# XSL Formatting Objects in XEP 3.0

## Abstract

This document describes the implementation of XSL Formatting Objects in XEP — an XSL Engine for PDF developed by RenderX, Inc, version 3.0. It lists all supported formatting objects and their properties, provides information about fallbacks for unsupported objects, and discusses details of XSL spec interpretation adopted in the engine.

## Table of Contents

# 1. XSL FO Support Summary

XEP 3.0 implements *Extensible Stylesheet Language version 1.0* as specified in *XSL 1.0 Recommendation of October 15, 2001*.

## 1.1. Formatting objects supported by XEP 3.0

| § | Object Name | Implemented |
|---|---|---|
| 6.4.2 | **‹fo:root›** | **Yes** |
| 6.4.3 | **‹fo:declarations›** | No |
| 6.4.4 | **‹fo:color-profile›** | No |
| 6.4.5 | **‹fo:page-sequence›** | **Yes** |
| 6.4.6 | **‹fo:layout-master-set›** | **Yes** |
| 6.4.7 | **‹fo:page-sequence-master›** | **Yes** |
| 6.4.8 | **‹fo:single-page-master-reference›** | **Yes** |
| 6.4.9 | **‹fo:repeatable-page-master-reference›** | **Yes** |
| 6.4.10 | **‹fo:repeatable-page-master-alternatives›** | **Yes** |
| 6.4.11 | **‹fo:conditional-page-master-reference›** | **Yes** |
| 6.4.12 | **‹fo:simple-page-master›** | **Yes** |
| 6.4.13 | **‹fo:region-body›** | **Yes** |
| 6.4.14 | **‹fo:region-before›** | **Yes** |
| 6.4.15 | **‹fo:region-after›** | **Yes** |
| 6.4.16 | **‹fo:region-start›** | **Yes** |
| 6.4.17 | **‹fo:region-end›** | **Yes** |
| 6.4.18 | **‹fo:flow›** | **Yes** |
| 6.4.19 | **‹fo:static-content›** | **Yes** |
| 6.4.20 | **‹fo:title›** | No |
| 6.5.2 | **‹fo:block›** | **Yes** |
| 6.5.3 | **‹fo:block-container›** | **Yes** |
| 6.6.2 | **‹fo:bidi-override›** | **Yes** |
| 6.6.3 | **‹fo:character›** | **Yes** |
| 6.6.4 | **‹fo:initial-property-set›** | **Yes** |
| 6.6.5 | **‹fo:external-graphic›** | No |

| § | Object Name | Implemented |
|---|---|---|
| 6.6.6 | **‹fo:instream-foreign-object›** | **Yes**[1] |
| 6.6.7 | **‹fo:inline›** | **Yes** |
| 6.6.8 | **‹fo:inline-container›** | No[2] |
| 6.6.9 | **‹fo:leader›** | **Yes**[3] |
| 6.6.10 | **‹fo:page-number›** | **Yes** |
| 6.6.11 | **‹fo:page-number-citation›** | **Yes** |
| 6.7.2 | **‹fo:table-and-caption›** | **Yes** |
| 6.7.3 | **‹fo:table›** | **Yes**[4] |
| 6.7.4 | **‹fo:table-column›** | **Yes** |
| 6.7.5 | **‹fo:table-caption›** | **Yes** |
| 6.7.6 | **‹fo:table-header›** | **Yes**[5] |
| 6.7.7 | **‹fo:table-footer›** | **Yes**[6] |
| 6.7.8 | **‹fo:table-body›** | **Yes** |
| 6.7.9 | **‹fo:table-row›** | **Yes** |
| 6.7.10 | **‹fo:table-cell›** | **Yes** |
| 6.8.2 | **‹fo:list-block›** | **Yes** |
| 6.8.3 | **‹fo:list-item›** | **Yes** |
| 6.8.4 | **‹fo:list-item-body›** | **Yes** |
| 6.8.5 | **‹fo:list-item-label›** | **Yes** |
| 6.9.2 | **‹fo:basic-link›** | **Yes** |
| 6.9.3 | **‹fo:multi-switch›** | **-** |

[1] Repertory of elements supported inside **‹fo:instream-foreign-object›** depends on availability of additional modules. Core XEP 3.0 distribution does not include any handler for **‹fo:instream-foreign-object›**.

[2] All contents is placed inline.

[3] In this version, only plain text can be put inside leaders with **leader-pattern="use-content"**.

[4] Table support is incomplete; see notes below.

[5] Repeatable table headers work reliably only if column breaks are disabled within table cells; see comments below.

[6] Repeatable table footers are not implemented: the footer is drawn once at the end of the table.

| § | Object Name | Implemented |
|---|---|---|
| 6.9.4 | **‹fo:multi-case›** | - |
| 6.9.5 | **‹fo:multi-toggle›** | - |
| 6.9.6 | **‹fo:multi-properties›** | - |
| 6.9.7 | **‹fo:multi-property-set›** | - |
| 6.10.2 | **‹fo:float›** | **Yes**[7] |
| 6.10.3 | **‹fo:footnote›** | **Yes**[8] |
| 6.10.4 | **‹fo:footnote-body›** | **Yes** |
| 6.11.2 | **‹fo:wrapper›** | **Yes** |
| 6.11.3 | **‹fo:marker›** | **Yes**[9] |
| 6.11.4 | **‹fo:retrieve-marker›** | **Yes** |

