 169, 255, 350
`text-transform=` property **501**
in inherited property summary 444
in object definition 111, 113-114, 159, 255, 350
<time> data type **438**
in property definition 488-489
`<title>` object **124**
in object summaries 413, 420
in parent content model 65, 412
in property definition 450-463, 465, 467, 469, 471-473, 480, 485-487, 489-490, 493-494, 496-499, 503
referenced 81, 85, 416, 426-428, 430, 433
top edge 86
`top=` property **502**
in set of common properties 425
referenced 88, 226
total page count, see page; count (total)
traits 51, 53
formatting 51
rendering 51
transforming information 23
`treat-as-word-space=` property **502**
in object definition 350
tuples 174, 179
Turing complete 24
typographical conventions 2

U

Unicode 87, 352-353, 442
`unicode-bidi=` property **502**
in object definition 361
referenced 353, 361
Universal Character Set (UCS), see Unicode
Universal Resource Identifier 19, 151, 320
<uri-specification> data type **438**
in property definition 451, 467, 470, 490, 494-495, 498, 500
referenced 490, 494
`url()` 151
US letter page dimensions 135-136

V

vendor questions 507-509
vertical alignment 100, 105, 135
`vertical-align=` property **502**
in object definition 112-114, 152, 155, 159, 161, 169, 228, 255, 350
in property definition 449, 453, 468
in shorthand definition 446
referenced 100, 105, 190
viewport area 90, 115
`visibility=` property **502**
in inherited property summary 444
in object definition 109, 112-114, 124, 159, 169, 194, 196, 198-199, 201, 255, 350
visual media 26, 59
vocabulary, XML 10, 18-19
`voice-family=` property **503**
in inherited property summary 444
in set of common properties 427
`volume=` property **503**
in inherited property summary 444
in set of common properties 427

W

W3C XSL Working Group 23
white-space characters
collapsing 110, 154

 Copyright © Crane Softwrights Ltd.

`white-space=` property **503**
in inherited property summary 444
in shorthand definition 446
referenced 110
`white-space-collapse=` property **503**
in inherited property summary 444
in object definition 109
referenced 110, 503
`white-space-treatment=` property **504**
in inherited property summary 444
in object definition 109
referenced 110, 503
widows 321-322
`widows=` property **504**
in inherited property summary 444
in object definition 109
referenced 321
`width=` property **504**
in object definition 112, 152, 155, 190, 192, 203, 227-228
referenced 150, 153, 155, 479
`word-spacing=` property **504**
in inherited property summary 444
in object definition 111, 113-114, 159, 169, 255, 350, 361
`wrap-option=` property **504**
in inherited property summary 445
in object definition 109, 112-114, 159, 255
referenced 110, 503
`<wrapper>` object **108**
in object summaries 413, 422
in parent content model 367, 411
in property definition 475-476
referenced ?, 73, 81, 104, 106, 366-367, 417, 419
wrapping lines 110
writing direction 78, 85-87, 116
writing mode, see writing direction
`writing-mode=` property **504**
in inherited property summary 445
in object definition 65, 119, 121, 192, 227-228, 238-239, 241-242
referenced 87, 115, 224, 352, 405, 468, 482
WSSSL 23

X

XML, see Extensible Markup Language (XML)
XML processor 24
`xml.lang=` property **505**
in inherited property summary 445
in object definition 65, 159, 431
in property definition 467, 478
in shorthand definition 446
`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) 394
XSLT, see Extensible Stylesheet Language Transformations (XSLT)
XSLT processor 24

Y

Z

`z-index=` property **505**
in object definition 227, 344
referenced 226, 384
zero-width space (``) 320
ZIP file of examples 5

Copyright © Crane Softwrights Ltd.