

見本

バッチ組版のための
XSL-FO 指南

2011 年 11 月第 4.0 版

 有限会社イー・エイド



アンテナハウス株式会社

「バッチ組版のための XSL-F0 指南」
藤島 雅宏・編著
アンテナハウス株式会社・協力
第 4.0 版 2011 年 11 月 11 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⁽¹⁾の勧告による XSL1.1⁽²⁾を基本としてます。近年 CSS3.0 の改良に合わせて XSL-FO (XSL Formatting Object) にも多くの機能がアンテナハウス拡張機能として追加されました。アンテナハウス(株)の AH Formatter V6.0 に実装された新たに追加された機能の使い方を含めて解説をします。

AH Formatter V6.0 は、2011 年 10 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 をお使いの皆さん方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、2010 年 10 月 20 日に日本語化されて日本規格協会から発行された JIS X 4179 を参照して下さい。また、Formatter V6.0 のオンラインマニュアルにも拡張機能に対する解説や技術資料がありますので参照して下さい。

AH 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 又は 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 AH Formatter V5.x と AH Formatter V6.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、V6 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 V6.x へ移行していくので、この解説書も最新版の AH Formatter V6.0 に合わせて解説しております。

(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。

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 AH 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性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。

特に気を付けなければならない点は、AH Formatter V6.0 では、独自の追加プロパティが AH Formatter V5.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 以降で追加された拡張機能を示すために、[V5.0] 及び [V6.0] 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

本書 V4.0 版では AH Formatter V6.0 で新たに加わった拡張機能を解説すると共に、これまでの版で使いにくかった所や例題を改善しました。

トンボ出力は、アンテナハウス(株)の追加仕様であり、AH Formatter V6.0 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。この他 V6.0 Lite では、拡張機能でサポートされていないものがあります。

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感謝致します。

— 目 次 —

1. XSL-FO と XSLT	1
XSLT について	3
p 要素	3
XSLT による XSL-FO の作り方	6
2. XSL-FO の基礎	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
要素と属性（プロパティ）	10
fosimple-page-master と関連属性	10
用紙サイズの簡略設定	11
margin 設定	13
margin の簡略記法	13
属性値の読み方	13
簡略記法の注意点	14
領域設定と関連属性	14
ページ指定の簡単な例	16
writing-mode="lr-tb"の例（横組み）	16
writing-mode="tb-rl"の例（縦組み）	19
FO 要素と属性	22
新しく出現した FO 要素	22
新しく出現した属性	24
reference-orientation と writing-mode	24
3. 領域の幾何的定義	27
領域の要素	27
行内領域(inline-area)	28
正規割付長方形	28
大割付長方形	28
XSL-FO の詳細へ	28
4. フォントサイズの設定	29
基本用紙サイズ・基本文字サイズ	29
長さの単位	29
font-size の基本仕様	30

フォントサイズを直接設定する	30
段落単位でフォントサイズを設定する	31
フォントサイズの絶対指定	31
フォントサイズの相対値指定	32
5. フォントの太さ	34
<code>font-weight</code> の基本仕様	34
太さの絶対値指定	34
6. フォントストレッチ	36
<code>font-stretch</code> の基本仕様	36
絶対指定	37
縦組み時の絶対指定	37
相対指定	39
縦組み時の相対指定	39
パーセント指定	41
縦組み時のパーセント指定	41
数値指定	42
縦組み時の数値指定	42
7. <code>font-stretch</code> と"空白"	43
<code>font-stretch</code> 指定時の"半角/全角スペース"	43
全角文字幅 "wch" v5.1 の導入	44
横組み場合	44
縦組みの場合	44
8. フォントファミリーの設定	45
<code>font-family</code> の基本仕様	45
直接指定	45
共通指定	46
代替えフォントの指定	47
<code>font-selection-strategy</code> の基本仕様	48
9. 文字の指定	50
文字をコードポイントで指定	50
異体字選択	50
OpenType フォントの字体切り替え	51
CJK の切替	52
リガチャ	52
代替え字形の指定 <code>axf:alt-glyph</code>	53
文字列の置換	54

