

Proposal for XSL-FO 2.0

— Improving compatibility between CSS and XSL-FO —

October 18, 2006
Antenna House, Inc.
www.antennahouse.com
info@antennahouse.com

Antenna House released XSL Formatter in 2001. XSL Formatter is now utilized by over 1000 customers worldwide, many of whom are the leading companies in their respective markets. While supporting these customers along with the many companies evaluating XSL Formatter each month our support group receives numerous inquiries about XSL-FO styleheets from the customers, specialists and the consultants who are involved in the development of the stylesheets. Quite often the stylesheet development problems can be traced back to the fact that the people are confused with the inconsistencies between the way CSS and XSL-FO handles properties. Based on these experiences, Antenna House recognizes that it's a very important subject to enhance the compatibility of two stylesheets, CSS and XSL-FO. We strongly feel that this would lead to increased utilization of both CSS and XSL-FO in the future.

Therefore, we want to report this problem between the compatibility of CSS and XSL-FO, and also make the following proposal for XSL-FO 2.0 specification.

Problem of Compatibility with CSS

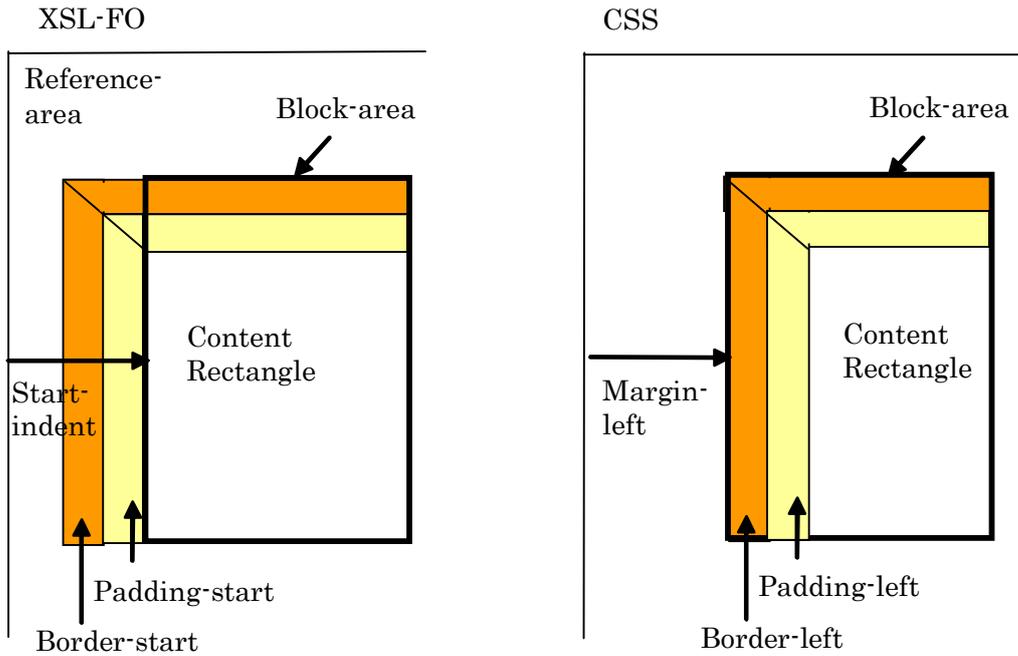
CSS has been in use longer than XSL-FO and the number of people who understand CSS in detail is far greater than those who understand XSL-FO. A majority of the people who develop XSL-FO stylesheet have experience with CSS. Moreover, many of the XSL-FO properties were derived from CSS. If a person who is familiar with CSS is going to develop an XSL-FO stylesheet, it would be thought that the CSS properties work similarly with XSL-FO, and many mistakes would be made. The problem is that the compatibility is imperfect despite many of the XSL-FO properties deriving from CSS.