## 1.2. Formatting properties supported by XEP 3.0

| § | Property Name | Implemented |
|---|---|---|
| 7.4.1 | **source-document** | No |
| 7.4.2 | **role** | No |
| 7.5.1 | **absolute-position** | **Yes**[10] |
| 7.5.2 | **top** | **Yes** |
| 7.5.3 | **right** | **Yes** |
| 7.5.4 | **bottom** | **Yes** |
| 7.5.5 | **left** | **Yes** |
| 7.6.1 | **azimuth** | - |
| 7.6.2 | **cue-after** | - |
| 7.6.3 | **cue-before** | - |
| 7.6.4 | **elevation** | - |

7 Top-floats (**float="before"**) area is drawn on top of the *following* page.

8 In a multi-column layout, footnotes are placed at the bottom of each column; see discussion below.

9 In the current version, markers cannot be specified as children of **‹fo:wrapper›**.

10 **absolute-position="fixed"** works on **‹fo:block-container›** only.

| § | Property Name | Implemented |
|---|---|---|
| 7.6.5 | **pause-after** | - |
| 7.6.6 | **pause-before** | - |
| 7.6.7 | **pitch** | - |
| 7.6.8 | **pitch-range** | - |
| 7.6.9 | **play-during** | - |
| 7.6.10 | **richness** | - |
| 7.6.11 | **speak** | - |
| 7.6.12 | **speak-header** | - |
| 7.6.13 | **speak-numeral** | - |
| 7.6.14 | **speak-punctuation** | - |
| 7.6.15 | **speech-rate** | - |
| 7.6.16 | **stress** | - |
| 7.6.17 | **voice-family** | - |
| 7.6.18 | **volume** | - |
| 7.7.1 | **background-attachment** | **Yes** |
| 7.7.2 | **background-color** | **Yes** |
| 7.7.3 | **background-image** | **Yes** |
| 7.7.4 | **background-repeat** | **Yes** |
| 7.7.5 | **background-position-horizontal** | **Yes**[11] |
| 7.7.6 | **background-position-vertical** | **Yes**[11] |
| 7.7.7 | **border-before-color** | **Yes** |
| 7.7.8 | **border-before-style** | **Yes** |
| 7.7.9 | **border-before-width** | **Yes** |
| 7.7.10 | **border-after-color** | **Yes** |
| 7.7.11 | **border-after-style** | **Yes** |
| 7.7.12 | **border-after-width** | **Yes** |
| 7.7.13 | **border-start-color** | **Yes** |

[11] When the background image is repeated along an axis, its offset on this axis is ignored.

| § | Property Name | Implemented |
|---|---|---|
| 7.7.14 | **border-start-style** | **Yes** |
| 7.7.15 | **border-start-width** | **Yes** |
| 7.7.16 | **border-end-color** | **Yes** |
| 7.7.17 | **border-end-style** | **Yes** |
| 7.7.18 | **border-end-width** | **Yes** |
| 7.7.19 | **border-top-color** | **Yes** |
| 7.7.20 | **border-top-style** | **Yes** |
| 7.7.21 | **border-top-width** | **Yes** |
| 7.7.22 | **border-bottom-color** | **Yes** |
| 7.7.23 | **border-bottom-style** | **Yes** |
| 7.7.24 | **border-bottom-width** | **Yes** |
| 7.7.25 | **border-left-color** | **Yes** |
| 7.7.26 | **border-left-style** | **Yes** |
| 7.7.27 | **border-left-width** | **Yes** |
| 7.7.28 | **border-right-color** | **Yes** |
| 7.7.29 | **border-right-style** | **Yes** |
| 7.7.30 | **border-right-width** | **Yes** |
| 7.7.31 | **padding-before** | **Yes** |
| 7.7.32 | **padding-after** | **Yes** |
| 7.7.33 | **padding-start** | **Yes** |
| 7.7.34 | **padding-end** | **Yes** |
| 7.7.35 | **padding-top** | **Yes** |
| 7.7.36 | **padding-bottom** | **Yes** |
| 7.7.37 | **padding-left** | **Yes** |
| 7.7.38 | **padding-right** | **Yes** |
| 7.8.2 | **font-family** | **Yes**[12] |

[12] Multiple fonts inside **font-family** are not supported. See note to **font-selection-strategy** property.

| § | Property Name | Implemented |
|---|---|---|
| 7.8.3 | **font-selection-strategy** | No[13] |
| 7.8.4 | **font-size** | **Yes** |
| 7.8.5 | **font-stretch** | **Yes** |
| 7.8.6 | **font-size-adjust** | **Yes** |
| 7.8.7 | **font-style** | **Yes** |
| 7.8.8 | **font-variant** | No |
| 7.8.9 | **font-weight** | **Yes** |
| 7.9.1 | **country** | No |
| 7.9.2 | **language** | **Yes** |
| 7.9.3 | **script** | No |
| 7.9.4 | **hyphenate** | **Yes** |
| 7.9.5 | **hyphenation-character** | **Yes** |
| 7.9.6 | **hyphenation-push-character-count** | **Yes** |
| 7.9.7 | **hyphenation-remain-character-count** | **Yes** |
| 7.10.1 | **margin-top** | **Yes** |
| 7.10.2 | **margin-bottom** | **Yes** |
| 7.10.3 | **margin-left** | **Yes** |
| 7.10.4 | **margin-right** | **Yes** |
| 7.10.5 | **space-before** | **Yes** |
| 7.10.6 | **space-after** | **Yes**[14] |
| 7.10.7 | **start-indent** | **Yes** |
| 7.10.8 | **end-indent** | **Yes** |
| 7.11.1 | **space-end** | **Yes** |
| 7.11.2 | **space-start** | **Yes** |

[13] Font selection algorithm does not look for availability of glyphs in the font. If you specify multiple choices in the **font-family** attribute, the first one that is registered in the system will be selected; if a glyph misses from the font, no attempt will be made to recover by trying alternative families (and the glyph will be replaced by a space). Care should be taken to ensure that all used glyphs are covered by the current font.

