

# Antenna House XSL Formatter 拡張仕様使いこなしガイド

---

# 前書き

Extensible Stylesheet Language (XSL) は、Extensible Markup Language (XML) 文書を組版のための形式へ変換するための仕様です。XSL は XML 文書を変換するための仕様である XSLT と、フォーマット化したオブジェクトの XML 仕様である XSL Formatting Objects (XSL-FO) に分かれます。「スタイルシート」として世界的に認知されているのは「Cascading Style Sheet (CSS)」ですが、XSL-FO は CSS 2 や DSSSL を土台として、XML 文書をページメディアとして表現する仕様です。詳細については、W3C の XSL 仕様などをご覧ください。

XSL-FO 1.1 (以下 XSL 1.1) は 2006 年 12 月に W3C 勧告となりました。XSL-FO プロセッサである Antenna House XSL Formatter (以下 AH XSL Formatter) は XSL 1.1 のほとんどの仕様を忠実に実装しています。しかし、業務などで実際に XSL-FO を使用するユーザからは、この仕様で規定されたよりも細かく組版を制御する機能の要望が多数寄せられます。AH XSL Formatter はユーザの要望に応え、さまざまな拡張機能を実装してきました。

本書は AH XSL Formatter 拡張仕様を用いた XSL-FO でどんな文書を組版することが可能なのか、どんな組版要望に応えられるのかを紹介することを目的として制作されました。

AH XSL Formatter のマニュアルはバージョンアップに伴い随時更新されます。本書とマニュアルで食い違う箇所がある場合、マニュアルでの表記、説明が優先されます。

本書は AH XSL Formatter V7.1 改訂 2 版を用いて組版されています。また、本書で紹介する AH XSL Formatter の動作も V7.1 改訂 2 版を基準としたものになります。

## 本書が対象とする読者について

拡張仕様や利用方法の説明については、ページシーケンスやブロックコンテナをはじめとする、XSL-FO の仕様についてある程度の知識を有している方を想定しています。また「所持している XML 文書を要件に沿った形で出力したい」「質の高い自動組版を実現したい」「XML に基づいた組版でどこまで可能なのか出力例を知りたい」といった方にも、AH XSL Formatter でどのような組版が実現可能かを紹介するショウケースとして本書が機能するように努めました。

本書から XSL-FO 仕様に興味を持ち、基礎から学びたいという方には弊社より刊行、公開している『XSL-FO の基礎 第 2 版』をおすすめします。

出力形式については、PDF 出力を中心に示しています。

## 本書中の表記について

本書では、具体的な要素、プロパティ、指定する値を等幅書体で表記しています。

特定の文字について言及するとき、Unicode での名称、「字形の例 (空白を示す文字類を除く)」、Unicode 符号位置を「U+」と 16 進数の値で表記しています。たとえば文字「あ」については、HIRAGANA LETTER A 「あ」(U+3042)、と表記します。

具体的な値ではなく `<number>` など値の型を示すもの、変数を `<LESS-THAN SIGN 「<」 (U+003C)` と `<GREATER-THAN SIGN 「>」 (U+003E)` 罫いの斜体で表記しています。

本文中で要素を明示するときは<fo:page-sequence>のように表記しています。

本文中で一部に\*を含む仕様名を表記をしているとき、その箇所をワイルドカードとして適合するすべての仕様名を示しています。

例示する FO で、一部読み易さのために、本来挿入すべきでない改行やインデントのある箇所があります。

表中の「√」は指定可能、「-」は指定不可能を表します。

## 用語について

**before、after、start、end top、bottom、left、right** は絶対的な配置方向を指します。しかし、組版では上下左右による指定であると不便ことがあります。before、after、start、end はページ上の相対的な配置方向を指します。writing-mode="lr-tb"であるようなレイアウトでは、left が start、top が before に対応します。

