

User agent refs

by Peter B. West

Table of contents

1	User Agent references in XSLFO.....	4
1.1	4.9.2 Viewport Geometry.....	4
1.2	5.1.3 Actual Values.....	4
1.3	5.5.7 Font Properties.....	4
1.4	5.9.13.1 Pixels.....	4
1.5	5.10.4 Property Value Functions.....	4
1.6	6.3 Formatting Objects Summary.....	5
1.7	6.4.1.2 Page-masters.....	5
1.8	6.4.20 fo:title.....	5
1.9	6.6.3 fo:character.....	5
1.10	6.9 Dynamic Effects: Link and Multi Formatting Objects.....	5
1.11	6.10 Out-of-Line Formatting Objects.....	6
1.12	6.10.2 fo:float.....	6
1.13	6.10.3 fo:footnote.....	6
1.14	7.3 Reference Rectangle for Percentage Computations.....	7
1.15	7.6.5 "pause-after" 7.6.6 "pause-before" 7.6.17 "voice-family".....	7
1.16	7.7.1 "background-attachment".....	7
1.17	7.7.9 "border-before-width".....	7
1.18	7.7.19 "border-top-color".....	7
1.19	7.7.20 "border-top-style".....	8
1.20	7.7.21 "border-top-width".....	8
1.21	7.8.2 "font-family".....	8

1.22 7.8.3 "font-selection-strategy"	8
1.23 7.8.4 "font-size"	8
1.24 7.8.8 "font-variant"	9
1.25 7.8.9 "font-weight"	9
1.26 7.13.1 "alignment-adjust"	9
1.27 7.13.3 "baseline-shift"	9
1.28 7.13.5 "dominant-baseline"	10
1.29 7.14.11 "scaling-method"	10
1.30 7.14.12 "width"	10
1.31 7.15.4 "line-height"	11
1.32 7.15.9 "text-align"	11
1.33 7.15.11 "text-indent"	11
1.34 7.16.2 "letter-spacing"	11
1.35 7.16.4 "text-decoration"	12
1.36 7.16.6 "text-transform"	12
1.37 7.16.8 "word-spacing"	12
1.38 7.17.1 "color"	12
1.39 7.17.3 "rendering-intent"	12
1.40 7.20.2 "overflow"	13
1.41 7.21.2 "leader-pattern"	13
1.42 7.21.4 "leader-length"	13
1.43 7.25.11 "media-usage"	13
1.44 7.25.13 "page-height"	14
1.45 7.25.15 "page-width"	14
1.46 7.26.5 "border-separation"	14
1.47 7.26.7 "caption-side"	15
1.48 7.27.2 "glyph-orientation-horizontal"	15
1.49 7.27.3 "glyph-orientation-vertical"	15
1.50 7.27.6 "unicode-bidi"	15

User agent refs

1.51 7.28.1 "content-type"	15
1.52 7.29.5 "border-color"	16
1.53 7.29.9 "border-spacing"	16
1.54 7.29.13 "font"	16
1.55 7.29.19 "pause"	16
1.56 7.29.21 "size"	16
1.57 7.29.23 "white-space"	16

1. User Agent references in XSLFO

1.1. 4.9.2 Viewport Geometry

If the block-progression-dimension of the reference-area is larger than that of the viewport-area and the overflow trait for the reference-area is scroll, then the inline-scroll-amount and block-scroll-amount are determined by a scrolling mechanism, if any, provided by the **user agent**. Otherwise, both are zero.

1.2. 5.1.3 Actual Values

A computed value is in principle ready to be used, but a user agent may not be able to make use of the value in a given environment. For example, a **user agent** may only be able to render borders with integer pixel widths and may, therefore, have to adjust the computed width to an integral number of media pixels.

1.3. 5.5.7 Font Properties

There is no XSL mechanism to specify a particular font; instead, a selected font is chosen from the fonts available to the **User Agent** based on a set of selection criteria. The selection criteria are the following font properties: "font-family", "font-style", "font-variant", "font-weight", "font-stretch", and "font-size", plus, for some formatting objects, one or more characters.

1.4. 5.9.13.1 Pixels

If the **User Agent** chooses a measurement for a 'px' that does not match an integer number of device dots in each axis it may produce undesirable effects...