[14] **space-after.conditionality="discard"** is not implemented, fallback value is **"retain"**.

| § | Property Name | Implemented |
|---|---|---|
| 7.12.1 | **relative-position** | No |
| 7.13.1 | **alignment-adjust** | **Yes** |
| 7.13.2 | **alignment-baseline** | **Yes** |
| 7.13.3 | **baseline-shift** | **Yes** |
| 7.13.4 | **display-align** | **Yes** |
| 7.13.5 | **dominant-baseline** | **Yes** |
| 7.13.6 | **relative-align** | No |
| 7.14.1 | **block-progression-dimension** | **Yes** |
| 7.14.2 | **content-height** | **Yes** |
| 7.14.3 | **content-width** | **Yes** |
| 7.14.4 | **height** | **Yes**[15] |
| 7.14.5 | **inline-progression-dimension** | **Yes** |
| 7.14.6 | **max-height** | No |
| 7.14.7 | **max-width** | No |
| 7.14.8 | **min-height** | No |
| 7.14.9 | **min-width** | No |
| 7.14.10 | **scaling** | **Yes** |
| 7.14.11 | **scaling-method** | No |
| 7.14.12 | **width** | **Yes**[16] |
| 7.15.1 | **hyphenation-keep** | No |
| 7.15.2 | **hyphenation-ladder-count** | No |
| 7.15.3 | **last-line-end-indent** | **Yes** |
| 7.15.4 | **line-height** | **Yes** |
| 7.15.5 | **line-height-shift-adjustment** | **Yes** |
| 7.15.6 | **line-stacking-strategy** | **Yes** |

[15] Supported only on table rows and block containers. Not supported on images (**‹fo:external-graphic›** and **‹fo:external-foreign-object›**).

[16] Supported only on block containers. Not supported on images (**‹fo:external-graphic›** and **‹fo:external-foreign-object›**).

| § | Property Name | Implemented |
|---|---|---|
| 7.15.7 | **linefeed-treatment** | **Yes**[17] |
| 7.15.8 | **white-space-treatment** | **Yes**[18] |
| 7.15.9 | **text-align** | **Yes**[19] |
| 7.15.10 | **text-align-last** | **Yes** |
| 7.15.11 | **text-indent** | **Yes** |
| 7.15.12 | **white-space-collapse** | **Yes**[20] |
| 7.15.13 | **wrap-option** | **Yes** |
| 7.16.1 | **character** | **Yes** |
| 7.16.2 | **letter-spacing** | **Yes** |
| 7.16.3 | **suppress-at-line-break** | No |
| 7.16.4 | **text-decoration** | **Yes** |
| 7.16.5 | **text-shadow** | No |
| 7.16.6 | **text-transform** | **Yes**[21] |
| 7.16.7 | **treat-as-word-space** | No |
| 7.16.8 | **word-spacing** | **Yes** |
| 7.17.1 | **color** | **Yes** |
| 7.17.2 | **color-profile-name** | No |
| 7.17.3 | **rendering-intent** | No |
| 7.18.1 | **clear** | **Yes** |
| 7.18.2 | **float** | **Yes** |

[17] Value **"treat-as-zero-width-space"** for **linefeed-treatment** is not implemented. This property doesn't work on inlines.

[18] Values **"ignore-if-before-linefeed"**, **"ignore-if-after-linefeed"**, **"ignore-if-surrounding-linefeed"** for **white-space-treatment** are not implemented. Value **"preserve"** behaves like **"ignore-if-surrounding-linefeed"**. This property doesn't work on inlines.

[19] **‹string›** values for **text-align** are not implemented.

[20] This property doesn't work on inlines.

[21] To transform a Unicode character to uppercase/lowercase, XEP uses Java methods **Character.toUpperCase()**/**Character.toLowerCase()**. In order for this property to work as expected, you should use correct Unicode values for glyphs in your fonts, and set up locale information in your Java machine properly.

| § | Property Name | Implemented |
|---|---|---|
| 7.18.3 | **intrusion-displace** | **Yes**[22] |
| 7.19.1 | **break-after** | **Yes** |
| 7.19.2 | **break-before** | **Yes** |
| 7.19.3 | **keep-together** | **Yes**[23] |
| 7.19.4 | **keep-with-next** | **Yes**[23] |
| 7.19.5 | **keep-with-previous** | **Yes**[23] |
| 7.19.6 | **orphans** | **Yes** |
| 7.19.7 | **widows** | **Yes** |
| 7.20.1 | **clip** | No |
| 7.20.2 | **overflow** | No |
| 7.20.3 | **reference-orientation** | **Yes** |
| 7.20.4 | **span** | **Yes** |
| 7.21.1 | **leader-alignment** | No |
| 7.21.2 | **leader-pattern** | **Yes** |
| 7.21.3 | **leader-pattern-width** | **Yes** |
| 7.21.4 | **leader-length** | **Yes** |
| 7.21.5 | **rule-style** | **Yes** |
| 7.21.6 | **rule-thickness** | **Yes** |
| 7.22.1 | **active-state** | **-** |
| 7.22.2 | **auto-restore** | **-** |
| 7.22.3 | **case-name** | **-** |
| 7.22.4 | **case-title** | **-** |
| 7.22.5 | **destination-placement-offset** | **-** |
| 7.22.6 | **external-destination** | **Yes**[24] |

[22] **"indent"** value is not implemented.

[23] **.within-page** component is unsupported; it is mapped to **.within-column**. Only **"auto"** and **"always"** values are recognized properly: numeric values are treated as **"always"**.