**インライン進行方向** 行内で文字が置かれていく方向です。行内進行方向とも呼びます。和文組版では JIS X 4051 『日本語文書の組版方法』および JIS X 4052 『日本語文書の組版指定交換形式』における「字詰め方向」に相当します。本書中では start 方向から end 方向を指します。

**ブロック進行方向** 行の数が増えていく方向です。和文組版では JIS X 4051 『日本語文書の組版方法』および JIS X 4052 『日本語文書の組版指定交換形式』における「行送り方向」に相当します。本書中では before 方向から after 方向を指します。

**段組** 本書では 2 段組以上の多段組を指して「段組」と表記しています。

**タブ** 本書ではタブ揃えに用いるオブジェクトを「タブ」、タブ文字を「TAB (U+0009)」と表記します。

**言語コード** 本書では、ISO 639、ISO 639-2 で規定される言語コードを指して「言語コード」と表記します。

**言語タグ** 本書では、言語コードと ISO 3166 で規定される国名コードをハイフンで結んだものを指して「言語タグ」と表記します。

**スクリプト** 本書では、ISO 15924 で規定されるスクリプトコード、またはスクリプトコードに対応する文字、表記記号の集合を指して「スクリプト」と表記します。

**欧文** 本書では、主にヨーロッパで使用されるスクリプトを指して「欧文」と表記します。欧文に分類されるスクリプトには Latn (ラテン文字)、Cyril (キリル文字) などがあります。ラテン文字には LATIN CAPITAL LETTER A 「A」(U+0041) などがあります。キリル文字には CYRILLIC CAPITAL LETTER DE 「Д」(U+0414) などがあります。

**CJK** 本書では、中国語、日本語、韓国語の言語またはスクリプトを総称して、「CJK」と表記します。

**和文** ひらがな、カタカナ、漢字、全角のアルファベット、日本語の文章に登場する約物などのスクリプトを総称して「和文」と表記します。和文を主な適用対象にしている仕様で、CJK に対し適用されるものもあります。

Unicode の General Category    Unicode の General Category ([http://www.unicode.org/reports/tr44/#General\\_Category\\_Values](http://www.unicode.org/reports/tr44/#General_Category_Values)) は、文字を大文字や小文字、数字などに区分しています。たとえば、LATIN CAPITAL LETTER A 「A」(U+0041) の General Category は Uppercase\_Letter (Lu) です。それぞれの区分について、本書では略字 (Abbr) 形で示します。

マニュアル    AH XSL Formatter のマニュアルを指します。日本語でのマニュアルの Web 版は <https://www.antenna.co.jp/AHF/help/ja/> で公開されています。

オプション設定    AH XSL Formatter の動作上の設定を指します。XML 形式のファイルとして AH XSL Formatter 動作時に読み込むことや、拡張仕様によって <fo:declarations> の子孫として記述することが可能です。

## 本書中の拡張仕様一覧について

本書の末尾に、本書に登場する拡張仕様の一覧を示しています。項目順はネームスペース、要素を示す <、> を省略したものになります。たとえば <axf:document-info> は 「d」 の項目となります。

## 拡張仕様項目の見方について

本書中の拡張仕様項目の表記について示します。

拡張仕様には、プロパティの拡張と要素の拡張が存在します。XSL 1.1 仕様または CSS3 仕様を基にした仕様で、適用範囲が拡張されている、または指定可能な値が拡張されている仕様については、**太字** で表記しています。要望がなく AH XSL Formatter が用意していない値については打ち消し線で表示しています。

AH XSL Formatter の動作上の設定には、オプション設定としてのみ制御可能なものも存在します。これらについても、抜粋して示します。オプション設定のネームスペースについては必要などきのみ表記します。

## 拡張要素、オプション設定の要素

Element Name	拡張要素の名称
Property	指定が必要あるいは有用なプロパティ
Parent	拡張要素を記述できる親要素

オプション設定で用いられる要素では、Element Name が Option Element Name となります。

Parent はその要素の親に置くことが可能な要素がごく限られるときに表記しています。

## 拡張プロパティ、オプション設定のプロパティ

Property Name	拡張プロパティの名称
Value	プロパティに指定可能な値
Initial	プロパティを指定しないときの初期値
Applies to	プロパティを指定可能な要素
Inherited	祖先での指定から値を継承するか。継承するときのみ「yes」を表記
Percentages	<percentage>で値を指定するときに基準とする対象