数字列の置換	54
漢数字	55
漢数字の形式指定	55
漢数字文字指定	56
位取り文字指定	57
漢数字への変換の仕方	58
10. 文字修飾	59
文字の修飾いろいろ	59
文字修飾のプロパティ	61
<code>axf:text-underline-position</code>	63
<code>baseline-shift</code>	66
11. 約物の処理	67
行頭・行末での全角約物の半角扱い	67
連続する約物の詰め処理	70
句読点のぶら下げ処理と行頭突き出し処理	74
行頭・行末禁則処理	76
禁則約物の追加及び削除	80
追い込み処理	82
<code>axf:avoid-widow-words</code>	83
12. 和・欧文間の空き	85
和・欧文間の空きに対する拡張	85
<code>axf:text-autospace-width</code>	85
13. 文字間の空きの設定	89
<code>letter-spacing</code> による調整	89
<code>justify</code> との関係	90
<code>letter-spacing</code> の調整	91
14. 段落内の文字の配置	93
<code>text-align</code>	93
<code>text-align</code> の設定	94
行頭字下げ	97
15. 行の高さ指定	98
絶対値で行の高さを指定	98
相対値で行の高さを指定	100
16. 異サイズの文字があるときの行の高さ	101
絶対値で行の高さを指定した場合	101
相対値で行の高さを指定した場合	102

line-height の纏め	103
line-height の値とは	103
17. line-stacking-strategy と行の高さ	104
line-stacking-strategy による行の高さの変化	104
line-stacking-strategy とは	107
行内領域との関係	108
18. 組方向	109
ページ内に横組みと縦組みの混在	109
縦組みの例	110
縦組み中横組み	111
平体指定時の縦組み中横組み	112
<code>fo:inline-container</code> での組方向変更	113
19. 英文組版	114
ハイフネーションなど	114
欧文組版のための拡張	115
一般的なハイフン付け特性	117
欧文組版のためのその他のプロパティ	119
SOFT HYPHEN のグリフの出力	122
20. ルビ機能の拡張	123
V6.0 によるルビ拡張	123
モノルビ	123
グループルビ	123
熟語ルビ	123
ルビ拡張要素	126
ルビ拡張プロパティ	127
熟語ルビ	134
縦組みでのルビ	136
21. 圈点	137
圏点関連のプロパティ	138
圏点の作例	141
22. 保持及び分割	144
分割	144
改行・改段・改ページ	146
保持	146
分割の禁止	147
23. 境界線	150

境界線の種類と太さ（対角線を含む）	150
24. 境界線を引く指定の仕方	152
境界線を引く辺を指定する	152
別々に各辺の指定をする	152
border 指定の省略形	153
padding を使った場合	154
padding 指定のプロパティ	155
margin を使った場合	156
その他のマージン特性	157
一般的な行内レベルのマージン特性	158
25. ボーダーとパディング	160
26. 角丸	165
ボーダーやエリアを角丸にする	165
影を付ける	166
27. テーブル	168
テーブルの構造	168
テーブルの要素	168
テーブルの基本的サンプル	172
テーブルにおける注意事項	173
テーブルに用いられる一般のプロパティ	174
テーブル独自のプロパティ	175
28. テーブルのカラム幅の設定	183
自動モード	183
固定モード	183
29. カラムスパンとロースパン	184
30. テーブルのボーダーとパディング	185
自動レイアウトの場合	185
固定レイアウトの場合	186
テーブルセルに角丸	187
31. セル内の文字の配置	188
text-align	188
text-align="string"	189
text-align="string" の拡張	189
display-align	190
display-align の拡張値："justify"	190
relative-align	191