[24] All external links are treated as URLs, and open in a browser.

| § | Property Name | Implemented |
|---|---|---|
| 7.22.7 | **indicate-destination** | **-** |
| 7.22.8 | **internal-destination** | **Yes** |
| 7.22.9 | **show-destination** | **-** |
| 7.22.10 | **starting-state** | **-** |
| 7.22.11 | **switch-to** | **-** |
| 7.22.12 | **target-presentation-context** | **-** |
| 7.22.13 | **target-processing-context** | **-** |
| 7.22.14 | **target-stylesheet** | **-** |
| 7.23.1 | **marker-class-name** | **Yes** |
| 7.23.2 | **retrieve-class-name** | **Yes** |
| 7.23.3 | **retrieve-position** | **Yes** |
| 7.23.4 | **retrieve-boundary** | **Yes** |
| 7.24.1 | **format** | **Yes** |
| 7.24.2 | **grouping-separator** | No |
| 7.24.3 | **grouping-size** | No |
| 7.24.4 | **letter-value** | No |
| 7.25.1 | **blank-or-not-blank** | **Yes** |
| 7.25.2 | **column-count** | **Yes** |
| 7.25.3 | **column-gap** | **Yes** |
| 7.25.4 | **extent** | **Yes** |
| 7.25.5 | **flow-name** | **Yes** |
| 7.25.6 | **force-page-count** | **Yes** |
| 7.25.7 | **initial-page-number** | **Yes** |
| 7.25.8 | **master-name** | **Yes** |
| 7.25.9 | **master-reference** | **Yes** |
| 7.25.10 | **maximum-repeats** | **Yes** |
| 7.25.11 | **media-usage** | No |
| 7.25.12 | **odd-or-even** | **Yes** |
| 7.25.13 | **page-height** | **Yes** |

| § | Property Name | Implemented |
|---|---|---|
| 7.25.14 | **page-position** | **Yes** |
| 7.25.15 | **page-width** | **Yes** |
| 7.25.16 | **precedence** | **Yes** |
| 7.25.17 | **region-name** | **Yes** |
| 7.26.1 | **border-after-precedence** | No |
| 7.26.2 | **border-before-precedence** | No |
| 7.26.3 | **border-collapse** | No[25] |
| 7.26.4 | **border-end-precedence** | No |
| 7.26.5 | **border-separation** | **Yes** |
| 7.26.6 | **border-start-precedence** | No |
| 7.26.7 | **caption-side** | **Yes**[26] |
| 7.26.8 | **column-number** | **Yes** |
| 7.26.9 | **column-width** | **Yes** |
| 7.26.10 | **empty-cells** | No[27] |
| 7.26.11 | **ends-row** | **Yes** |
| 7.26.12 | **number-columns-repeated** | **Yes** |
| 7.26.13 | **number-columns-spanned** | **Yes** |
| 7.26.14 | **number-rows-spanned** | **Yes** |
| 7.26.15 | **starts-row** | **Yes** |
| 7.26.16 | **table-layout** | No[28] |
| 7.26.17 | **table-omit-footer-at-break** | No |

25 **border-collapse** has a fixed value of **"separate"**.

26 Only **"before"** and **"after"** values are implemented: **caption-side="start"** falls back to **"before"**, and **caption-side="end"** falls back to **"after"**.

27 In the current implementation, all cells present in the source document are shown regardless of their content being empty; cells not present in the source aren't visible at all.

28 **table-layout** has a fixed value of **"fixed"**.

| § | Property Name | Implemented |
|---|---|---|
| 7.26.18 | **table-omit-header-at-break** | **Yes**[29] |
| 7.27.1 | **direction** | **Yes** |
| 7.27.2 | **glyph-orientation-horizontal** | No |
| 7.27.3 | **glyph-orientation-vertical** | No |
| 7.27.4 | **text-altitude** | **Yes** |
| 7.27.5 | **text-depth** | **Yes** |
| 7.27.6 | **unicode-bidi** | **Yes**[30] |
| 7.27.7 | **writing-mode** | **Yes**[31] |
| 7.28.1 | **content-type** | **Yes** |
| 7.28.2 | **id** | **Yes** |
| 7.28.3 | **provisional-label-separation** | **Yes** |
| 7.28.4 | **provisional-distance-between-starts** | **Yes** |
| 7.28.5 | **ref-id** | **Yes** |
| 7.28.6 | **score-spaces** | No |
| 7.28.7 | **src** | **Yes** |
| 7.28.8 | **visibility** | No |
| 7.28.9 | **z-index** | No |
| 7.29.1 | **background** | **Yes** |
| 7.29.2 | **background-position** | **Yes** |
| 7.29.3 | **border** | **Yes** |
| 7.29.4 | **border-bottom** | **Yes** |
| 7.29.5 | **border-color** | **Yes** |
| 7.29.6 | **border-left** | **Yes** |
| 7.29.7 | **border-right** | **Yes** |

[29] In the current implementation, repeatable table headers can be used reliably only if page breaks are disabled within cells; see comments about **‹fo:table-header›** implementation.

[30] Bidi implementation differs from Unicode Bidi algorithm: any markup element opens a new level of embedding. Consequently, **unicode-bidi="normal"** is not supported (treated as **"embed"**); see detailed discussion below.