1.5. 5.10.4 Property Value Functions

1.5.1. Function: object merge-property-values(NCName)

The merge-property-values function returns a value of the property whose name matches the argument, or if omitted for the property for which the expression is being evaluated. The value returned is the specified value on the last fo:multi-property-set, of the parent fo:multi-properties, that applies to the **User Agent** state. If there is no such value, the computed value of the parent fo:multi-properties is returned...

The test for applicability of a **User Agent** state is specified using the "active-state" property.

1.6. 6.3 Formatting Objects Summary

1.6.1. multi-property-set

The fo:multi-property-set is used to specify an alternative set of formatting properties that, dependent on a **User Agent** state, are applied to the content.

1.6.2. title

The fo:title formatting object is used to associate a title with a given page-sequence. This title may be used by an interactive **User Agent** to identify the pages. For example, the content of the fo:title can be formatted and displayed in a "title" window or in a "tool tip".

1.7. 6.4.1.2 Page-masters

... When pages are used with a **User Agent** such as a Web browser, it is common that the each document has only one page. The viewport used to view the page determines the size of the page. When pages are placed on non-interactive media, such as sheets of paper, pages correspond to one or more of the surfaces of the paper.

1.8. 6.4.20 fo:title

1.8.1. Common Usage:

... This title may be used by an interactive **User Agent** to identify the pages.

1.9. 6.6.3 fo:character

1.9.1. Constraints:

The dimensions of the areas are determined by the font metrics for the glyph.

When formatting an fo:character with a "treat-as-word-space" value of "true", the **User Agent** may use a different method for determining the inline-progression-dimension of the area.

1.10. 6.9 Dynamic Effects: Link and Multi Formatting Objects

1.10.1. 6.9.1 Introduction

Dynamic effects, whereby user actions (including **User Agent** state) can influence the behavior

and/or representation of portions of a document, can be achieved through the use of the formatting objects included in this section:

- One-directional single-target links.
- The ability to switch between the display of two or more formatting object subtrees. This can be used for, e.g., expandable/collapsible table of contents, display of an icon or a full table or graphic.
- The ability to switch between different property values, such as color or font-weight, depending on a **User Agent** state, such as "hover".

1.11. 6.10 Out-of-Line Formatting Objects

1.11.1. 6.10.1.3 Conditional Sub-Regions

... There may be limits on how much space conditionally generated areas can borrow from the region-reference-area. It is left to the **user agent** to decide these limits.

... An interactive **user agent** may choose to create "hot links" to the footnotes from the footnote-citation, or create "hot links" to the before-floats from an implicit citation, instead of realizing conditional sub-regions.

1.12. 6.10.2 fo:float

1.12.1. Constraints:

... The **user agent** may make its own determination, after taking into account the intrusion adjustments caused by one or more overlapping side-floats, that the remaining space in the inline-progression-direction is insufficient for the next side-float or normal block-area. The **user agent** may address this by causing the next side-float or normal block-area to "clear" one of the relevant side-floats, as described in the "clear" property description, so the intrusion adjustment is sufficiently reduced. Of the side-floats that could be cleared to meet this constraint, the side-float that is actually cleared must be the one whose after-edge is closest to the before-edge of the parent reference-area.

The **user agent** may determine sufficiency of space by using a fixed length, or by some heuristic such as whether an entire word fits into the available space, or by some combination, in order to handle text and images.

1.13. 6.10.3 fo:footnote

1.13.1. Constraints:

... The second block-area and any additional block-areas returned by an fo:footnote must be placed on the immediately subsequent pages to the page containing the first block-area returned by the fo:footnote, before any other content is placed. If a subsequent page does not contain a region-body, the **user agent** must use the region-master of the last page that did contain a region-body to hold the additional block-areas.

1.14. 7.3 Reference Rectangle for Percentage Computations

...

1.14.1. Exceptions ...

5. When the absolute-position is "fixed", the containing block is defined by the nearest ancestor viewport area. If there is no ancestor viewport area, the containing block is defined by the **user agent**.

1.15. 7.6.5 "pause-after" 7.6.6 "pause-before" 7.6.17 "voice-family"

Initial: depends on **user agent**

1.16. 7.7.1 "background-attachment"

1.16.1. fixed

... **User agents** may treat fixed as scroll. However, it is recommended they interpret fixed correctly, at least for the HTML and BODY elements, since there is no way for an author to provide an image only for those browsers that support fixed.