オプション設定で指定するプロパティでは、Property Name が Option Property Name となります。本書中では Inherited は継承されるときのみ、Percentages は<percentage>による指定が可能で、かつ基準となる対象が FO に存在するときのみ項目として表示しています。

Value 中の記号の見方を次に示します。記号は、組み合わせについてのものと、繰り返しについてのものがあります。

- 単純に並べられた値は、順序通りすべてが現われる必要がある
- 「&&」で区切られた値は、任意の順序ですべてが現われる必要がある。
- 「||」で区切られた値は、任意の順序で少なくともひとつ以上が現われる必要がある。
- 「|」で区切られた値は、どれかひとつが現われる必要がある。
- 「[」と「]」で囲まれた内容はグループとして扱う。
- 「\*」は 0 回以上の繰り返し。
- 「+」は 1 回以上の繰り返し。
- 「?」は 0 回または 1 回。
- 「#」は COMMA 「,」(U+002C) 区切りで 1 回以上の繰り返し。
- 「{N}」は N 回の繰り返し。
- 「{N,}」は N 回以上の繰り返し。
- 「{N,M}」は N 回以上 M 回以下の繰り返し。
- グループの後ろの「!」は、グループが必須で、少なくともひとつが現われる必要がある。

## Value

プロパティへ指定する値について、本文として説明しているもの、表として説明しているものがあります。

<プロパティ>	意味
<プロパティ>へ指定する Value	Value を指定したときの動作

## 複雑な Value

プロパティや関数などで、複雑な指定が必要なものがあります。このとき、プロパティの表では仮の Value を表記し、詳細を別の表に示すものがあります。

Value	Value Composition
<複雑な Value>	<複雑な Value の構成>

## 関数

プロパティの指定に、関数による指定を行うものがあります。

Function Name	関数の名称
Argument Value	関数の引数に指定可能な値

## 本書中の商標、登録商標について

Microsoft<sup>®</sup>、Windows<sup>®</sup>は米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

Adobe<sup>®</sup>、Adobe Acrobat<sup>®</sup>、Adobe Reader<sup>®</sup>、PostScript<sup>®</sup>は Adobe Systems Incorporated（アドビシステムズ社）の米国およびその他の国における商標または登録商標です。

Unicode<sup>®</sup>は、Unicode, Inc.の登録商標です。

PANTONE<sup>®</sup>およびその他の Pantone Limited Liability Company の登録商標は、Pantone Limited Liability Company が所有しています。

そのほか記載されている会社名、製品名等は、各社の登録商標もしくは商標、または弊社の商標です。

本文中では、<sup>®</sup>、<sup>™</sup>等のマークは省略されていることがあります。

# 目次

第1章 ショウケース.....	19
1.1 節 文書の種類と拡張のユースケース.....	20
1.2 節 観光パンフレット風のページデザイン.....	22
1.3 節 縦組小説風のページデザイン.....	26
1.4 節 自由配置の多い雑誌風のページデザイン.....	31
1.5 節 欧文雑誌風のページデザイン.....	35
1.6 節 論文誌風のページデザイン.....	42
1.7 節 数学参考書風のページデザイン.....	46
1.8 節 帳票風のページデザイン.....	50
第2章 AH XSL Formatter 拡張仕様と応用例 .....	55
2.1 節 単位.....	56
2.1.1 長さの絶対値を利用する.....	56
2.1.2 長さの相対値を利用する.....	56
2.1.3 単位のない「0」を指定する.....	57
2.1.4 解像度指定に dpi を利用する .....	57
2.1.5 px 単位の変換を制御する.....	58
2.1.6 ページサイズを判型で指定する.....	58
2.1.7 ユースケース：拡張された単位を利用する.....	59
2.2 節 ページ制御.....	61
2.2.1 見開きでページをまたぐ要素を配置する.....	61
2.2.2 ページシーケンスを入れ子で記述する.....	64
2.2.3 ページ数を制御する.....	64
2.2.4 ページ出力順を逆順にする.....	65
2.2.5 ページシーケンスマスタの繰り返しを制御する.....	65
2.2.6 任意の空ページを挿入する.....	66