[31] Only **"lr-tb"** and **"rl-tb"** values are supported. All other values are treated as **"lr-tb"**.

| § | Property Name | Implemented |
|---|---|---|
| 7.29.8 | **border-style** | **Yes** |
| 7.29.9 | **border-spacing** | **Yes** |
| 7.29.10 | **border-top** | **Yes** |
| 7.29.11 | **border-width** | **Yes** |
| 7.29.12 | **cue** | - |
| 7.29.13 | **font** | **Yes** |
| 7.29.14 | **margin** | **Yes** |
| 7.29.15 | **padding** | **Yes** |
| 7.29.16 | **page-break-after** | **Yes** |
| 7.29.17 | **page-break-before** | **Yes** |
| 7.29.18 | **page-break-inside** | **Yes** |
| 7.29.19 | **pause** | - |
| 7.29.20 | **position** | **Yes** |
| 7.29.21 | **size** | **Yes** |
| 7.29.22 | **vertical-align** | **Yes** |
| 7.29.23 | **white-space** | **Yes** |
| 7.29.24 | **xml:lang** | No |

## 1.3. Notes on Formatting Objects Implementation

**‹fo:bidi-override›**

In the currrent implementation of bidi algorithm, any markup element opens a new level of embedding. Consequently, **unicode-bidi="normal"** is not supported: **‹fo:bidi-override›** behaves as if **unicode-bidi="embed"** were specified.

**‹fo:inline-container›**

Unsupported; contents are placed inline.

**‹fo:multi-switch›**
**‹fo:multi-case›**
**‹fo:multi-toggle›**
**‹fo:multi-properties›**
**‹fo:multi-property-set›**

Unsupported; contents are ignored. These elements deal with interactivity. PDF/PostScript being intrinsically static formats, none of them is applicable.

**‹fo:footnote›**

Footnotes are placed *inside a column* (see notice below about conformance).

**‹fo:footnote-body›** inherits properties from its **‹fo:flow›** ancestor, disregarding all intermediate parents (see notice about conformance).

**‹fo:float›**

Like footnotes, before-floats are placed *inside a column* (see notice below about conformance). The before-float appears on the top of the column *next to the current* one.

**‹fo:float›** inherits properties from its **‹fo:flow›** ancestor, disregarding all intermediate parents (see notice about conformance).

**‹fo:table›**

Table support has the following limitations:

- only fixed table layout is implemented;
- only separate borders model is implemented;
- only table headers can be repeated at page breaks.

**‹fo:table-header›**

Repeatable table header is implemented in a temporary, not conformant style: it is inserted *after all cells that may be pending from the previous column*, and not on the top of the page. This means that it only produces acceptable visual results when all cells have a **keep-together.within-column="always"** attribute.

**‹fo:table-footer›**

Table footer repetition is not implemented. The element contents is repeated once at the end of table.

**‹fo:table-caption›**

Only **"before"** and **"after"** captions are implemented. Side captions are treated as follows: **caption-side="start"** falls back to **"before"**, and **caption-side="end"** falls back to **"after"**.

**‹fo:leader›**

In this version, leaders with **leader-pattern="use-content"** can take only plain text inside; all formatting will be lost.

**‹fo:marker›**

This version cannot process markers specified as children of a **‹fo:wrapper›**.

## 1.4. Supported Expressions

XEP 3.0 implements a subset of XSL expressions algebra. The following operators and functions are recognized:

- Arithmetical operators: **+**, **-**, *****, **div**, **mod**
- **floor()**
- **ceiling()**

- **round()**
- **abs()**
- **max()**
- **min()**
- **rgb()**
- **rgb-icc()** (supported partially — only RGB fallback value is used)
- **from-nearest-specified-value()**
- **from-parent()**
- **from-table-column()**
- **inherited-property-value()**
- **proportional-column-width()**
- **body-start()** (standalone use only, cannot be an operand in expressions)
- **label-end()** (standalone use only, cannot be an operand in expressions)

Support for expressions is subject to the following limitations:

1. For compound expressions, the result of evaluation of all intermediate subexpressions should have a valid XSL type. For example, expression `(2in * 2in) div 1in` is not supported (because its first subexpression yields dimensionality of square inches that is not a valid XSL unit), while `2in * (2in div 1in)` works.
2. Expressions that require knowledge of layout to evaluate (e.g. block widths expressed in percents) can only be used as standalone expressions, not parts of a bigger expression, and cannot be referenced by property-value functions. The same limitation applies also to **body-start()** and **label-end()** functions.
3. Property value functions (**from-nearest-specified-value()**, **from-parent()**, **from-table-column()**, **inherited-property-value()**) cannot be used in shorthands, and cannot take shorthand property names as their arguments.
4. Property value functions that take **start-indent**/**end-indent** as arguments may work incorrectly if the block with indents is placed into another block that has CSS-style **margin-*** attributes. For safety, we recommend using either expressions with indents, or CSS margins; mixing these two coding styles in the same stylesheet may yield inpredictable results.

# 2. Non-Conformance Issues

XEP 3.0 is known to have the following non-conformities to the XSL 1.0 Recommendation:

**Area placement for footnotes and before-floats with multiple columns**

The conditional area for footnotes is subtracted from the column area, rather than from the page area (as prescribed by the spec). This influences the layout when multiple columns are present:

the spec expects all footnotes to be formatted into a single one-column region at the bottom of the page, whereas XEP distributes them into columns where the respective footnote reference occurs.

**Inherited properties on ‹fo:footnote-body› and ‹fo:float› elements**

In the XSL 1.0 Recommendation, **‹fo:footnote-body›** and **‹fo:float›** obey common inheritance rules. It implies that they get inherited properties from the anchor point — despite being formatted into a separate area. This scheme turned out to be extremely unpractical: footnotes/floats would inherit font attributes from inline elements, **keep-together** constraints from headings, indents from lists etc. To ensure that footnotes and floats look uniformly in the XSL Recommendation model, a stylesheet writer would have to care to specify an explicit value for virtually every inhertable property on each **‹fo:footnote-body›**/**‹fo:float›**.