32. テーブルのページ分れ	192
表のヘッダ・フッタの繰り返し	192
表のヘッダ・フッタを繰り返す	192
ヘッダ・フッタの繰り返しの拒否	193
分断されたテーブルセルの上下の線を生かす	194
分断されたテーブルセルの内容を繰り返す	195
33. テーブルマーカー	197
テーブルの続きを表示	197
小計の表示	200
34. レイアウトに変化を付ける	203
テーブルの外枠の太さを変える	203
テーブルの外枠に角丸を指定する	204
テーブルの外枠とヘッダの一体化	204
テーブルの外枠を消す	205
斜線を用いた例	206
テーブルセル内での縦組み指定	207
表頭を回転させる	208
35. テーブルの回転	211
幅広のテーブルを収める	211
36. 箇条書き	214
リストの構造	214
リストの FO 要素	214
リスト形式の基本的構造	216
リスト形式のフォーマット	217
番号なしリストのフォーマット	218
番号付きリストのフォーマット	220
定義リストのフォーマット	222
リストブロックをネストする	223
実例中での箇条書きの例	224
37. リーダー	226
リーダーの FO 要素	226
リーダーのプロパティ	226
fo:leader	228
fo:leader の単純な指定	228
leader-pattern の値を"rule"にする	229
leader-pattern の値を"use-content"にする	230

リーダーを広げる	231
use-content に特殊文字を	232
1 行の中で fo:leader を 2 回使う	233
38. 行外ブロックの処理	234
float の FO 要素と属性	234
float の FO 要素	234
fo:float の属性	234
開始・終了フロートと図形	237
開始・終了フロートと intrusion-displace 指定	239
フロートと clear=inside/outside	241
前方/開始/終了フロートと脚注	243
フロートと脚注	244
39. 拡張フロート機能	246
ページや段を意識した拡張	246
フローイング関係の拡張プロパティ	246
新しく拡張された単位	254
フロート拡張機能の事例	255
ページのフロートと段のフロート	255
40. 脚注・傍注	257
脚注と傍注	257
脚注に対するテンプレート	258
脚注との分離など	260
脚注番号に対する拡張機能	260
段組時の脚注	262
脚注に関する拡張機能	263
脚注の配置	263
41. 段組み	267
ページ全体を段組にする	267
段組に関係するプロパティ	267
block-container の中で段組にする	268
balance を制御する	269
42. 段間罫	271
通常の段間罫	271
段間罫の制御	272
最後の段に罫を引かない	272
最後の段にも罫を引く	273

段の後に段間罫を引く	273
罫線の長さ指定	274
段間罫の位置指定	275
段間罫の拡張プロパティ	276
43. 改訂バーとチェンジバー	278
改訂バー	278
チェンジバー	281
特性の違い	285
44. 行番号	288
行番号をつける	288
行番号表示の複雑な指定	289
行ナンバーに使われるプロパティ	291
45. ブロックの配置	295
ブロック間のスペース	295
46. ブロックコンテナ	296
絶対位置指定 (<code>absolute-position="absolute"</code>)	296
絶対位置決め特性のプロパティ	297
ブロック・コンテナの位置と大きさの指定	299
ブロック・コンテナの位置と大きさの指定（段組）	301
絶対位置指定のブロック・コンテナ	302
固定位置指定のブロック・コンテナ	303
47. オーバーフロー制御	304
ブロックコンテナオーバーフローの基本処理	304
<code>overflow</code> のプロパティ	305
オーバーフローの拡張機能	306
オーバーフローの追加仕様 1	309
オーバーフローの追加仕様 1（縦組み時）	310
オーバーフローの追加指定 2	311
オーバーフローの追加指定 2（縦組み時）	312
オーバーフローの追加指定 3	313
オーバーフローの追加指定 3（縦組み時）	314
48. 画像の取り込み	315
外部の画像イメージを取り込む	315
背景に画像イメージを取り込む	316
画像に関するプロパティ	318
49. 各種グラフィックの取り込み	320