2.2.7 ユースケース：異なるページサイズの図表を挿入する.....	66
2.2.8 ユースケース：PDF のページ順を変更して出力する .....	68
2.3 節 オブジェクト修飾.....	70
2.3.1 ボーダーを修飾する.....	70
2.3.2 波線の形状を制御する.....	70
2.3.3 二重線の修飾を変更する.....	71
2.3.4 ボーダーの角を丸める.....	71
2.3.5 ボーダー接合部の形状を指定する.....	72
2.3.6 対角罫を引く.....	73
2.3.7 区画のボーダーを表示する.....	73
2.3.8 オブジェクトにボックスシャドウを付加する.....	74
2.3.9 ブロックを変形する.....	74
2.3.10 行番号を付加する.....	77
2.3.11 行番号のフォントを修飾する.....	79
2.3.12 行番号を修飾する.....	80
2.3.13 行番号の位置を指定する.....	82
2.3.14 行番号領域の幅を指定する・揃えを変更する.....	82
2.3.15 行番号の表示を回転する.....	83
2.3.16 折り返し行の末尾に文字（行継続マーク）を付与する.....	84
2.3.17 行継続マークの位置を調整する.....	85
2.3.18 行継続マークの修飾を指定する.....	85
2.3.19 ブロックに付加する改訂バーに文字や画像を置く.....	86
2.3.20 下線、上線、打ち消し線を修飾する.....	87
2.3.21 テキストに引く線の位置を制御する.....	88
2.3.22 縦組テキストの傍線方向を制御する.....	89
2.3.23 テキストに影を付加する、ぼかしを制御する.....	90
2.3.24 ユースケース：ニューモーフィズムデザインの修飾.....	91
2.3.25 ユースケース：ソースコード表示を修飾する.....	92

2.3.26	ユースケース：吹き出し会話を作成する.....	95
2.3.27	ユースケース：改訂バーにリビジョン番号を付加する.....	97
2.4 節	段組.....	101
2.4.1	同じフローに異なる段組を混在させる.....	101
2.4.2	段ごとのコンテンツの高さを揃える.....	101
2.4.3	段間罫を引く.....	102
2.4.4	段間罫の長さとお揃えを指定する.....	102
2.4.5	段の内容が存在しない箇所で段間罫を引く.....	103
2.4.6	段によって要素の表示位置を変更する.....	103
2.5 節	表組.....	105
2.5.1	ページ、段分割でヘッダ、フッタの動作を変更する.....	105
2.5.2	表のヘッダ、フッタの注の繰り返しを制御する.....	105
2.5.3	セル分割時のコンテンツを制御する.....	106
2.5.4	幅の自動計算に利用する行数を制限する.....	107
2.5.5	セル内でのテキストのお揃えを制御する.....	108
2.5.6	セル内のテキストをブロック進行方向で均等割り付けにする.....	108
2.5.7	rowspan したセルがある表の高さを揃える.....	109
2.5.8	表のヘッダ、フッタで複数行を含むマークを取り出す.....	109
2.5.9	ユースケース：修飾した表を作成する.....	110
2.6 節	リーダー.....	114
2.6.1	リーダーのお揃えを制御する.....	114
2.6.2	リーダー自体を両端揃えにする.....	114
2.6.3	リーダーのある行を均等割り付けにする.....	115
2.7 節	リスト.....	116
2.7.1	ユースケース：箇条書きを段組にする.....	116
2.8 節	グラフィック（色、画像、背景、数式）.....	118
2.8.1	色を指定する.....	118
2.8.2	色名で指定する.....	118