1.17. 7.7.9 "border-before-width"

1.17.1. <length-conditional>

... If border-before-width is specified using one of the width keywords the .conditional component is set to "discard" and the .length component to a **User Agent** dependent length.

1.18. 7.7.19 "border-top-color"

1.18.1. <color>

... If an element's border color is not specified with a "border" property, **user agents** must use the

value of the element's "color" property as the computed value for the border color.

1.19. 7.7.20 "border-top-style"

Conforming HTML **user agents** may interpret 'dotted', 'dashed', 'double', 'groove', 'ridge', 'inset', and 'outset' to be 'solid'.

1.20. 7.7.21 "border-top-width"

1.20.1. thin ... medium ... thick ...

... The interpretation of the first three values depends on the **user agent**.

1.21. 7.8.2 "font-family"

Initial: depends on **user agent**

1.22. 7.8.3 "font-selection-strategy"

There is no XSL mechanism to specify a particular font; instead, a selected font is chosen from the fonts available to the **User Agent** based on a set of selection criteria. The selection criteria are the following font properties: "font-family", "font-style", "font-variant", "font-weight", "font-stretch", and "font-size", plus, for some formatting objects, one or more characters.

... This fallback may be to seek a match using a **User Agent** default "font-family", or it may be a more elaborate fallback strategy where, for example, "Helvetica" would be used as a fallback for "Univers".

If no match has been found for a particular character, there is no selected font and the **User Agent** should provide a visual indication that a character is not being displayed (for example, using the 'missing character' glyph).

1.23. 7.8.4 "font-size"

1.23.1. <absolute-size>

An <absolute-size> keyword refers to an entry in a table of font sizes computed and kept by the **user agent**. Possible values are:

[xx-small | x-small | small | medium | large | x-large | xx-large]

1.23.2. <relative-size>

User agent refs

A <relative-size> keyword is interpreted relative to the table of font sizes and the font size of the parent element. Possible values are:

[larger | smaller]

For example, if the parent element has a font size of "medium", a value of "larger" will make the font size of the current element be "large". If the parent element's size is not close to a table entry, the **user agent** is free to interpolate between table entries or round off to the closest one. The **user agent** may have to extrapolate table values if the numerical value goes beyond the keywords.

1.23.3. <length>

A length value specifies an absolute font size (that is independent of the **user agent's** font table).

1.24. 7.8.8 "font-variant"

1.24.1. small-caps

... If a genuine small-caps font is not available, **user agents** should simulate a small-caps font...

1.25. 7.8.9 "font-weight"

1.25.1. XSL modifications to the CSS definition:

... The association of other weights within a family to the numerical weight values is intended only to preserve the ordering of weights within that family. **User agents** must map names to values in a way that preserves visual order; a face mapped to a value must not be lighter than faces mapped to lower values. There is no guarantee on how a **user agent** will map fonts within a family to weight values. However, the following heuristics...

1.26. 7.13.1 "alignment-adjust"

1.26.1. auto

... If the baseline-identifier does not exist in the baseline-table for the glyph or other inline-area, then the **User Agent** may either use heuristics to determine where that missing baseline would be or may use the dominant-baseline as a fallback.

1.27. 7.13.3 "baseline-shift"

1.27.1. sub/super

... Because in most fonts the subscript position is normally given relative to the "alphabetic" baseline, the **User Agent** may compute the effective position for sub/superscripts [*sub: spec typo!*] when some other baseline is dominant. ... If there is no applicable font data the **User Agent** may use heuristics to determine the offset.

1.28. 7.13.5 "dominant-baseline"

... If there is no baseline-table in the nominal font or if the baseline-table lacks an entry for the desired baseline, then the **User Agent** may use heuristics to determine the position of the desired baseline.

1.29. 7.14.11 "scaling-method"

1.29.1. auto

The **User Agent** is free to choose either resampling, integer scaling, or any other scaling method.

1.29.2. integer-pixels

The **User Agent** should scale the image such that each pixel in the original image is scaled to the nearest integer number of device-pixels that yields an image less-than-or-equal-to the image size derived from the content-height, content-width, and scaling properties.

1.29.3. resample-any-method

The **User Agent** should resample the supplied image to provide an image that fills the size derived from the content-height, content-width, and scaling properties. The **user agent** may use any sampling method.