In this situation, we could not help sacrifying conformance to usability. In XEP 3.0 as in previous versions of XEP, out-of-line elements inherit properties from their ancestor **‹fo:flow›**, thus introducing a kind of “region-to-region” inheritance — from body-region to its conditional subregions.

**Inherited start-indent and end-indent on reference areas**

Another situation where holistic inheritance scheme adopted in XSL Recommendation produces indesirable side effects. According to the spec, indents are inherited from **‹fo:table›** to its descendants, i.e. table cells. But inside cells, indents are measured from another reference edge! Apparently, this has the following effect: when you specify an indent for the table, the contents of each cell is shifted by the same amount from the edge of the cell.

XEP breaks inheritance at this point: if an object introduces a new reference area, **start-indent** and **end-indent** for it are not inherited from its parent. In particular, this applies to **‹fo:table-cell›**, **‹fo:block-container›**, and **‹fo:inline-container›**.

**Border and padding on regions**

In the XSL Recommendation, border and padding properties are permitted on region elements (**‹fo:region-body›**, **‹fo:region-before›**, **‹fo:region-after›**, **‹fo:region-start›**, and **‹fo:region-end›**); but the spec says they only may accept values of 0 *(sic!)*. In XEP, we dare give a natural interpretation to non-zero values of these properties — draw a border around the respective region area, and pad its content rectangle by the specified amount. No warning is issued.

**Treatment of empty blocks**

By the spec, an empty block that has a non-null padding and/or border should be visible. XEP suppresses all blocks that have no visible contents regardless of their border or padding attributes.

**Table layout and border model**

Current version implements only the separate border model (**border-collapse="separate"**) and fixed table layout (**table-layout="fixed"**). However, the default values for these properties are different (**border-collapse="collapse"**, **table-layout="automatic"**). To ensure future portability, it is recommended to put an explicit value for these properties on each table; otherwise your documents may behave differently in future XSL FO implementations with support for automatic layout and/or collapsing borders.

# 3. Extensions to the XSL 1.0 Recommendation

XEP implements several extensions to the Specification, placed into a separate namespace: **xmlns:rx="http://www.renderx.com/XSL/Extensions"**. They add support for useful functionality that cannot be expressed by XSL Formatting Objects.

## 3.1. Document Information

This extension permits passing a set of name/value pairs to the generator of the output format. A typical application is setting PDF document info fields (‘Author’ and ‘Title’). Implementation uses two extension elements: **‹rx:meta-info›** and **‹rx:meta-field›**.

**‹rx:meta-info›**

This element is merely a container for one or more **‹rx:meta-field›** elements. It should be the first child of **‹fo:root›**.

**‹rx:meta-field›**

This element specifies a single name/value pair. It has two mandatory attributes: **name** and **value**. Current implementation of the PDF and PostScript generators recognizes four possible values for **name**:

- **name="author"** fills the ‘Author’ field in the resulting PDF file with a string specified by the **value** property;
- **name="title"** fills the ‘Title’ field;
- **name="subject"** fills the ‘Subject’ field;
- **name="keywords"** fills the ‘Keywords’ field.

All other values for **name** are ignored. The ‘Creator’ field in the PDF file is set to `"XEP 3.0"`; there is no means to control it from the source file.

In the PostScript generator module, the document info fields are added using **pdfmark** operator; the respective fields will be filled when PostScript is converted to PDF using *Adobe Acrobat Distiller* or *GhostScript*.

## 3.2. Document Outline (Bookmarks)

An often requested feature for PDF rendering component. Implementation uses three extension elements: **‹rx:outline›**, **‹rx:bookmark›**, and **‹rx:bookmark-label›**.

**‹rx:outline›**

Top-level element of the document outline tree. Should be located before any **‹fo:page-sequence›** elements, but after **‹fo:layout-master-set›** and **‹fo:declarations›** (if present). Contains one or more **‹rx:bookmark›** elements.

**‹rx:bookmark›**

This element contains information about a single bookmark. It contains a mandatory **‹rx:bookmark-label›** element as its first child, and zero or more nested **‹rx:bookmark›** elements that describe subordinated bookmarks. Bookmark destination is expressed either by **internal-destination** property (for internal navigation), or by **external-destination** (for extra-document links).

**‹rx:bookmark-label›**

This element contains text of a bookmark label; it must be the first child of its parent **‹fo:bookmark›**. Contents of this element is plain text.

## 3.3. Indexes

Building page number lists for indexes is not possible within XSL 1.0. XEP 3.0 provides this functionality via extension elements/properties.

**rx:key**

This attribute can be specified on any element that can carry an **id** (and thus be a target to **‹fo:page-number-citation›**). Its content is a term used to group element references in an index entry. Unlike **id**, **rx:key** need not be unique across the document.

**‹rx:page-index›**

This element creates a list of page numbers for an index entry. It groups page numbers for all formatting objects with a given **rx:key**, removing duplicates and eventually merging consequent pages into sequences. The element accepts the following properties:

**ref-key (required)**

Selects elements for the index entry. All elements whose **rx:key** equals the value of this attribute, and only those, will be selected for processing.

**list-separator**

String used to separate page numbers in the list. Default is comma plus space: `", "`.

**range-separator**

String used to separate page numbers that form a continuous range. Default is en dash: **"–"** (`U+2013`).

**merge-subsequent-page-numbers**

Controls whether sequences of adjacent page numbers should be merged into ranges. Default is **"true"**.

☞ In XEP 3.0, **‹rx:index›** only searches elements preceding it in the document. No forward references are implemented.