2.8.3	色を CMYK で指定する .....	119
2.8.4	HSL で指定する .....	119
2.8.5	透過色を含む色を指定する.....	119
2.8.6	PDF 出力時の透明の変換を制御する .....	120
2.8.7	PDF 出力時にカラースペースを変換する .....	120
2.8.8	画像のスケール合わせを検知する.....	121
2.8.9	ラスターライズされるベクタ画像の解像度を指定する.....	121
2.8.10	PDF に直接格納できない画像形式の変換を制御する .....	122
2.8.11	画像埋め込みのパススルーを制御する.....	123
2.8.12	画像、背景画像の解像度を制御する.....	123
2.8.13	画面上での画像のアンチエイリアス処理を制御する.....	124
2.8.14	ページマスタ、ページシーケンスへ背景を指定する.....	124
2.8.15	ラスター画像のダウンサンプリングを制御する.....	124
2.8.16	SVG の既定サイズを指定する .....	126
2.8.17	背景にグラデーションを指定する.....	126
2.8.18	繰り返すグラデーションを指定する.....	127
2.8.19	EPS 画像の処理を制御する.....	127
2.8.20	ユースケース：文書全体の文字色を指定する.....	129
2.8.21	ユースケース：画像を円形にくり抜いて表示する.....	130
2.9 節	フロート.....	131
2.9.1	フロート配置の参照領域を指定する.....	132
2.9.2	フロートとアンカを制御する.....	133
2.9.3	フロート配置を指定する.....	134
2.9.4	テキストの回り込みを制御する.....	135
2.9.5	フロートのマージンを指定する.....	136
2.9.6	フロートのオフセット位置を変更する.....	137
2.10 節	注.....	139
2.10.1	合印、脚注番号を自動化する.....	139

2.10.2	注の配置を制御する.....	140
2.10.3	注の揃えを制御する.....	144
2.10.4	同じ注の表示を抑制する.....	145
2.10.5	注の並べ方を制御する.....	145
2.10.6	注とアンカの分離を抑制する.....	145
2.10.7	長い脚注を複数ページ、段に分割する.....	146
2.10.8	ユースケース：後注をページ下部に配置する.....	147
2.11 節	マーカ、相互参照.....	149
2.11.1	マーカ参照の重複を除去する.....	149
2.11.2	ページ番号の桁数を仮組で配置する.....	150
2.11.3	ページ番号を逆順に振る.....	151
2.11.4	ページ番号に段番号を表示する.....	151
2.11.5	ページ番号表示とは別に物理的なページ数を得る.....	152
2.11.6	相対ページ番号を得る.....	152
2.11.7	ページ番号の前置、後置文字列を抑制する.....	153
2.11.8	索引で連続するページ参照の表示を制御する.....	154
2.12 節	段落の始め方、終わり方.....	156
2.12.1	イニシャルキャップを指定する.....	156
2.12.2	イニシャルキャップヘテキストを食い込ませる.....	159
2.12.3	イニシャルキャップに先行する約物を制御する.....	160
2.12.4	段落最終行の最低文字数を指定する.....	162
2.12.5	ページ、段最初の段落先頭のインデントを指定する.....	163
2.12.6	先頭行の揃えを制御する.....	165
2.12.7	最終行の均等割り付けを制御する.....	166
2.13 節	行分割、禁則制御.....	168
2.13.1	欧文の行分割を制御する.....	168
2.13.2	単語の分割規則を制御する.....	171
2.13.3	分割禁止語を登録する.....	172