Windows BMP	320
GIF: Graphic Interchange Format	320
WMF: Windows Metafile	321
EMF: Enhanced Metafile	322
PNG: Portable Network Graphic	323
JPEG	324
TIFF の圧縮方法とカラー空間	325
50. 画像倍率の選択	329
高さ倍率のスケールリストが定義されている場合	329
scale-to-fit	329
scale-down-to-fit	330
scale-up-to-fit	331
幅倍率のスケールリストが定義されている場合	332
scale-to-fit	332
scale-down-to-fit	333
scale-up-to-fit	334
高さ倍率のスケールリストが定義されていない場合	335
scale-to-fit	335
scale-down-to-fit	336
scale-up-to-fit	337
幅倍率のスケールリストが定義されていない場合	338
scale-to-fit	338
scale-down-to-fit	339
scale-up-to-fit	340
固有倍率のある場合	341
51. SVG の取り込み	342
SVG を画像として取り込む	342
Scalable Vector Graphics	344
Instream SVG	345
Gradation	347
Gray Scale Gradation	348
Gray Scale	348
52. MathML	349
53. PDF の埋め込み	350
オンラインマニュアルから	350
PDF 埋込の事例	351

54. XSL-FO の構造	352
ページマスター	352
個々のページマスターの設定	352
ページシーケンス	355
ページシーケンスのページ数制御	357
ページシーケンスの繰り返し	358
XSL-FO のヘッダ部の実例	359
fo:layout-master-set まで	359
解説	360
各ページ毎の組指定	362
表紙の組指定	362
目次の組指定	362
本文と索引の組指定	363
55. 背景画像	364
ページに背景画像を指定	364
背景画に対する基本的プロパティ	364
背景画に対する拡張プロパティとその値	365
代替えテキスト	367
ext-background-image の例	367
56. 本書のスタイルシート	369
XSL-FO 指南の XML データ	369
スタイルシート	370
XSL の構造と外部参照スタイルシート	370
ページ書式の設定	370
page-sequence-master	371
本文処理	372
柱の設定	372
ノンブルの処理など	373
ページ番号処理	374
marker	378
見出しの作成	379
ツメの作り方	380
ツメの領域を設定する	381
ツメを付ける領域を決める	381
繰り返しを設定する	382
本文とのリンクを設定する	383

ツメをデザインする	383
トンボの設定と塗り足し	384
57. テーブルに対するスタイルシート	385
表に使われる要素	385
XML コンテンツの例	386
表を表す XSL-FO の要素	387
スタイルシートの作成	388
プロパティセット	388
"table" と "table/title" のテンプレート	389
"col" のテンプレート	390
"thead" と "tfoot" のテンプレート	391
"tbody" のテンプレート —<tr>—	391
"tbody" のテンプレート —<th>—	392
"tbody" のテンプレート —<td>—	393
"cell-span" のテンプレート	394
処理の実例	394
出力例	395
その他の事例	396
58. リストに対するスタイルシート	397
XML コンテンツにおけるリスト形式	397
XSL-FO におけるリスト形式	398
番号付きリストを処理するテンプレート	399
リストに対するプロパティセット	399
"ol" に対するテンプレート	399
"ol/li" に対するテンプレート	400
ラベルの書式	401
位置指定の纏め	401
事例	401
出力例	402
format プロパティの拡張	402
番号なしリストを処理するテンプレート	405
"ul" のテンプレート	405
"ul/li" のテンプレート	406
ラベルに画像を用いる	407
番号なしリストの例	407
出力例	408

定義リストを処理するテンプレート	408
定義リストに対するプロパティセット	408
2つの定義型リストの形式	409
"dl.format.list"のテンプレート	409
"process.dl.list"のテンプレート	410
"dt"のテンプレート	412
"dd"のテンプレート	412
XSL-FO での定義型リストの例	412
HTML での定義型リスト	413
59. トンボ	414
トンボ出力	414
クロップオフセット	414
塗り足し	415
トンボマーク	415
60. 目次と索引	418
事例イメージ	418
索引のページ	423
ノンブルの振り方	424
目次	424
図版目次	424
序文	424
本文	425
本文続き	425
用語解説	425
巻末索引	426
ノンブルに総頁も付加する	426
目次の作り方	426
索引取りの仕方	428
ページ表示の整理いろいろ	429
すべてそのまま表示	429
重複するページはマージ	430
連続するページを纏める	430
範囲内の図版などはマージ	431
連続ページはマージ	431
すべてマージ	431
61. ブックマーク（しおり）	432

