


見本

バッチ組版のための  
XSL-FO 指南

2010年05月第3.0版

 有限会社イー・エイド

 *A Data Usability Company*  
アンテナハウス

アンテナハウス株式会社

「バッチ組版のための XSL-FO 指南」  
藤島 雅宏・編著  
アンテナハウス株式会社・協力  
第 3.0 版 2010 年 05 月 12 日・発行  
発行者：藤島 雅宏  
©有限会社イー・エイド

# はじめに

---

この解説書では、W3C<sup>(1)</sup>の勧告による XSL1.1<sup>(2)</sup>に基づき、アンテナハウス(株)の XSL Formatter V4.x 及び AH Formatter V5.0 に実装されている XSL-FO (XSL Formatting Object) の使い方を解説をします。

AH Formatter V5.0 は、2009 年 3 月に発表された Formatter の最新版で、XSL-FO による組版だけでなく、CSS3 によるページ組版をも実現した画期的なフォーマッターです。この解説書では XSL-FO による組版仕様を解説しております。CSS による組版用スタイルシートについては別の教科書を参照して下さい。

XSL-FO をお使いの皆さんが実務上でご利用いただけるように、XSL-FO の機能をできるだけ網羅して解説するようにしました。しかし、XSL-FO の総括的な解説書ではありませんので、詳しくは XSL-FO の源資料である W3C の XSL1.1 (英文) か、日本工業規格の TR X 0088 を参照して下さい。

XSL Formatter では、XML 文書から XSL スタイルシートを用いて XSL-FO に変換する XSLT プロセッサとして、Windows で標準的に組み込まれている Microsoft 社の MSXML3 または 4 を使っています。

この解説書自身も XML 文書と XSLT によるスタイルシートを使って作られています。しかしながら、ここでは XSL-FO に主眼をおいていますので、説明の例は直接 XSL-FO で記述したものを主体に提示しています。

この解説書は XSL-FO の機能を分かりやすく実例を挙げて解説しています。実務上では XSL-FO を直接書き出すことは少ないでしょうが、XSL-FO を生成する手法については他の教科書に譲って、ここでは説明の都合上理解の手助けになるもののみ XSLT にも言及しています。

アンテナハウス(株)の XSL Formatter V4.x と AH Formatter V5.x とでは、組版機能上の差異があり、一部のプロパティでは使い方が異なったり、V5 から新たに加わった機能もあります。アンテナハウス(株)は今後 AH Formatter V5.x へ移行していきますので、この解説書も最新版の AH Formatter V5.1 に合わせて解説しております。アンテナハウス(株)では日本語組版を充実させるために独自追加機能を提案し、採用しています。これが大変有用ですので、この解説書では積極的に使用しています。これらの拡張機能はアンテナハウス(株)の XSL Formatter 以外では使用できないのですが、この解説書を利用する方々の多くが、他のフォーマッタを使用する必要がないと思いますので、特に意識せず通常の機能の如く扱っています。拡張要素には、名前空間識別子の `axf:` が付されていますので、これに注目していれば、万一他のシステムへ移行するときに峻別することは容易です。

---

(1) W3C(World Wide Web Consortium)は、Web 標準の策定に取組む国際的なコンソーシアム国際学術研究機関です。

(2) XSL(Extensible Stylesheet Language) Version1.1 は、2006 年 12 月 5 日に勧告が出されました。

特に気を付けなければならない点は、AH Formatter V5.x では、独自の追加プロパティが XSL Formatter V4.x と違ったり、プロパティの値が拡張されています。プロパティの拡張分は基準のプロパティとの区別が付きませんので、他社のシステムでは希望の機能が作用しないことを十分ご認識の上運用して下さい。

V5.0 以降で追加された機能を示すために、`V5.0` などと表示を付加しました。

アンテナハウス (株) では、逐次改訂版がリリースされています。最新のニュースと共に、改訂情報をチェックすることをお勧めします。

## 改訂版について

V2.1 版で簡易製本による冊子を作りましたが、実際に自分で使用してみると、A4 サイズでは机上に広げたときにかかなりのスペースをとられます。リファレンスブックとして使用するには少し大きすぎたと感じましたので、B5 サイズに組み直しました。序でにミスプリントや言い足りないところを補完しました。

トンボ出力は、アンテナハウス (株) の追加仕様であり、AH Formatter V5.1 Lite ではサポートされていませんが、有用な機能ですので仕様を掲載しました。

## 謝辞

この解説書のベースは、アンテナハウス (株) の Web ページに紹介されている XSL-FO の例題を、一つに纏め筆者なりに一部手を入れたり、新しい例題を追加したりしました。このような流用を快くご承諾下さり、内容についてアドバイス下さいましたアンテナハウス (株) の小林徳滋社長と技術担当の皆様にご感謝致します。

また、フォントのバリエーションを示すために使わせて頂いた様々なフォントは、(株) ニイスの製品です。サンプル用として快くご提供下さいました (株) ニイスの伊藤晃社長にご感謝致します。

# 目次

<b>1. XSL-FO と XSLT</b> .....	1
XSLT について .....	3
p 要素 .....	3
XSLT による XSL-FO の作り方は割愛しました .....	6
<b>2. XSL-FO の基礎</b> .....	7
用紙と領域 .....	7
用紙と領域の設定 .....	7
領域の位置関係 .....	7
用紙と領域設定の例 .....	8
writing-mode .....	9
簡略設定 .....	11
用紙サイズの簡略設定 .....	11
margin の簡略記法 .....	12
属性値の読み方 .....	12
簡略記法の注意点 .....	13
領域の設定の例 .....	14
writing-mode="tb-rl" の例 (横組み) .....	15
writing-mode="tb-rl" の例 (縦組み) .....	18
<b>3. XSL-FO の構造</b> .....	21
ページマスター .....	21
個々のページマスターの設定 .....	22
ページシーケンス .....	23
ページシーケンスのページ数制御 .....	24
ページシーケンスの繰り返し .....	24
XSL-FO のヘッダー部の実例 .....	26
fo:layout-master-set まで .....	26
解説 .....	27
各ページ毎の組指定 .....	29
表紙の組指定 .....	29
目次の組指定 .....	29
本文と索引の組指定 .....	30
本解説書のスタイルシート (参考) .....	30
出力仕様 .....	31
XSL の構造と外部参照スタイルシート .....	31