The examples below show what mistakes the people who know CSS often make and what can be done with CSS but not with XSL-FO, etc.

margin and indent

- When right/left borders and paddings are specified to fo:block, it's expected that the width of the contents of block will become smaller, but in actuality the width of the contents does not change and right/left borders and paddings overflow outside. In CSS, it is because border and padding on either side are taken toward the inside of a block region, on the other hand in XSL-FO border and padding on either side are taken toward the outside of a block region.
- ※ For example, in XSL-FO, indent, border, and padding on the left-hand side of a page have a relationship like the following figure on the left. On the other hand, in CSS they have a relationship like the following figure on the right. The reason for this is because start-indent and end-indent which were not in CSS are introduced in XSL-FO and margin-left and margin-right were taken over for them. In the case of the following figure, margin-left is calculated as follows.

$$\text{margin-left} = \text{start-indent} - \text{padding-start} - \text{border-start-width}$$



- If a left margin is specified for an fo:table, the table will move to the right for that amount and at the same time the contents of the cell in the table additionally moves to the right for the same amount.

<fo:table margin-left="0mm" ...>

世界人権宣言	UDHR
人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、	Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world,
人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人々の最高の願望として宣言されたので、	Whereas disregard and contempt for human rights have resulted in barbarous acts which have outraged the conscience of mankind, and the advent of a world in which human beings shall enjoy freedom of speech and belief and freedom from fear and want has been proclaimed as the highest aspiration of the

<fo:table margin-left="20mm" ...>

世界人権宣言	UDHR
人類社会のすべての構成員の固有の尊厳と平等で譲ることのできない権利とを承認することは、世界における自由、正義及び平和の基礎であるので、	Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world,
人権の無視及び軽侮が、人類の良心を踏みにじった野蛮行為をもたらし、言論及び信仰の自由が受けられ、恐怖及び欠乏のない世界の到来が、一般の人々の最高の願望	Whereas disregard and contempt for human rights have resulted in barbarous acts which have outraged the conscience of mankind, and the advent of a world in which human beings shall

Red arrows indicate a 20mm margin-left on the table and a 20mm start-indent on the text within each cell.

※ Specification of margin-left is converted for start-indent. Since start-indent is an inherited property, it is even inherited to the contents of the cell in the table and the table-cell contents are indented. In order to prevent this, you have to specify start-indent="0pt" to fo:table-cell. There is the same problem also with fo:block-container.

- Contents of the cell in the table in fo:list-item-body shift to the right.

- ※ It's because `start-indent` is specified to `fo:list-item-body` and it is inherited. This is the same case as the previous section.

The auto value for margin

- If `margin-left="auto"` `margin-right="auto"` are specified to `fo:table` and expected that the table will be centered, it would not be effective. The same behavior will happen with `fo:block-container`.
 - ※ In CSS, it's possible to have the position of a block centered with `margin-left: auto;` `margin-right: auto;`, but it's impossible in XSL-FO.

Margin and `<space>.conditionality`

- If `margin-top` or `margin-bottom` is specified to a block, the space will be produced not only at the beginning and end of the block, but also whenever the block is broken by a page break.
 - ※ `margin-top` and `margin-bottom` are converted into `space-before` and `space-after`, then conditionality becomes "retain". Since its behavior is that the space is generated regardless of whether the block area is broken, the space will be generated whenever the block is broken. If `space-before` and `space-after` are specified instead of `margin-top` and `margin-bottom`, `.conditionality="discard"` is default and the space would not be generated whenever the block is broken. But `space-before` of the block that starts the reference area, and `space-after` of the block that ends the reference area will be discarded regardless of whether the block area is broken.
- Inline-level element, `margin-left` is disregarded at the start of the line and `margin-right` is disregarded at the end of the line. Such behavior cannot be found in CSS.
 - ※ It's because inline-level element, `margin-left` and `margin-right` are converted into `space-start` and `space-end`, then `.conditionality="discard"` is applied.

Proposal for improving compatibility with CSS

margin and indent

`start-indent` and `end-indent` were introduced to XSL-FO 1.0 and these were made as inherited properties. That would be the problem. The `margin-*` properties that were primary properties in CSS were provided only for CSS compatibility in XSL-FO 1.0.

Proposal:

It would be fine if make `margin-*` are primary properties as well as in CSS. The `margin-*` properties should be mutually converted into `space-*` and `start/end-indent` properties. The `start/end-indent` should be non-inherited properties.

Define the following properties so that margin can be specified with `writing-mode` relativity.

`margin-before`, `margin-after`, `margin-start`, `margin-end`

The initial value is `Opt`. `start-indent` and `end-indent` are calculated as follows.