本文での指定	432
ブックマークの設定	433
しおりの要素とプロパティ	434
62. 流し込みマップ	437
流し込みマップによる領域指定の柔軟性拡大	437
複数の流れこみをそれぞれの領域に	439
"A"と("B1"/"B2")を 1 つの領域"R"に流し込む	442
1 つのフロー"A"を 2 つの領域"R"と"S"に流し込む	443
"A"/("B1"/"B2")を領域"R"/"S"に流し込む	445
63. 多言語組版	448
対応言語	448
64. 備 考	451
プロパティのデータ型	451

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いましたが、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くべき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つてから、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繁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い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い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した。

スタイルシートは、XML コンテンツの中から必要なデータを取りだして、書籍の形に組むための組版指示を、XSL-FO の仕様に従つ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いた 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が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 AH Formatter だけです。AH 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 AH 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 又は 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 AH Formatter に XML と XSL スタイルシートを与えると、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。

共通に使われない場合などは、アトリビュートセットを設けないで、テンプレートの中で一々設定しても構いません。

テンプレートの中に設定

```
<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良いと思います。

XSLT による XSL-FO の作り方

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 につ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。しかしながら、XSL-FO を活用するには、スタイルシートについて基礎的な知識は必要と思われますので、巻末の方で纏めて解説しました。

"[テーブルに対するスタイルシート\(Page-385\)](#)"、"[リストに対するスタイルシート\(Page-214\)](#)"、"[ツメの作り方\(Page-380\)](#)"テンプレートを参照して下さい。

本書が、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、ご活用頂ければ幸甚です。

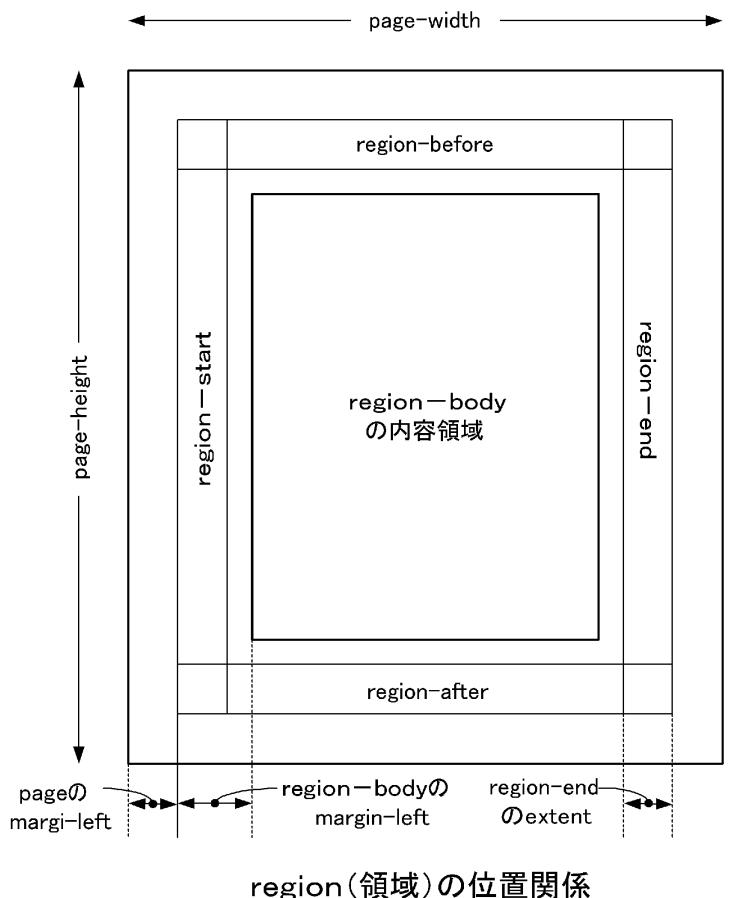
2. XSL-FO の基礎

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⁽³⁾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。

(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 (ノンブル) や、通し柱などを配置するフッタ領域 (region-after) となります。左右にも同様に領域が取られます。横組みを基本とする用紙設定では、左側を region-start、右側を region-end と呼びます。

この図の before、after、start 及び 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す。

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 及び -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" (次頁) で確認して見ましょう

Standard-PagemasterA

