

Table 15-2 (continued)

<i>Data Type</i>	<i>Definition</i>	<i>Examples</i>
Signed Length	A signed integer or signed real followed by a unit.	5px -0.5in
Unsigned Length	An unsigned integer or unsigned real number followed by a unit.	10px 0.5cm
Positive Length	A positive integer or positive real number followed by a unit.	10px 1pc
Percent	A signed real number that must be divided by 100 to get its actual value.	100.0 -43.2 0.0
Space Specifier	Minimum length semicolon maximum length semicolon optimal length semicolon precedence semicolon conditionality.	0px;72px;12px; force;discard
Limit Specifier	Minimum length semicolon maximum length.	0px;72px
Color	A named color or a hexadecimal triple in the form #RRGGBB.	white #FFFFFF
URI	A Uniform Resource Identifier; in practice, a URL.	http://www.w3 .org/index.html /index.html / ../index.html
Language	An ISO 639 language code.	en la
Font Name	The name of a font, either actual or symbolic.	Times New Roman serif
Font List	Font names separated by commas and possibly whitespace.	Times New Roman, Times, serif
Enumeration	An XML enumeration.	(airplane   train   car   horse)
String	Any sequence of characters.	Fred Lucy and Ethel Castles don't have phones.

## Informational Properties

There are two informational properties, which can be applied to any formatting object. However, neither has a direct affect on the formatting. In essence, these are non-formatting properties.

### The id Property

The first such property is `id`. This is an XML ID-type attribute. The value of this property must, therefore, be an XML name that's unique within the style sheet and within the output formatting object document. The last requirement is a little tricky since it's possible that one template rule in the stylesheet may generate several hundred elements in the output document. The `generate-id()` function of XSL transformations can be useful here.

### The language Property

The second such property is `language`. This specifies the language of the content contained in this element. Generally, the value of this property is an ISO 639 language code such as `en` (English) or `la` (Latin). It may also be the keyword `none` or `use-document`. The latter means to simply use the language of the input as specified by the `xml:lang` attribute. For example, consider the first verse of Caesar's *Gallic Wars*:

```
<fo:block id="verse1.1.1" language="la">
  Gallia est omnis divisa in partes tres,
  quarum unam incolunt Belgae, aliam Aquitani,
  tertiam qui ipsorum lingua Celtae, nostra Galli appellantur
</fo:block>
```

Although the `language` property has no direct effect on formatting, it may have an indirect effect if the formatter selects layout algorithms depending on the language. For instance, the formatter may use different default writing modes for Arabic and English text. This carries over into determination of the start and end regions, and the inline progression direction.

## Paragraph Properties

Paragraph properties are styles that normally are thought of as applying to an entire block of text in a traditional word processor, though perhaps block-level text properties are more appropriate here. For example, indentation is a paragraph property, because you can indent a paragraph but you can't indent a single word separate from its enclosing paragraph.

## Break Properties

The break properties specify where page breaks are and are not allowed. There are five loosely related break properties:

- ♦ `keep-with-next`
- ♦ `keep-with-previous`
- ♦ `break-before`
- ♦ `break-after`
- ♦ `inhibit-line-breaks`

The `keep-with-next` and `keep-with-previous` properties are Booleans that specify whether the formatting object should remain in the same parent-formatting object as the next and previous formatting objects, respectively. This has the effect of keeping two formatting objects on the same page, but it's more strict than that.