4. 背景画像 .....	32
ページに背景画像を指定 .....	32
背景画に対するプロパティとその値 .....	32
ext-background-image の例 .....	33
代替テキスト .....	34
5. 領域の幾何的定義 .....	35
領域の要素 .....	35
行内領域(inline-area) .....	36
正規割付長方形 .....	36
大割付長方形 .....	36
XSL-FO の詳細へ .....	36
6. フォントサイズの設定 .....	37
基本用紙サイズ、基本文字サイズ .....	37
フォントサイズを直接設定する .....	37
段落単位でフォントサイズを設定する .....	37
長さの単位 .....	38
フォントサイズの絶対指定 .....	39
フォントサイズの相対値指定 .....	40
7. フォントストレッチ .....	42
絶対指定 .....	42
相対指定 .....	43
パーセント指定 .....	44
数値指定 .....	44
8. フォントファミリーの設定 .....	46
直接指定 .....	46
共通指定 .....	47
代替えフォントの指定 .....	48
9. 文字の指定 .....	50
文字をコードポイントで指定 .....	50
外字の指定 .....	50
OpenType フォントの字体切り替え .....	51
リガチャ .....	51
代替字形の指定 .....	52
文字列の置換 .....	53
数字列の置換 .....	53
漢数字 .....	54

漢数字の形式指定 .....	54
漢数字文字指定 .....	55
位取り文字指定 .....	56
漢数字への変換の仕方 v5.0 .....	57
<b>10. 文字修飾</b> .....	<b>59</b>
文字の修飾いろいろ .....	59
文字修飾のプロパティ .....	61
文字修飾のプロパティ .....	62
<b>baseline-shift</b> .....	64
<b>11. 約物の処理</b> .....	<b>66</b>
行頭・行末での全角約物の半角扱い .....	66
連続する約物の詰め処理 .....	69
句読点のぶら下げ処理と行頭突き出し処理 .....	72
行頭・行末禁則処理 .....	74
禁則約物の追加および削除 .....	78
追い込み処理 .....	79
<b>axf:avoid-widow-words</b> .....	80
<b>12. 和・欧文間の空き</b> .....	<b>82</b>
和・欧文間の空きに対する拡張 .....	82
<b>axf:text-autospace-width</b> .....	82
<b>13. 文字間の空きの設定</b> .....	<b>86</b>
<b>letter-spacing</b> による調整 .....	86
<b>justify</b> との関係 .....	87
<b>14. 段落内の文字の配置</b> .....	<b>89</b>
<b>text-align</b> の設定 .....	89
<b>15. 行の高さ指定</b> .....	<b>92</b>
絶対値で行の高さを指定。 .....	92
相対値で行の高さを指定。 .....	93
<b>16. 異サイズの文字がある時の行の高さ</b> .....	<b>94</b>
絶対値で行の高さを指定した場合。 .....	94
相対値で行の高さを指定した場合。 .....	95
<b>17. line-stacking-strategy と行の高さ</b> .....	<b>96</b>
<b>line-stacking-strategy</b> による行の高さの変化 .....	96
<b>line-height</b> の纏め .....	99
line-height の値とは .....	99
行内領域との関係 .....	100

line-stacking-strategy とは .....	101
18. 組方向 .....	102
ページ内に横組みと縦組みの混在 .....	102
縦組みの例 .....	103
縦組み中横組み .....	104
fo:inline-container での組方向変更 .....	105
19. 英文組版 .....	106
ハイフネーションなど .....	106
欧文組版のための拡張 .....	107
一般的なハイフン付け特性 .....	109
欧文組版のためのその他のプロパティ .....	111
20. ルビの例 .....	115
擬似的な解法 .....	115
ルビの例さまざま .....	116
縦組みでのルビ .....	117
21. 保持及び分割 .....	119
分割 .....	119
改行・改段・改ページ .....	121
保持 .....	121
分割の禁止 .....	122
22. 境界線 .....	126
境界線の種類と太さ（対角線を含む） .....	126
23. 境界線を引く指定の仕方 .....	128
境界線を引く辺を指定する .....	128
別々に各辺の指定をする .....	128
border 指定の省略形 .....	129
padding を使った場合 .....	130
padding 指定のプロパティ .....	131
margin を使った場合 .....	132
24. ボーダーとパディング .....	133
25. 角丸 .....	138
ボーダーやエリアを角丸にする .....	138
影を付ける .....	139
26. テーブル .....	141
テーブルの構造 .....	141



テーブルの要素 .....	141
テーブルの基本的サンプル .....	142
テーブルにおける注意事項 .....	143
テーブルに用いられる一般のプロパティ .....	144
テーブル独自のプロパティ .....	145
<b>27. テーブルのカラム幅の設定 .....</b>	<b>153</b>
自動モード .....	153
固定モード .....	153
<b>28. カラムスパンとロースパン .....</b>	<b>154</b>
<b>29. テーブルのボーダーとパディング .....</b>	<b>155</b>
自動レイアウトの場合 .....	155
固定レイアウトの場合 .....	156
テーブルセルに角丸 .....	157
<b>30. セル内の文字の配置 .....</b>	<b>158</b>
<b>text-align</b> .....	158
<b>text-align="string"</b> .....	159
text-align="string" の拡張 .....	159
<b>display-align</b> .....	160
<b>relative-align</b> .....	161
<b>31. テーブルのページ分け .....</b>	<b>162</b>
表のヘッダ・フッタの繰り返し .....	162
表のヘッダ・フッタを繰り返す .....	162
ヘッダー・フッターの繰り返しの拒否 .....	163
分断されたテーブルセルの上下の線を生かす .....	164
分断されたテーブルセルの内容を繰り返す .....	165
表頭を回転させる .....	167
その他の例 .....	169
<b>32. レイアウトに変化を付ける .....</b>	<b>170</b>
テーブルの外枠の太さを変える .....	170
テーブルの外枠に角丸を指定する .....	171
テーブルの外枠とヘッダーの一体化 .....	171
テーブルの外枠を消す .....	172
斜線を用いた例 .....	173
テーブルセル内での縦組み指定 .....	174
その他の事例 .....	175