## 3.4. Flow sections

Flow sections are a generalization of blocks with **span="all"**. They permit to split the flow into subflows with different column counts in each subflow. The following element does it:

**‹rx:flow-section›**

This element must be a direct child of **‹fo:flow›**; it can be mixed with other block-level elements. It explicitly creates a *span-reference-area* with **column-count** and **column-gap** traits taken from the respective properties of **‹rx:flow-section›**.

## 3.5. Background Image Scaling

In XSL 1.0, there is no provision to scale/size a background image. XEP 3.0 implements this functionality via extension properties.

**rx:background-content-height**
**rx:background-content-width**
**rx:background-scaling**

These properties have exactly the same semantics as **content-height**, **content-width**, and **scaling**, respectively. They apply to the image specified in **background-image** property (or inside **background** shorthand).

# 4. Graphic Formats

## 4.1. Bitmap Graphics

XEP 3.0 supports the following graphics formats:

**GIF**

GIF support is limited to non-interlaced images only. Moreover, if the image palette contains less than 256 colors (2, 4, or 16), there is a limit on the maximum image size (approximately 5,200 bytes; the exact figure depends on the palette size).

GIF transparency is supported in PDF output.

**JPEG**

Grayscale, RGB, and CMYK JPEGs are supported. Data stream is copied directly from the image file to the resultant PDF or PostScript, so there's no additional loss of quality.

**PNG**

XEP 3.0 recognizes all types of PNG images described in the PNG specification, and reproduces them with the following limitations:

- Alpha channel is completely ignored — sample values are not adjusted by the alpha.
- 16-bit component colors are trimmed down to 8-bit.

Single-color transparency is supported in PDF output only. For indexed-color images with alpha, the first completely transparent color in the palette is used.

☞ Combining single-color transparency with 16-bit color is not safe in XEP 3.0 because of color depth reduction and consequent merging of adjacent colors.

If the image has an explicit gamma, it is corrected to the sRGB value of 2.2.

**TIFF**

XEP supports the following principal TIFF flavors:

- Strip-based TIFFs: uncompressed/PackBits/LZW, grayscale/RGB/CMYK;
- Bilevel images with CCITT compression;

- Tiled TIFFs: grayscale/RGB (no CMYK), uncompressed/PackBits (no LZW).

TIFF images with separate color planes (`PlanarConfiguration=2`) and/or associated alpha channel (`ExtraSamples=1`) are not supported.

☞ In XEP 3.0, processing of tiled TIFFs and bilevel images with CCITT compression requires Java 2 AWT (JDK/JRE 1.2 or higher). On Java 1.1.8 and MS JVM, XEP only supports basic TIFF flavors: strip-based, uncompressed/PackBits/LZW, grayscale/RGB/CMYK.

When a bitmap graphic has no built-in resolution or dimension data (this is always the case for GIF images, but may also occur in other image types), its resolution defaults to 120 dpi (5 dots of a 600-dpi printer) as prescribed by the CSS2 Spec. This differs from XSL specification that suggests, though not mandates, using 0.28 mm as a pixel size. In our opinion, a smaller pixel size gives better print results: the proportion between pixel size and page width is similar to that of a computer screen. With lower resolutions, it often happens that large GIF/JPEG images fit onto a screen but not into the printable area on the page.

### 4.2. Vector Graphics

In XEP 3.0 the only supported vector graphics format is *EPS*. EPS images are supported *in PostScript generator only*. In the PDF generation module, they are replaced by a bitmap preview image (EPSI or TIFF) if available; otherwise, the correspondent area is left blank.

# 5. Fonts and Internationalization

## 5.1. Font Formats

XEP 3.0 can use the following types of fonts:

- Adobe standard fonts (including OpenType fonts from CJK font packs);
- PostScript Type 1 fonts;
- TrueType fonts (for PDF output only).

All fonts except for OpenType CJK can be embedded into the output file, or specified as external fonts; in the latter case, the resulting file will only be viewable on systems that have the correspondent font configured for use with viewing/printing application. Typically, all fonts are embedded except for standard Adobe PDF fonts; for some applications, embedding basic fonts may also be required.

An embedded font can be *subsetted*: instead of storing the entire font in the document, it is possible to leave only those glyphs that are actually used in the text. This option reduces the document size but makes it unsuitable for subsequent editing. Subsetting is implemented for both TrueType and PostScript fonts.

☞ Some fonts may contain internal flags that prohibit their embedding or subsetting. XEP honors these flags, and may refuse to embed or subset your font if the respective action is not authorized by the flags inside it.

### 5.1.1. Standard Adobe Fonts

#### 5.1.1.1. Latin Adobe Fonts

XEP supports all characters in the Extended Roman character set used in standard Adobe Acrobat fonts — `Times`, `Helvetica`, and `Courier`. This character set is a union of `WinAnsi` (that is in turn a superset of `ISO-8859-1`), `MacRoman`, and `AdobeStandard` character sets. To insert a glyph, common XML rules apply: every encoding recognized by Xerces is usable with XEP.

XEP also supports `Symbol` and `ZapfDingbats` fonts. Symbols from these fonts are also accessed by Unicode values. For Symbol, mapping of Unicode to glyph names is contained in the *Adobe Glyph List* (http://partners.adobe.com/asn/developer/typeforum/glyphlist.txt; hereinafter, *AGL*); for ZapfDingbats, the mapping was taken from a separate document, also available at the Adobe technical support site: http://partners.adobe.com/asn/developer/typeforum/zapfdingbats.txt.