... This is defined as a preference to allow the **user agent** the flexibility to adapt to device limitations and to accommodate over-constrained situations involving min/max dimensions and scale factors.

1.30. 7.14.12 "width"

... The width of a replaced element's box is intrinsic and may be scaled by the **user agent** if the value of this property is different than 'auto'.

1.31. 7.15.4 "line-height"

1.31.1. normal

Tells **user agents** to set the computed value to a "reasonable" value based on the font size of the element.

... When an element contains text that is rendered in more than one font, **user agents** should determine the "line-height" value according to the largest font size.

1.32. 7.15.9 "text-align"

... The actual justification algorithm used is **user agent** and written language dependent. Conforming **user agents** may interpret the value 'justify' as 'left' or 'right', depending on whether the element's default writing direction is left-to-right or right-to-left, respectively.

1.33. 7.15.11 "text-indent"

... **User agents** should render this indentation as blank space.

1.34. 7.16.2 "letter-spacing"

1.34.1. normal

The spacing is the normal spacing for the current font. This value allows the **user agent** to alter the space between characters in order to justify text.

1.34.2. <length>

This value indicates inter-character space in addition to the default space between characters. Values may be negative, but there may be implementation-specific limits. **User agents** may not further increase or decrease the inter-character space in order to justify text.

Character-spacing algorithms are **user agent** dependent. Character spacing may also be influenced by justification (see the "text-align" property).

When the resultant space between two characters is not the same as the default space, **user agents** should not use ligatures.

Conforming **user agents** may consider the value of the 'letter-spacing' property to be 'normal'.

1.34.3. XSL modifications to the CSS definition:

... For "normal": .optimum = "the normal spacing for the current font" / 2, .maximum = auto, .minimum = auto, .precedence = force, and .conditionality = discard. A value of auto for a component implies that the limits are **User Agent** specific.

... The CSS statement that "Conforming **user agents** may consider the value of the 'letter-spacing' property to be 'normal'." does not apply in XSL, if the **User Agent** implements the "Extended" property set.

... The algorithm for resolving the adjusted values between word spacing and letter spacing is **User Agent** dependent.

1.35. 7.16.4 "text-decoration"

... If the element has no content or no text content (e.g., the IMG element in HTML), **user agents** must ignore this property.

1.35.1. blink

... Conforming **user agents** are not required to support this value.

1.36. 7.16.6 "text-transform"

... Conforming **user agents** may consider the value of "text-transform" to be "none" for characters that are not from the ISO Latin-1 repertoire and for elements in languages for which the transformation is different from that specified by the case-conversion tables of Unicode or ISO 10646.

1.37. 7.16.8 "word-spacing"

... Word spacing algorithms are **user agent**-dependent.

1.37.1. XSL modifications to the CSS definition:

... The algorithm for resolving the adjusted values between word spacing and letter spacing is **User Agent** dependent.

1.38. 7.17.1 "color"

Initial: depends on **user agent**

1.39. 7.17.3 "rendering-intent"

1.39.1. auto

This is the default behavior. The **User Agent** determines the best intent based on the content type. For image content containing an embedded profile, it shall be assumed that the intent specified within the profile is the desired intent. Otherwise, the **user agent** shall use the current profile and force the intent, overriding any intent that might be stored in the profile itself.

1.40. 7.20.2 "overflow"

1.40.1. scroll

This value indicates that the content is clipped and that if the **user agent** uses a scrolling mechanism that is visible on the screen (such as a scroll bar or a panner), that mechanism should be displayed for a box whether or not any of its content is clipped.

1.40.2. auto

The behavior of the "auto" value is **user agent** dependent, but should cause a scrolling mechanism to be provided for overflowing boxes.

1.41. 7.21.2 "leader-pattern"

1.41.1. dots

... The choice of dot character is dependent on the **user agent**.

1.42. 7.21.4 "leader-length"

... **User agents** may choose to use the value of "leader-length.optimum" to determine where to break the line, then use the minimum and maximum values during line justification.

1.43. 7.25.11 "media-usage"

1.43.1. auto

The **User Agent** determines which value of "media-usage" (other than the "auto" value) is used. The **User Agent** may consider the type of media on which the presentation is to be placed in making this determination.