33. テーブルマーカー .....	177
テーブルの続きを表示 .....	177
小計の表示 .....	180
34. テーブルの回転 .....	183
幅広のテーブルを収める .....	183
35. テーブルに対するスタイルシート .....	186
表に使われる要素 .....	186
XML コンテンツの例 .....	187
表を表す XSL-FO の要素 .....	188
スタイルシートの作成 .....	189
プロパティセット .....	189
"table"と"table/title"のテンプレート .....	190
"col"のテンプレート .....	191
"thead"と"tfoot"のテンプレート .....	192
"tbody"のテンプレート —<tr>— .....	192
"tbody"のテンプレート —<th>— .....	193
"tbody"のテンプレート —<td>— .....	194
"cell-span"のテンプレート .....	195
処理の実例 .....	195
出力例 .....	196
36. 箇条書き .....	197
リスト形式のフォーマット .....	197
リスト形式の基本的構造 .....	198
番号なしリストのフォーマット .....	199
番号付きリストのフォーマット .....	200
format プロパティの拡張 .....	203
定義リストのフォーマット .....	205
リストの構造 .....	205
リストブロックをネストする。 .....	206
実例中での箇条書きの例 .....	207
37. リストに対するスタイルシート .....	209
XML コンテンツにおけるリスト形式 .....	209
XSL-FO におけるリスト形式 .....	210
番号付きリストを処理するテンプレート .....	211
リストに対するプロパティセット .....	211
"ol"に対するテンプレート .....	211

"ol/li"に対するテンプレート .....	212
ラベルの書式 .....	213
位置指定のまとめ .....	214
事例 .....	214
出力例 .....	214
<b>番号なしリストを処理するテンプレート .....</b>	<b>215</b>
"ul"のテンプレート .....	215
"ul/li"のテンプレート .....	216
ラベルに画像を用いる .....	217
番号なしリストの例 .....	217
出力例 .....	218
<b>定義リストを処理するテンプレート .....</b>	<b>218</b>
定義リストに対するプロパティセット .....	218
2つの定義型リストの形式 .....	219
"dl.format.list"のテンプレート .....	220
"process.dl.list"のテンプレート .....	221
"dt"のテンプレート .....	222
"dd"のテンプレート .....	222
XSL-FO での定義型リストの例 .....	223
HTML での定義型リスト .....	223
<b>38. リーダー .....</b>	<b>225</b>
<b>fo:leader</b> .....	<b>225</b>
fo:leader の単純な指定 .....	225
leader-pattern の値を"rule"にする .....	226
leader-pattern の値を"use-content"にして、パターンに使う文字を指定する。 .....	226
リーダーを広げる .....	227
use-content に特殊文字を .....	228
1行の中で fo:leader を2回使う .....	229
リーダーのプロパティ .....	230
<b>39. 行外ブロックの処理 .....</b>	<b>232</b>
開始、終了フロートと図形 .....	232
開始、終了フロートと intrusion-displace 指定 .....	234
フロートと clear=inside/outside .....	236
前方、開始、終了フロートと脚注 .....	239
フォローと脚注 .....	240
<b>40. 脚注・傍注 .....</b>	<b>242</b>

脚注と傍注 .....	242
脚注に対するテンプレート .....	243
脚注との分離など .....	245
脚注番号に対する拡張機能 .....	245
段組時の脚注 .....	247
脚注に関する拡張機能 .....	248
脚注の配置 .....	248
<b>41. 段組み</b> .....	<b>252</b>
ページ全体を段組にする .....	252
段組に関するプロパティ .....	252
<b>block-container</b> の中で段組にする .....	253
balance を制御する .....	254
<b>42. 段間罫</b> .....	<b>256</b>
通常の段間罫 .....	256
段間罫の制御 .....	257
最後の段に罫を引かない .....	257
最後の段にも罫を引く .....	258
段の後に段間罫を引く .....	258
罫線の長さ指定 .....	259
段間罫の位置指定 .....	260
段間罫の拡張プロパティ .....	261
<b>43. 改訂バーとチェンジバー</b> .....	<b>263</b>
改訂バー .....	263
チェンジバー .....	264
特性の違い .....	267
<b>44. 行番号</b> .....	<b>270</b>
行番号をつける .....	270
行番号表示の複雑な指定 .....	271
行ナンバーに使われるプロパティ .....	273
<b>45. ブロックの配置</b> .....	<b>277</b>
ブロックの配置 .....	277
<b>46. ブロックコンテナ</b> .....	<b>278</b>
絶対位置指定 (absolute-position="absolute") .....	278
ブロック・コンテナの位置と大きさの指定 .....	279
ブロック・コンテナの位置と大きさの指定 (段組) .....	281
絶対位置指定のブロック・コンテナ .....	282