2.13.4	条件によってハイフネーションを抑制する.....	173
2.13.5	ハイフネーション情報を制御する.....	174
2.13.6	ハイフネーション例外情報を追加する.....	176
2.13.7	ハイフネーション前後の最小文字数を制御する.....	177
2.13.8	大文字の語のハイフネーション処理を制御する.....	178
2.13.9	CJK の禁則を制御する .....	178
2.13.10	ポーランド、ハンガリー、チェコ語の禁則を指定する.....	180
2.13.11	SOLIDUS 「/」 (U+002F) を伴う語の分割を抑制する .....	180
2.13.12	テキストの途中で次行以降のインデントを指定する.....	181
2.13.13	SOFT HYPHEN (U+00AD) の動作を変更する.....	182
2.14 節	段落の配置、行グリッド.....	184
2.14.1	行グリッドを制御する.....	184
2.14.2	ページ、段先頭のブロックの表示を制御する.....	185
2.15 節	はみ出し（オーバーフロー）処理.....	186
2.15.1	はみ出し時の処理を指定する.....	186
2.15.2	はみ出しのエラー報告を制御する.....	187
2.15.3	はみ出しが生じたブロックをテキストで置換する.....	188
2.15.4	はみ出しを詰め込み表示する.....	189
2.15.5	詰め込み処理方法ごとに詰め込みの限界値を指定する.....	190
2.15.6	詰め込み処理後の最終行の揃えを制御する.....	192
2.15.7	インライン進行方向のはみ出し処理を制御する.....	193
2.15.8	はみ出したブロックの揃えを指定する.....	194
2.15.9	インラインコンテナがはみ出したときの揃えを指定する.....	194
2.15.10	keep-together の上限を指定する.....	194
2.16 節	文字間のアキ、揃え制御.....	196
2.16.1	欧文で約物とのアキを制御する.....	196
2.16.2	和欧文間のアキを制御する.....	198
2.16.3	全角約物の詰め処理を指定する.....	199

2.16.4	追い込みを制御する.....	201
2.16.5	letter-spacing によるアキの端を揃える.....	202
2.16.6	文字数に合わせて CJK 文字の字取りを制御する.....	203
2.16.7	均等割り付けの処理を制御する.....	203
2.16.8	Kashida の割合を制御する.....	204
2.16.9	タブ揃えを指定する.....	205
2.16.10	TAB (U+0009) の処理を変更する.....	206
2.16.11	固定幅空白字形の処理を変更する.....	207
2.16.12	NO-BREAK SPACE (U+00A0) をアキ調整から除外する.....	208
2.17 節	文字、書式.....	209
2.17.1	テキストの書式を制御する.....	209
2.17.2	独自カウンタを定義する.....	211
2.17.3	文字の正規化と除外指定を行う.....	212
2.17.4	大文字小文字、全角文字半角文字を変換する.....	213
2.17.5	テキストを置換する.....	214
2.17.6	縦組での文字の方向を制御する.....	214
2.17.7	ユースケース：一部の文字を縦組用の文字へ置換する.....	216
2.18 節	フォント.....	218
2.18.1	字幅を指定する.....	218
2.18.2	疑似的な太字を制御する.....	218
2.18.3	既定のフォントサイズ、色を指定する.....	219
2.18.4	統合漢字などで採用される既定の CJK 言語を指定する.....	219
2.18.5	フォントサイズ調整の基準をフォント名で指定する.....	220
2.18.6	カーニング（詰め処理）を指定する.....	220
2.18.7	字形やフィーチャを指定する.....	221
2.18.8	欧文のキャピタルを指定する.....	222
2.18.9	数字、序数、分数の字形を指定する.....	224
2.18.10	リガチャ（合字）を制御する.....	224