If the parent area is a reference area:

start-indent = margin-start + padding-start + border-start-width
end-indent = margin-end + padding-end + border-end-width

If the parent area is a non-reference area:

start-indent = from-parent(start-indent) + margin-start + padding-start + border-start-width
end-indent = from-parent(end-indent) + margin-end + padding-end + border-end-width

When margin-start (or corresponding margin-top/bottom/left/right) and start-indent are both specified, margin-start takes priority. When neither are specified, the initial value, margin-start="0pt" takes priority. The same rule should be applied to margin-end and end-indent.

Thereby, the width of the contents of the block becomes small like in CSS when border or padding are specified to the right/left side of a block even if indent or margin are not specified. Moreover unexpected inheritance of start-indent and end-indent to the block in fo:table-cell will be dissolved.

If start-indent and end-indent are desired to be inherited as same as in XSL 1.x (in order not to change the width of the contents, etc. even if border and padding on either side are specified), specify as follows.

```
start-indent="inherit"  
end-indent="inherit"
```

※ "inherit" is equivalent to "from-parent()". Since it is not an inheritable property any longer "inherited-property-value()" cannot be used.

The auto value for margin

Proposal:

It would be fine if it's possible to center or right align using a value of auto of margin-* as well as in CSS.

Example:

```
<fo:block-container width="10cm" margin-left="auto" margin-right="auto">  
  <fo:block>centered block</fo:block>  
</block-container>  
<fo:block-container width="10cm" margin-left="auto">  
  <fo:block> right -aligned block</fo:block>  
</block-container>
```

margin and <space>.conditionality

In converting from margin-* to space-*, the value for .conditionality in space-before/after in block level is "retain". The value for .conditionality in space-start/end in inline level is "discard". Both are not desirable.

It is a problem that there are only two values, "discard" and "retain" for .conditionality of <space>.

Proposal:

It would be fine if "discard-at-break" is added to the values of .conditionality, then it would be discarded only at the broken portion.

When margin-* is specified and transformed into space-*, the value of .conditionality for all of space-before, space-after, space-start, and space-end should be "discard-at-break". If margin-* is not specified, "discard" is an initial value as well as in XSL-FO 1.x. Thereby the compatibility with XSL-FO 1.x and the compatibility with CSS will go together.

"discard-at-break" should be added to the value of <length-conditional> of border-*-width and padding-* as well, then change the initial value to it. In XSL 1.x, although "discard" was an initial value, the operation of discarding for border and padding is done only at a break portion. By the new definition, "discard" is effective for the start and the end of the reference/line area even if it is not a break portion.

Other Topics

- table-caption
- property values, "left" and "right"
- vertical-align shorthand property
- fo:float
- absolute-position

Reflecting of changes and addition with CSS 2.1

- new values of white-space property
- definition of rect() for <shape>
- initial value of border-collapse property

For the compatibility of CSS3 and XSL-FO 2.0

Although CSS3 is under development, the present draft contains advanced functionalities more than in XSL 1.1. We want to make them available to use in XSL-FO 2.0.

CSS Under Construction:

<http://www.w3.org/Style/CSS/current-work>

CSS3 modules we want to use with XSL-FO:

CSS3 Text: <http://www.w3.org/TR/css3-text/>

CSS3 Text Layout: <http://www.w3.org/Style/CSS/current-work#text>

CSS3 Ruby: <http://www.w3.org/TR/css3-ruby/>

CSS3 Backgrounds and Borders: <http://www.w3.org/TR/css3-background/>

CSS3 Fonts: <http://www.w3.org/TR/css3-fonts/>

CSS3 Box Model: <http://www.w3.org/TR/css3-box/>

CSS3 Multi-columns: <http://www.w3.org/TR/css3-multicol/>

CSS3 Advanced Layout: <http://www.w3.org/TR/css3-layout/>

CSS3 Color: <http://www.w3.org/TR/css3-color>

CSS3 Lists: <http://www.w3.org/TR/css3-lists/>

CSS3 Tables: <http://www.w3.org/Style/CSS/current-work#tables>

CSS3 Positioning: <http://www.w3.org/Style/CSS/current-work#positioning>

CSS3 Line Layout: <http://www.w3.org/TR/css3-linebox/>

etc.