固定位置指定のブロック・コンテナ .....	283
<b>47. オーバーフロー制御 .....</b>	<b>284</b>
<b>ブロックコンテナオーバーフローの基本処理 .....</b>	<b>284</b>
overflow のプロパティ .....	285
オーバーフローの拡張機能 .....	286
オーバーフローの追加仕様 1 .....	289
オーバーフローの追加指定 2 .....	290
オーバーフローの追加指定 3 .....	291
<b>48. 画像の取り込み .....</b>	<b>292</b>
外部の画像イメージを取り込む .....	292
背景に画像イメージを取り込む .....	293
<b>49. 各種グラフィックの取り込み .....</b>	<b>295</b>
Windows BMP .....	295
GIF: Graphic Interchange Format .....	295
WMF: Windows Metafile .....	296
EMF: Enhanced Metafile .....	297
PNG: Portable Network Graphic .....	298
JPEG .....	299
TIFF の圧縮方法とカラー空間 .....	300
<b>50. 画像倍率の選択 .....</b>	<b>304</b>
高さ倍率のスケールリストが定義されている場合 .....	304
scale-to-fit .....	304
scale-down-to-fit .....	305
scale-up-to-fit .....	306
幅倍率のスケールリストが定義されている場合 .....	307
scale-to-fit .....	307
scale-down-to-fit .....	308
scale-up-to-fit .....	309
高さ倍率のスケールリストが定義されていない場合 .....	310
scale-to-fit .....	310
scale-down-to-fit .....	311
scale-up-to-fit .....	312
幅倍率のスケールリストが定義されていない場合 .....	313
scale-to-fit .....	313
scale-down-to-fit .....	314
scale-up-to-fit .....	315

固有倍率のある場合 .....	316
<b>51. SVG の取り込み .....</b>	<b>317</b>
.....	317
Scalable Vector Graphics .....	319
Instream SVG .....	320
Gradation .....	322
Gray Scale Gradation .....	323
Gray Scale .....	323
<b>52. MathML .....</b>	<b>324</b>
<b>53. PDF の埋め込み .....</b>	<b>325</b>
オンラインマニュアルから .....	325
PDF 埋込の事例 .....	326
<b>54. 目次と索引 .....</b>	<b>327</b>
事例イメージ .....	327
索引のページ .....	332
ノンブルの振り方 .....	333
目次 .....	333
図版目次 .....	333
序文 .....	333
本文 .....	334
本文続き .....	334
用語解説 .....	334
巻末索引 .....	335
ノンブルに総頁も付加する .....	335
目次の作り方 .....	336
索引取りの仕方 .....	337
ページ表示の整理いろいろ .....	338
すべてそのまま表示 .....	338
重複するページはマージ .....	339
連続するページを纏める .....	339
範囲内の図版などはマージ .....	340
図版等は別扱いで、連続ページはマージ .....	340
すべてマージ .....	340
<b>55. ブックマーク（しおり） .....</b>	<b>341</b>
本文での指定 .....	341
ブックマークの設定 .....	342

<b>56. 流し込みマップ</b> .....	344
<b>流し込みマップによる領域指定の柔軟性拡大</b> .....	344
フロー"A"を領域"R"に、フロー("B1"、"B2")を領域"S"に流し込む .....	344
"A"と("B1"、"B2")を1つの領域"R"に流し込む .....	346
1つのフロー"A"を2つの領域"R"と"S"に流し込む .....	348
"A"、("B1"、"B2")を領域"R"、"S"に流し込む .....	349
<b>57. トンボ</b> .....	352
<b>トンボ出力</b> .....	352
クロップオフセット .....	352
塗り足し .....	353
トンボマーク .....	353
<b>58. 備考</b> .....	355
<b>プロパティのデータ型</b> .....	355
<b>索引</b> .....	<b>357</b>



This is a blank page.



# 1. XSL-FO と XSLT

HTML や CSS が Web ブラウザに対して画面上での表示について指定しているのに対して、XSL-FO は主に紙媒体における文書を組版するための仕様です。紙媒体を基本としていますから、紙のサイズ、紙面の中のどの位置にドキュメントを配置するかなどから始めます。ドキュメントのタイトルや段落に対して、書体、サイズ、縦書きか横書きか、など書籍組版に必要な指定が細かに制定されています。更に、目次や索引取りの機能が充実しています。

HTML では当初ドキュメント自身に組版指示を指定してきましたが、データであるコンテンツと組版に必要な指示とが混在してきて分かりにくくなりました。そこでコンテンツと組版指示を分離して、HTML コンテンツがコンピュータによるデータ処理を容易にするように簡素化されました。そして、CSS の規格が組版機能を指定するものとして制定され、その機能は現在でも拡充されています。

HTML の場合は、ブラウザが HTML と CSS とを解釈して直接画面表示やプリンタ出力に必要な変換を一度にしています。従って、HTML によるコンテンツと CSS による組版指示を与えれば、画面表示やプリンタ出力ができますので、プロセスが分かりやすく感じられます。HTML のコンテンツを頭から順に<h>や<p>のタグに応じてそれぞれの指定を CSS から貰って、次々に画面に表示していけば良いので処理手順も簡単です。

XML データから書籍組版をする場合、ドキュメントはページ毎に区切られて、それぞれに柱やノンブルと言った書籍の体裁にするための付属物が付きます。目次や索引取りでは、一旦組み上がった状態を作ってから、該当するページ番号を取り出してこなければなりません。これらの組版処理を楽にさせるために、XML コンテンツと XSL スタイルシートによって、中間ファイルである XSL-FO のデータを作ります。XSL-FO のデータには組版に必要な指示が、テキストの必要個所に逐一挿入されています。XSL-FO による中間ファイルは、組版指示の中にテキストが埋もれているように見えるくらい複雑なデータ型式ですが、この規格をしっかりと勉強しておくことで、XSL-FO に変換するスタイルシート設計が確かなものになります。

このような XSL-FO による中間データを作ることによって、XML コンテンツと組版指示のスタイルシートを分離した大きな効果ができました。それは、XML コンテンツがコンピュータによるデータ処理用に作られていて、書籍の体裁にはほど遠いスタイルであっても、スタイルシートによって書籍に向けたフォーマットに組み直すことが可能になりました。XML コンテンツでは、コンピュータによるデータ処理に向けたタグ付けがされていて、書籍のように章やタイトルなどの体裁が全くなくても構わないのです。