The `break-before` property inserts a break before the formatting object starts. Possible things to break include `column`, `page`, `odd-page` and `even-page`. The value may also be `none` or `auto-page`. The `break-after` property inserts a break after the formatting object finishes. The same values are used as for `break-before`. For example, this template rule ensures that each SONNET of sufficiently small size prints on a page of its own:

```
<xsl:template match="SONNET">
  <fo:block break-before="page" break-after="page">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

Finally, the `inhibit-line-breaks` property is a Boolean that can be set to true to indicate that not even a line break is allowed, much less a page break.

## Hyphenation Properties

The hyphenation properties determine whether hyphenation is allowed and how it should be used. This applies only to soft or "optional" hyphens such as the ones sometimes used to break long words at the end of a line. It does not apply to hard hyphens such as the ones in the word *mother-in-law*, though these hyphens may affect where soft hyphens are allowed. There are six hyphenation properties. They are:

- ♦ `hyphenate`: automatic hyphenation is allowed only if this Boolean property has the value `true`
- ♦ `hyphenation-char`: the Unicode character used to hyphenate words, such as `-` in English
- ♦ `hyphenation-keep`: one of the four keywords (`column`, `none`, `page`, `spread`) that specify whether hyphenation is allowed at the end of a facing page pair or column

- ♦ `hyphenation-ladder-count`: an unsigned integer that specifies the maximum number of hyphenated lines that may appear in a row
- ♦ `hyphenation-push-char-count`: an unsigned integer that specifies the minimum number of characters that must follow an automatically inserted hyphen. (Short syllables look bad in isolation.)
- ♦ `hyphenation-remain-char-count`: an unsigned integer specifying the minimum number of characters that must precede an automatically inserted hyphen

Hyphenation also depends on the language and script in use. Thus, the following three properties have particular impact here:

- ♦ `country`
- ♦ `language`
- ♦ `script`

For example:

```
<fo:block hyphenate=true
  hyphenation-char="-"
  hyphenation-keep="none"
  hyphenation-ladder-count="2"
  hyphenation-push-char-count="4"
  hyphenation-remain-char-count="4" >
  some content...
</fo:block>
```

XSL does not specify a syllable-breaking algorithm to determine where a soft hyphen may be applied. Even with these properties allowing hyphenation, it's still completely up to the formatter to figure out how to hyphenate individual words.

### The vertical-align Property

The `vertical-align` property determines the vertical position of a formatting object on its line. It is identical in behavior to the CSS2 property of the same name. There are eight possible keyword values for this property:

1. `baseline`: align the baseline of the box with the baseline of the line box
2. `sub`: align the baseline of the box with the baseline of subscripts inside the line box
3. `super`: raise the baseline of the box to the baseline of superscripts in the line box
4. `top`: align the top of box with the top of the line box

5. `middle`: align the midpoint of the box with the baseline of the line box, plus half the x-height of the line box
6. `bottom`: align the bottom of the box with the bottom of the line box
7. `text-top`: align the top of the box with the top of the font
8. `text-bottom`: align the bottom of the box with the bottom of the font

You can also set `vertical-align` to a signed length that raises or lowers the box by the specified distance from the baseline.

### Indentation Properties

The four indent properties `start-indent`, `end-indent`, `text-indent` and `last-line-end-indent` specify how far lines are indented from the edge of the text. The `start-indent` property offsets all lines from the start edge (left edge in English). The `end-indent` property offsets all lines from the end edge (right edge in English). The `text-indent` property offsets only the first line from the start edge. The `last-line-end-indent` property offsets only the last line from the start edge. Values are given as a signed length. Using a positive value for `start-indent` and a negative value for `text-indent` creates hanging indents. For example, a standard paragraph with 0.5-inch, first-line indent might be formatted this way:

```
<fo:block text-indent="0.5in">  
  The first line of this paragraph is indented  
</fo:block>
```

A block quote with a 1-inch indent on all lines on both sides is formatted like this:

```
<fo:block start-indent="1.0in" end-indent="1.0in">  
  This text is offset one inch from both edges.  
</fo:block>
```

### Character Properties

Character properties describe the qualities of individual characters, although they can apply to elements that contain characters such as `fo:block` and `fo:list-item-body` elements. These include color, font, style, weight, and similar properties.

#### The color Property

The `color` property sets the foreground color of the contents using the same syntax as the CSS `color` property. For example, this colors the text “Lions and tigers and bears, oh my!” pink:

```
<fo:inline-sequence color="#FFCCCC">  
  Lions and tigers and bears, oh my!  
</fo:inline-sequence>
```

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.