NOTE:

For example, the **User Agent** could use the following decision process. If the media is not

continuous and is of fixed bounded size, then the "paginate" (described below) is used. Otherwise, the "bounded-in-one-dimension" is used.

1.43.2. bounded-in-one-dimension

... It is an error if more or less than one of "page-height" or "page-width" is specified on the first page master that is used. The **User Agent** may recover as follows:...

1.43.3. unbounded

Only one page is generated per fo:page-sequence descendant from the fo:root. Neither "page-height" nor "page-width" may be specified on any page master that is used. If a value is specified for either property, it is an error and a **User Agent** may recover by ignoring the specified value. ...

1.44. 7.25.13 "page-height"

1.44.1. auto

The "page-height" shall be determined, in the case of continuous media, from the size of the **User Agent** window...

1.44.2. NOTE:

A **User Agent** may provide a way to declare the media for which formatting is to be done. This may be different from the media on which the formatted result is viewed. For example, a browser **User Agent** may be used to preview pages that are formatted for sheet media. In that case, the size calculation is based on the media for which formatting is done rather than the media being currently used.

1.45. 7.25.15 "page-width"

1.45.1. auto

The "page-width" shall be determined, in the case of continuous media, from the size of the **User Agent** window...

1.46. 7.26.5 "border-separation"

1.46.1. <length-bp-ip-direction>

User agent refs

... Rows, columns, row groups, and column groups cannot have borders (i.e., **user agents** must ignore the border properties for those elements).

1.47. 7.26.7 "caption-side"

... For a caption that is on the left or right side of a table box, on the other hand, a value other than "auto" for "width" sets the width explicitly, but "auto" tells the **user agent** to chose a "reasonable width".

1.48. 7.27.2 "glyph-orientation-horizontal"

1.48.1. <angle>

... The **User Agent** shall round the value of the angle to the closest of the permitted values.

1.49. 7.27.3 "glyph-orientation-vertical"

1.49.1. auto

... The determination of which characters should be auto-rotated may vary across **User Agents**.

1.49.2. <angle>

... The **User Agent** shall round the value of the angle to the closest of the permitted values.

1.50. 7.27.6 "unicode-bidi"

1.50.1. XSL modifications to the CSS definition:

... Fallback:

If it is not possible to present the characters in the correct order, then the **UserAgent** should display either a 'missing character' glyph or display some indication that the content cannot be correctly rendered.

1.51. 7.28.1 "content-type"

... This property specifies the content-type and may be used by a **User Agent** to select a rendering processor for the object.

1.51.1. auto

No identification of the content-type. The **User Agent** may determine it by "sniffing" or by other means.

1.52. 7.29.5 "border-color"

... If an element's border color is not specified with a "border" property, **user agents** must use the value of the element's "color" property as the computed value for the border color.

1.53. 7.29.9 "border-spacing"

... Rows, columns, row groups, and column groups cannot have borders (i.e., **user agents** must ignore the border properties for those elements).

1.54. 7.29.13 "font"

... If no font with the indicated characteristics exists on a given platform, the **user agent** should either intelligently substitute (e.g., a smaller version of the "caption" font might be used for the "small-caption" font), or substitute a **user agent** default font.

1.55. 7.29.19 "pause"

Initial: depends on **user agent**

1.56. 7.29.21 "size"

... Relative page boxes allow **user agents** to scale a document and make optimal use of the target size.

... **User agents** may allow users to control the transfer of the page box to the sheet (e.g., rotating an absolute page box that's being printed).

- Rendering page boxes that do not fit a target sheet
If a page box does not fit the target sheet dimensions, the **user agent** may choose to:
 - Rotate the page box 90 degrees if this will make the page box fit.
 - Scale the page to fit the target.The **user agent** should consult the user before performing these operations.
- Positioning the page box on the sheet
When the page box is smaller than the target size, the **user agent** is free to place the page box anywhere on the sheet.

1.57. 7.29.23 "white-space"

User agent refs

1.57.1. normal

This value directs **user agents** to collapse sequences of whitespace, and break lines as necessary to fill line boxes. ...

1.57.2. pre

This value prevents **user agents** from collapsing sequences of whitespace. ...

... Conforming **user agents** may ignore the 'white-space' property in author and user style sheets but must specify a value for it in the default style sheet.