スタイルシートは、XML コンテンツの中から必要なものをデータを取りだして、書籍の形に組むために必要な組版指示を XSL-FO の仕様に従ってそのデータに付加します。例えば XML コンテンツでは表の形に整形されていなくても、スタイルシートによって表の形になるように XSL-FO の表形式のデータに加工すれば、所望の出力物を得ることができます。これは、XML コンテンツと、書籍のための組版指示とを分離した大きな利点です。

XSL-FO のデータは XML 型式で書かれますので、XML コンテンツの XML データから XSL-FO のデータに変換するとき、ツリー構造を変えたり、不要なものを外して、必要なものを付加するなどの加工が容易にできます。そのための規格として XSLT(XSL Transformation)があります。XSLT は当初 XSL の一部でしたが、現在では独立した規格となり、XSL は XSL-FO の規格となっています。XSLT は XML データの取扱に特化されたもので、「XML 文書の構造を変換するための言語」といわれ、その機能は XPath 仕様を取り込んでどんどん拡充されています。XPath 仕様は独立した言語として定義され、ソース文書からデータを選択して目的の文書に配置するなどの機能を備えており、XSLT 以外のところでも活用されています。

XSLT が XML データの構造変換に使えると言うことは、XSLT によって画面表示に向けた HTML の型式に変換することも可能です。作成された HTML を画面表示するための CSS が予めできていれば、画面表示には向かないような構造の XML コンテンツから、XSLT のスタイルシートによって所定の形の HTML に変換すれば、直ちに画面表示ができます。すなわち XSLT は Web ブラウザ用の HTML の形に変形したり、書籍用の XSL-FO による型式に変換することも自由になります。

このように XSL-FO の規格が明確になるに従い、目的がハッキリしてくるとこれに変換するための XSLT が定義され、さらに XSLT の機能を拡充するために XPath が取り入れられるなどと、次々に拡張されているためとても複雑に見えています。XML データを処理するに XSLT によるスタイルシートを書きます。しかし、XSLT は XML データの構造を変換するための規格ですから、XSLT によってどのようなデータを作ればよいかは分からなくては困ります。そのために先ず、XSL-FO の仕様をしっかりと理解しておく必要があります。XSL-FO の仕様は W3C によって設定され公開されています。システムベンダーはこの仕様に従ってフォーマッタを開発します。

XSL-FO で指定された指示に従って、所望するスタイルに組み上げるエンジンが必要です。これを XSL-FO プロセッサと言います。欧文組版を主としたものは幾つかのプロセッサがありますが、日本語組版を解決しているものは、アンテナハウス (株) の XSL Formatter だけです。XSL Formatter は欧文やアラビア語など多言語の組版にも対応しているため、海外でも多く使われています。更に XSL Formatter は和文組版に必要な拡張機能を独自に追加して、日本語組版への機能拡張をしています。これらの拡張機能は日本語組版独自の機能に限らず、欧文組版にも活用できるものが多いので、XSL-FO の規格を制定し管理している W3C に機能拡張を提案しています。

XSLT を処理する XSLT プロセッサは、Windows 版では、特に設定していない場合は MSXML4 または MSXML3 を標準の XSLT プロセッサとして使用しています。別の XSLT プロセッサを使用する場合や、Solaris/Linux/Macintosh/HP-UX/AIX 版では、使用する XSLT プロセッサの設定を行ってください。

なお、XSL-FO の中間データは、実務上ではプロセッサの内部で処理されるだけで、オペレータが介在する必要はありません。アンテナハウスの XSL Formatter に XML と XSL スタイルシートを与えれば、オペレータは中間ファイルの XSL-FO データを意識することなく、所望の紙出力や PDF 出力を得ることができます。

## XSLT について

いきなり XSL-FO の事例を見ますと、とても複雑で大変に見えますが、実務上は XSLT などを用いて XSL-FO を作成しますので、もっと分かりやすい表記になっています。非常に簡単な例を挙げて、XSLT 上での記述の簡潔さを見ます。

本文の XML コンテンツを XSLT で処理する例を見て、XML から XSL-FO に変換する様子を見ておきます。

### p 要素

たとえば XML データの中に次のような p 要素があるとき、例えば次のような形式で書かれます。

#### XML コンテンツの p 要素の例

```
<p>これは段落のサンプルです。段落は fo:block に変換されます。行頭の字下げには、text-indent="1em"と指定されているので、先頭行は一文字分だけ字下げされます。行揃えは text-align="justify"の指定で「両端揃え」としています。ただし段落の最終行は自動的に左揃えとなります。これは text-align="justify"の場合は、最終行に適用される text-indent-last の初期値が自動的に start になってくれるためです。</p>
```

p 要素を処理する XSL スタイルシートの一例を次に示します。書き方は冗長なように見えますが、何を設定しているかがすっきりと分かります

#### p 要素を処理するテンプレート

```
<xsl:attribute-set name="p">
  <xsl:attribute name="font-family">'MS 明朝'</xsl:attribute>
  <xsl:attribute name="font-size">12pt</xsl:attribute>
  <xsl:attribute name="line-height">1.4em</xsl:attribute>
  <xsl:attribute name="text-indent">1em</xsl:attribute>
  <xsl:attribute name="space-before">0.7em</xsl:attribute>
  <xsl:attribute name="space-after">0.7em</xsl:attribute>
  <xsl:attribute name="text-align">justify</xsl:attribute>
```

```
</xsl:attribute-set>

<xsl:template match="p">
  <fo:block xsl:use-attribute-sets="p">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
```

テンプレートは簡単で、アトリビュートセットの設定に従って fo:block に変換するのみです。アトリビュートセットは、使用する書体を'MS 明朝'とし、文字サイズは 12pt、行の高さを 1.4em (em は使用する文字サイズの大きさ)、先頭行のインデントは text-indent で 1em、行揃えの text-align は justify で、段落の前後に 0.7em の空白を付けています。