```
<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っています。その間に先ず用紙の幅 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。

領域の設定の例

ページの領域 Standard-PageMasterA

領域の種類	region-name	background-color	margin				extent
			top	bottom	left	right	
用紙			10mm	10mm	5mm	5mm	
fo:region-body	既定値	yellow	8mm	8mm	8mm	8mm	
fo:region-before	既定値	aqua					5mm
fo:region-after	既定値	fuchsia					5mm
fo:region-start	既定値	navy					5mm
fo:region-end	既定値	silver					5mm

ページ領域の設定はページモデルの最初の設定です。

用紙サイズ（横 210mm、縦 297mm）に対して、上下に 10mm、左右に 5mm のマージンをとっています。（一番外側の白い部分）

更に上下左右に 8mm のマージンを取ってボディ領域（黄色い部分）をとっています。

続いて、上（before）、下（after）、左（start）及び右（end）にそれぞれ領域をとります。領域の幅はすべて 5mm にしてあります。これらはボディ領域を取るときに設定した上下左右のマージン内に作られています。

プログラムでは以下のように書かれます。

```
<fo:simple-page-master margin-top="10mm" margin-bottom="10mm"
    margin-right="5mm" margin-left="5mm" page-height=" 297mm "
    page-width="210mm" 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>
```

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。

要素と属性（プロパティ）

XSL-FO で定義されている FO (フォーマット化オブジェクト) 要素があります。FO 要素内には対応できる属性(プロパティ)が設定できます。それぞれ必要に応じて解説していきます。

ここで用紙と領域設定に現れる要素とプロパティを解説します。

fo:simple-page-master と関連属性

fo:simple-page-master

ページのサイズや各領域の寸法などの構造を指定します。図：region (領域) の位置関係に示すようにメインの region-body とその周りの 4 つの領域を指定します。

内容

(region-body+, region-before?, region-after?, region-start?, region-end?)

適用される属性

- 一般の Margin プロパティ
- master-name
- page-height
- page-width
- size
- reference-orientation
- writing-mode

内容の region-body+ とあるのは、必ず 1 つ以上の region-body を設定する必要があります。他の region-before? などは、必ずしも設定する必要はありませんが、設定するならそれぞれ 1 回設定します。

適用される特性のそれぞれの意味は、それぞれ後述されます。

master-name

マスタの名前を識別します。

プロパティの値 :

<name>

値の意味は :

<name>

マスタを識別名前で、空であってはいけません。また、一意でなければなりません。

fo:page-sequence-master 上で指定された場合、マスタ名を識別します。fo:page-sequence 上の特性の値として連続して参照され、ページの生成する際に page-sequence-master の使用を要求します。

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。

用紙サイズの簡略設定

CSS で使われる簡略設定が使うことができます。

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 とを並記します。値が一つしかない場合は、"width"と"height"が同じ値の正方形と見なします。

auto

システムのデフォルト設定に従います。

<page-size>

用紙サイズを指定します。表<page-size>を参照してください。

[**portrait** | **landscape**]

"portrait"と"landscape"は用紙の"縦置き"又は"横置き"を指定します。

size="A4 landscape"とは、"A4 用紙を横置きで"を意味します。

||

<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>

<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 設定

margin-top, margin-bottom, margin-left, margin-right

プロパティの値 :

<margin-width>

margin-width 値の意味は :

auto

CSS2 の街頭部分を参照

<length>

固定幅を指定します。

<percentage>

パーセント値は、親のブロック幅に関して算出されます。

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>

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。

<margin-width>{1,4}

"margin"では、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。

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 以降 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。

領域設定と関連属性

fo:region-body

ページの中央に位置し、本文を流し込む領域です。fo:simple-page-master によって、用紙から左右上下の margin をとったページ参照領域の、更に内側の上下左右に region-body の margin を取って、区画表示領域が設定されます。

内容

EMPTY

適用される属性

一般の Border, Padding 及び Background プロパティ

一般の Margin プロパティ

clip	region-name
------	-------------

column-count	reference-orientation
--------------	-----------------------

display-align	writing-mode
---------------	--------------

overflow	
----------	--