Some glyphs in ZapfDingbats (decorated brackets) have no correspondences in Unicode 3.0; such glyphs are assigned codes in the Adobe corporate-use area. Be careful when using them with PostScript output: many common versions of ZapfDingbats font don't contain these glyphs. For example, you won't be able to view/print them from Aladdin GhostView.

XEP samples include three files where all glyphs available from standard Adobe fonts are listed, with their Adobe glyph names and Unicode values:

- **adobe-standard.pdf** lists all glyphs from Roman Extended character set;
- **symbol.pdf** lists all glyphs from Symbol character set;
- **zapf-dingbats.pdf** lists all glyphs from Zapf Dingbats character set;

#### 5.1.1.2. Standard Fonts for CJK Versions of Acrobat

Asian versions of Adobe Acrobat Reader use additional fonts to display characters for Chinese, Japanese, and Korean. XEP configuration files already contain descriptors for Adobe fonts included into Chinese Simplified, Chinese Traditional, Japanese, and Korean font packs; however, you should make the fonts themselves accessible to the formatter in order to use them with XEP. Please refer to *“XEP 3.0 User Guide”* for instructions.

### 5.1.2. PostScript Type 1 fonts

In version 3.0, there are limitations on the characters that can be included into the PDF or PostScript output:

- for fonts in standard encoding (having **StandardEncoding** or **AdobeStandardEncoding** in the **Encoding** field in the AFM file), accessible characters should belong either to one of the predefined Adobe encodings (WinAnsi or MacRoman), or to the built-in encoding of the font;
- for all other fonts (having **FontSpecific** encoding in the AFM), only characters from the built-in encoding are accessible.

Type 1 font support in XEP is based on direct mapping of Unicode characters to glyph names. Built-in character codes aren't used in the formatting; however, they determine character repertory available for a particular output format.

Mapping of Unicode values to glyph names is described in the Adobe document *"Unicode and Glyph Names"* (see http://partners.adobe.com/asn/developer/typeforum/unicodegn.html). By default, the *Adobe Glyph List* (http://partners.adobe.com/asn/developer/typeforum/glyphlist.txt; hereinafter, *AGL*) is used to determine mapping from Unicode to glyph names; it is hardwired inside XEP. For fonts having non-standard glyph names (e.g. for alphabets not included in the AGL) there is a possibility to specify an explicit code-to-name mapping via an external glyph list.

### 5.1.3. TrueType Fonts

TrueType fonts in XEP 3.0 are supported in the PDF generator only. PostScript generator can only use Type1 fonts.

To be usable by XEP 3.0, a TrueType font must be Unicode-enabled (i.e. have an internal **cmap** table for mapping glyph IDs to Unicode). XEP can access a full range of glyphs from a TrueType font.

## 5.2. Line Breaking Algorithm

XEP uses the following line breaking algorithm:

1. Line break is forced by explicit linefeed characters: `U+000A`, `U+000D`, `U+2028`, `U+2029`, unless they are suppressed by linefeed normalization;
2. Line break is permitted at space characters: `U+0009`, `U+0020`, `U+2000`–`U+200B`, `U+3000`;
3. If **hyphenate** trait is set to **"true"** and all hyphenation conditions (**hyphenation-push-character-count**, **hyphenation-remain-character-count**, etc.) are satisfied, then line break is permitted after a soft hyphen: `U+00AD`. The instance of soft hyphen at the end of line is replaced by text specified in **hyphenation-character** trait; all other instances of `U+00AD` are suppressed.
4. Unless permitted by the above rules, line break is inhibited in the following conditions:
    - before and after non-breaking spaces, hyphens, and joiners: `U+00A0`, `U+200C`, `U+200D`, `U+2011`, `U+202F`;
    - before trailing punctuation characters, closing brackets and quotes, small Katakana and Hiragana characters, superscript characters, etc.;
    - after opening brackets and quotes, Spanish leading punctuations, currency symbols, etc.;
5. Unless prohibited by the above rules, line break is permitted before or after Kanji, Katakana, Hiragana, and Hangul characters;
6. Otherwise, line break is prohibited.

The algorithm is evidently oversimplified; it will be refined in further versions of XEP when more feedback about non-European scripting systems is received.

## 5.3. Bidirectionality

XEP 3.0 provides a limited support for bidirectionality. Implicit ordering of glyphs in an inline area that contains only text (no intervening markup) is governed by a simplified version of Unicode bidi algorithm:

- only one level of embedding is opened — no nested spans;

- positional variant substitution and ligaturization are not supported: all glyphs appear exactly as they were coded in the XSL FO source.

Moreover, bidirectional reordering does not work across markup: any intruding tag splits the sequence into separate pieces that are ordered according to the dominant direction.

XEP 3.0 supports explicit redefinition of writing direction (using **‹fo:bidi-override›** element and its **direction** attribute). Use it wherever possible to avoid dependencies on implicit bidi reordering.

## 5.4. Hyphenation

XEP 3.0 uses Unicode soft hyphen characters (`U+00AD`) to mark possible hyphenation points. These characters can be contained in the source XSL FO document (e.g. come from an external hyphenation software). XEP can also add them inside: it contains a hyphenator class that inserts soft hyphens to all text data before they are passed to the formatter.

The hyphenator implements a simplified version of Liang's algorithm (the same as used in TeX), and reuses TeX hyphenation patterns. Unlike TeX, hyphenation points aren't prioritized: all possible word breaks are considered equivalent.

XEP's distribution includes patterns for the following languages: English (American and British), French, German, Spanish, and Russian. All patterns were borrowed from CTAN (the Comprehensive TeX Archive Network, http://www.ctan.org), with some modifications for non-English patterns. More patterns can be added if necessary; the procedure is described in *“XEP 3.0 User Guide”*.