#### 上記の設定で処理されたときの fo の例

```
<fo:block font-family="MS 明朝" font-size="12pt" line-height="1.4em"
text-indent="1em" space-before="0.7em" space-after="0.7em" text-align="justify">これは段落のサンプルです。段落は fo:block に変換されます。行頭の字下げには、text-indent="1em"と指定されているので、先頭行は一文字分だけ字下げされます。行揃えは text-align="justify"の指定で「両端揃え」としています。ただし段落の最終行は自動的に左揃えとなります。これは text-align="justify"の場合は、最終行に適用される text-indent-last の初期値が自動的に start になってくれるためです。</fo:block>
```

このようにして作られる XSL-FO によってフォーマッタが実際の組版イメージを作ります。

#### 'MS 明朝'、12pt、行の高さ 1.4em とした場合

これは段落のサンプルです。段落は fo:block に変換されます。行頭の字下げには、text-indent="1em"と指定されているので、先頭行は一文字分だけ字下げされます。行揃えは text-align="justify"の指定で「両端揃え」としています。ただし段落の最終行は自動的に左揃えとなります。これは text-align="justify"の場合は、最終行に適用される text-indent-last の初期値が自動的に start になってくれるためです。

アトリビュートセットで、書体を'MS ゴシック'、文字サイズ 10.5pt に変更します。左右のインデントを 3em として追加しました。

#### 書体、文字サイズなどを変更した場合

```
<xsl:attribute-set name="p">
```

```
<xsl:attribute name="font-family">'MS ゴシック'</xsl:attribute>
<xsl:attribute name="font-size">10.5pt</xsl:attribute>
<xsl:attribute name="start-indent">3em</xsl:attribute>
<xsl:attribute name="end-indent">3em</xsl:attribute>
----省略----
</xsl:attribute-set>
```

'MS ゴシック'、10.5pt、左右のインデント 3em とした場合

これは段落のサンプルです。段落は fo:block に変換されます。行頭の字下げには、text-indent="1em" と指定されているので、先頭行は一文字分だけ字下げされます。行揃えは text-align="justify" の指定で「両端揃え」としています。ただし段落の最終行は自動的に左揃えとなります。これは text-align="justify" の場合は、最終行に適用される text-indent-last の初期値が自動的に start になってくれるためです。

アトリビュートセットを設けなくて、テンプレートの中で一々設定しても構いません。

```
<xsl:template match="p">
  <fo:block font-family="MS 明朝" font-size="10.5pt"
    line-height="1.4em" text-indent="1em" space-before="0.7em"
    space-after="0.7em" text-align="justify" />
  <xsl:apply-templates />
</fo:block>
</xsl:template>
```

このように XSLT によるスタイルシートによって同じ XML データがいろいろな書式で組版出来ることがお分かりと思います。

## "XSLT の教科書"

XSLT でスタイルシートを書くには、XSLT の教科書を勉強する必要があります。現在市販されているものや、Web ページ上で入門やチュートリアルとして易しく解説しているものがありますので、それらを参考にして下さい。アンテナハウス (株) のホームページにあるサンプルなどでトレーニングしておくと思います。

## XSLT による XSL-FO の作り方は割愛しました

通常の処理では、XML コンテンツから XSL スタイルシートを用いて XSL-FO を生成するのが一般的です。従って、実務上は XSLT の解説をしないと十分ではありません。また、説明の都合上 XSLT によるスタイルシートに言及した方が理解しやすいと思われるものには、簡単な例を提示しました。XSLT は元々は XSL 規格の一部でしたが、その機能が多義に渡って拡充するに伴い、現在では XSL から独立して XSLT の規格として存在しています。XSLT に付いての詳細な解説はそれだけで大きなものになりますので、この解説では必要なことしか触れていません。XML コンテンツから XSL-FO 形式に変換するには、XSLT 以外に Java などのソフトを用いても可能ですが、これも他の解説書に譲ります。

この解説では XSL-FO の機能を解説することに専念しました。これから XSL-FO の機能を勉強される方への入門書として、XSL-FO を実用されている方々へのレファレンスブックとして、ご活用下さい。

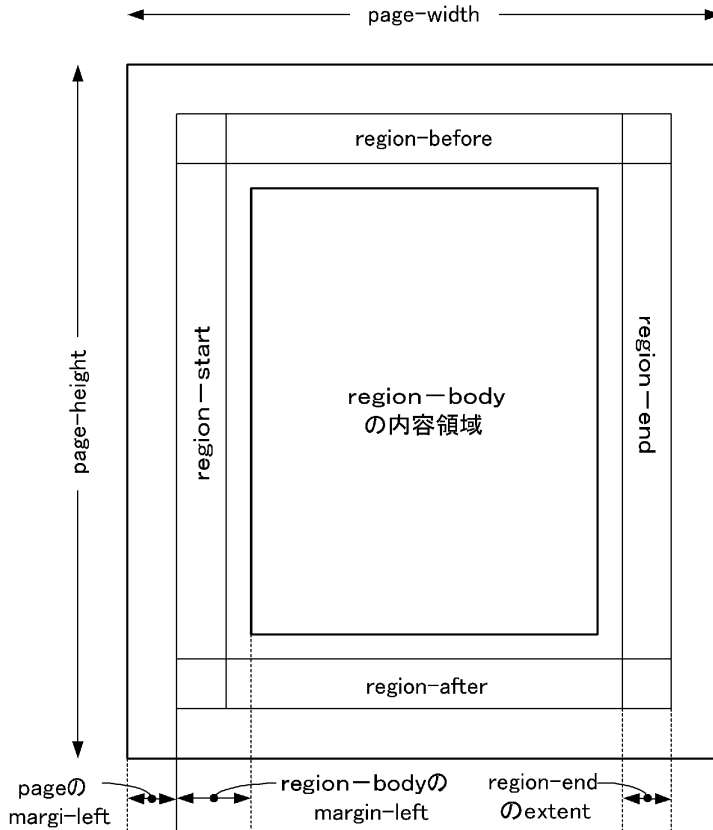
## 2. XSL-FO の基礎

### 用紙と領域

#### 用紙と領域の設定

XSL-FO は書籍としての体裁を意識して XML 文書をフォーマットするものです<sup>(3)</sup>ので、紙ベースの媒体にページ単位で出力することを意識しています。フォーマット結果や PDF 様式に出力したものを Windows 画面上で閲覧したり、プリンタを用いて出力することができます。したがって用紙サイズとプリンタの出力機能に依存するところがあります。

#### 領域の位置関係



まず使用する用紙の寸法などを設定します。用紙サイズの中に、上下左右にマージンをとって作業領域（ページ参照領域）を設定します。これはプリンタによっては用紙の全域一杯には出力できなかったり、プリントアウトしたものを綴じて使用するとき、綴じ代として残す余白などです。用紙サイズ一杯に文字やイラストを配置したり、あるいはオーバーフローで所定の領域をはみ出しても、Windows の画面上では用紙サイズ内はすべて表示されますが、プリンタから用紙一杯に出力できるとは保証できません。

region (領域) の位置関係

(3) 一部音声に関する Aural プロパティが含まれていますが、この解説では全く触れていません

作業領域の中にさらに上下左右にマージンをとって、ボディ領域 (**region-body**) を設定します。ここが本文領域です。

また、作業領域の上下左右の辺から内側に向かって所定の寸法をとって、それぞれ4つの領域を設定することができます。これらは本文領域の上側にヘッダー領域 (**region-before**) として通し柱などと言われる書名や章、節などの表示を配置したり、本文領域の下側にページ番号 (ノンプル) や、通し柱などを配置するフッター領域 (**region-after**) となります。左右にも同様に領域が取られます。左側を **region-start**、右側を **region-end** と呼びます。

この図の **before**、**after**、**start** および **end** と呼ばれる領域は通常の横組みの場合で、行頭側を **start**、行末側を **end** と言い、行の進む方向に対して初めの方を **before**、終わりの方を **after** と言います。縦組みなど組み方向が違う場合には、上下左右との関係が変わりません。

ここで注意しなければならないのは、ボディ領域を決定する上下左右のマージンと、**region-top**、**-bottom**、**-start** および **-end** の4つの領域とは別々に設定されることです。これらはボディ領域の上下左右の値と同じにすればお互いに接して配置されます。実務上はそれぞれの領域がお互いに干渉しないように値が設定されるでしょう。

## 用紙と領域設定の例

ページ領域の設定を "Standard-PagemasterA" で確認して見ましょう

```
<fo:simple-page-master page-width="210mm" page-height="297mm"
  margin-top="10mm" margin-bottom="10mm" margin-left="5mm"
  margin-right="5mm" master-name="Standard-PageMasterA">
  <fo:region-body background-color="yellow" margin-top="8mm"
    margin-bottom="8mm" margin-left="8mm" margin-right="8mm" />
  <fo:region-before background-color="aqua" extent="5mm" />
  <fo:region-after background-color="fuchsia" extent="5mm" />
  <fo:region-start background-color="navy" extent="5mm" />
  <fo:region-end background-color="silver" extent="5mm" />
</fo:simple-page-master>
```

**<fo:simple-page-master** が、一枚のページ設定を表します。 **master-name="Standard-PageMasterA">** で用紙に名前を付けて括弧で囲っています。その間に先ずページの幅 **page-width** と高さ **page-height** が設定され、**margin-top**、**bottom**、**-left**、**-right** の余白が設定されます。

更に、**region-body** が **margin-top**、**-bottom**、**-left**、**-right** にそれぞれ 8mm ずつ取って設定されています。このとき **region-body** の **margin** は先に取った用紙の **margin** の内側に取られます。

続いて、**region-before**、**-after**、**-start**、**-end** がそれぞれの **extent** として 5mm ずつ取られます。



ここで、用紙の margin には、"top"、"bottom"、"left"、"right"が使われています。これは、用紙の上下左右を云います。

これに対して、region には、"before"、"after"、"start"、"end"が使われています。これは前ページでも説明しましたように、文字を組んでいくときの行頭側を"start"、行末側を"end"と言い、行が並べられて行く方向に対して前の方を"before"、後の方を"after"と言います。

このように、region やその他組み方向と関係するものに対しては、横組みか縦組みかによって上下左右との関係が変わりますので注意が必要です。

### **"page-height"**

用紙の縦サイズを指定します。

プロパティの値：

auto | <length>

値の意味は：

**auto**

実装の省略値指定によります。

**<length>**

ページの幅を、固定値で指定します。

### **"page-width"**

用紙の横サイズを指定します。

プロパティの値：

auto | <length>

値の意味は、page-height に準じます。

## **writing-mode**

fo:simple-page-master に writing-mode の設定がないので、デフォルトで横組みとして扱われています。writing-mode とは、文字の組まれていく方向、即ち横組みか縦組みかなどの設定です。writing-mode="lr-tb"は横組み、writing-mode="tb-rl"は縦組みを表します。

### **"writing-mode"**

プロパティの値：

lr-tb | rl-tb | tb-rl | lr | rl | tb | inherit

値の意味は：

**lr-tb**

行内構成要素及び行内のテキストは右向きに書かれる。行及びブロックは、下向きに配置される。

**注記** 代表的には、これは通常の”アルファベット”テキストに指定される表記方向である。

- ・ 行内進行方向を右向きとする。テキスト内に左向きに読む文字が表記されている場合、グリフ領域の進行方向は、Unicode 双方向(BIDI)アルゴリズムによって、更に変更されてもよい。
- ・ ブロック進行方向を下向きとする。
- ・ シフト方向を上向きとする。

**rl-tb**

行内構成要素及び行内のテキストは左向きに書かれる。行及びブロックは下向きに配置される。

**注記** 代表的には、この表記方法はアラビア語及びヘブライ語のテキストに使用される表記方向である。

- ・ 行内進行方向を左向きとする。テキスト内に右向きに読む文字又は数字が表示される場合、グリフ領域の行内進行方向は、Unicode 双方向(BIDI)アルゴリズムによって更に変更されてもよい。
- ・ ブロック進行方向を下向きとする。
- ・ シフト方向は上向きとする。

**tb-rl**

行内構成要素及び行内のテキストは下向きに書かれる。行及びブロックは左向きに配置される。

**注記** 代表的には、この表記方向は中国語及び日本語のテキストで使用される表記方向である。

次の方向を確立する。

- ・ 行内進行方向を下向きとする。
- ・ ブロック進行方向を左向きとする
- ・ シフトの方向は右向きとする

**lr**

lr-tb の簡略形

**rl**

rl-tb の簡略形

**tb**

tb-rl の簡略形

### XSL1.0 との非互換

XSL1.1 で追加された `from-page-master-region()` 関数により、XSL1.0 との非互換が生じています。XSL 1.1 では、`fo:region-*` に `writing-mode` や `reference-orientation` を指定しても、そのままでは無視されることになっています。これらの指定を有効にするためには、`fo:page-sequence` へ

```
writing-mode="from-page-master-region()"
reference-orientation="from-page-master-region()"
```

の指定が必要です。また、常に XSL1.0 互換として動作させるのならば、オプション設定ファイルで `default-from-page-master-region="true"` を指定しておくこともできます。

## 簡略設定

XSL-FO では余り推奨していませんが、CSS で使われる簡略設定が使うことができます。

### 用紙サイズの簡略設定

上の例では用紙サイズの設定を、`page-width="210mm" page-height="297mm"`などとしていました。簡略設定の `size` 属性を用いるとこれを簡略記述することができます。

#### "size"

プロパティの値：

```
<length>{1,2} | auto | [ <page-size> || [portrait | landscape ] ]
```

値の意味は：

**<length>{1,2}**

`size="210mm 297mm"`のように `width` と `height` とを並べて書くことができます。

**auto**

システムのデフォルト設定に従います。

**[<page-size>||[portrait|landscape]]**

用紙サイズを指定します。下記の表に示す CSS の `size` を参照してください。

"portrait"と"landscape"は用紙の"縦置き"と横置き"を表しています。

`size="A4 landscape"`と指定すると、"A4 用紙を横置きで"と指定したことになります。

||

`<page-size>`と`[portrait|landscape]`とを併記することを意味します。

<page-size> に以下が指定できます。大文字小文字は区別されません。

<page-size>	短×長		<page-size>	短×長
A3	297×420mm		Letter	8.5×11in
A4	210×297mm		Legal	8.5×14in
A5	148×210mm		Ledger	11×17in
A6	105×148mm		Statement	5.5×8.5in
B4	250×353mm		Executive	7.25×10.5in
ISO-B4	250×353mm		C	17×22in
JIS-B4	257×364mm		D	22×34in
B5	176×250mm		E	34×44in
ISO-B5	176×250mm		Quarto	8.5×10.83in
JIS-B5	182×257mm		ISO-Designated	110×220mm
B6	125×176mm		ISO-C3	324×458mm
ISO-B6	125×176mm		ISO-C4	229×324mm
JIS-B6	128×182mm		ISO-C5	162×229mm
Folio	210×330mm		ISO-C6	114×162mm
Hagaki	100×148mm			

## margin の簡略記法

領域の設定で、margin-top="10mm" margin-bottom="10mm" などと 4 方向の値を別々に書きましたが、簡略記法では次のように書くことができます。

margin="10mm"

ブロック領域の margin に対して、マージン幅を 1 つ以上 4 つまで設定できます。

### "margin"

プロパティの値：

<margin-width>{1,4}

## 属性値の読み方

属性値 (プロパティ) の値にある <length>{1,2} などは次のように読みます。

### "属性値の読み方"

<length> 又は <margin-width>

長さの寸法を指定します。10mm、2em、などと数値と単位を書きます。

**{1,4}**

寸法指定の値を幾つ設定できるかを示します。この場合は、1つ以上4つまでを表します。寸法指定が2つ以上ある場合は、スペースで分離します。

**値が1つの場合**

上下左右のすべてに同じ値が適用されます。

**値が2つの場合**

上下マージンが最初の値を、左右マージンに2つ目の値が適用されます。

**値が3つの場合**

上マージンに最初の値が、左右マージンに2つ目の値が、下マージンに3つ目の値が適用されます。

**値が4つの場合**

上マージン、右マージン、下マージン、左マージンにそれぞれの値が適用されます。

**<page-size>**

page-size の表に示す、"A3"、"A4"などの値を示します。

**[portrait | landscape]**

"portrait"または"landscape"を選択します。真ん中の"|"は"or"の意味です。

||

<page-size>と[portrait|landscape]とを併記することを意味します。

## 簡略記法の注意点

XSL のプロパティでの簡略記法は CSS の定義を引き継いでいるので、その値は CSS のように評価されます。つまり、

`margin="0pt -10pt"`

は、ひとつの式としてではなく、ふたつの値と評価されます。しかし、簡略記法でないときは、これはひとつの式として評価されます。例えば、次はひとつの式です。

`margin-left="0pt -10pt"`

AH Formatter V5.0 では、簡略記法でのこのようなあいまいな表現に対して、次のように処理します。

- "0pt 10pt" のようにひとつの式になり得ないときはふたつの値
- "0pt -10pt" のように符号と数値がくっついているときはふたつの値
- "0pt - 10pt" のように符号と数値の間に空白が含まれるときはひとつの式
- "0pt-10pt" はエラー (XSL 仕様の 5.9.5 Numerics 参照)

簡略記法中で式を使うときに、括弧で囲むなどすることもできます。