2.18.11	装飾的な字形を指定する.....	226
2.18.12	上付き、下付き文字用の字形を指定する.....	227
2.18.13	CJK の異体字、半角・ルビ字形、繁体・簡体字を指定する .....	227
2.18.14	OpenType のフォントフィーチャを直接制御する .....	228
2.18.15	WOFF、WOFF2 フォントなどを指定する .....	228
2.18.16	オブリーク体の傾きを指定する.....	229
2.18.17	フォントファミリの記法を厳密にチェックする.....	230
2.18.18	スクリプトごとのゼネリックフォントを指定する.....	230
2.18.19	フォントのフォールバックとエラー報告を制御する.....	231
2.18.20	フォントファミリをオプション設定から置き換える.....	233
2.19 節	ルビ、圏点.....	234
2.19.1	ルビを記述する.....	234
2.19.2	ルビのフォントを修飾する.....	235
2.19.3	ルビの揃えを指定する.....	237
2.19.4	ルビの配置を指定する.....	238
2.19.5	親文字をはみ出すルビを制御する.....	238
2.19.6	小書きかなのルビを普通のかなに変換する.....	240
2.19.7	圏点を指定する.....	241
2.19.8	ユースケース：さまざまなルビを指定する.....	244
2.19.9	ユースケース：大きな親文字にルビを記述する.....	246
第 3 章	PDF 出力.....	249
3.1 節	PDF 出力.....	250
3.1.1	出力する PDF のバージョンを指定する .....	250
3.1.2	ネットワークでの表示用に最適化する.....	251
3.1.3	ファイルサイズを圧縮する.....	252
3.1.4	パスワードを指定する.....	252
3.1.5	暗号化を制御する.....	253
3.1.6	画像、テキストのコピーを禁止する.....	254

3.1.7	PDF のレイヤを指定する .....	254
3.1.8	ページを逆順に出力する.....	256
3.1.9	PDF 内部で利用される実数値の範囲を制御する .....	256
3.2 節	文書情報の制御.....	258
3.2.1	基本的な文書情報を指定する.....	259
3.2.2	文書の著者、書き手についての情報を指定する.....	259
3.2.3	作成日、修正日を指定する.....	259
3.2.4	著作権情報を指定する.....	261
3.2.5	文書情報を XMP ファイルで指定する .....	261
3.2.6	PDF を開いたときの PDF ビューアの表示を指定する .....	262
3.2.7	カスタムプロパティを指定する.....	262
3.3 節	PDF 注釈 .....	264
3.3.1	PDF 注釈の種類を指定する .....	264
3.3.2	PDF 注釈のタイトルと作成者を記述する .....	265
3.3.3	PDF 注釈の作成日と修正日を記述する .....	265
3.3.4	ファイル添付注釈を指定する.....	266
3.3.5	PDF 注釈の背景色を指定する .....	266
3.3.6	フリーテキスト注釈のボーダーを修飾する.....	266
3.3.7	PDF 注釈のアイコンを指定する .....	267
3.3.8	フリーテキスト注釈を修飾する.....	268
3.3.9	ポップアップ、ラバースタンプ注釈のサイズを指定する.....	269
3.3.10	PDF 注釈の配置を指定する .....	269
3.3.11	出力 PDF の PDF 注釈に対し可能な操作を制御する .....	270
3.3.12	閲覧するときの PDF 注釈の開閉状態を指定する .....	271
3.3.13	PDF 注釈のテキストを記述する .....	271
3.3.14	PDF を埋め込むときに保持する PDF 注釈を指定する .....	272
3.3.15	ユースケース：表組に元データを添付する.....	272
3.4 節	リンク制御.....	274

3.4.1	外部リンクの動作を指定する.....	274
3.4.2	リンク先になるときの開かれ方（宛先形式）を指定する.....	275
3.4.3	ユースケース：フォームフィールドをリンクにする.....	276
3.5 節	多分冊出力制御.....	277
3.5.1	分冊箇所を指定する.....	277
3.5.2	分冊出力される PDF のファイル名を指定する.....	277
3.5.3	分冊での文書情報を一括で指定する・個別に指定する.....	278
3.5.4	分冊の開始番号を指定する.....	278
3.5.5	分冊でのしおり情報を記述する.....	278
3.5.6	多分冊でしおりに外部リンク、内部リンクを指定する.....	280
3.5.7	分冊されたしおりのテキストを修飾する.....	281
3.5.8	分冊でのしおりの展開を制御する.....	281
3.5.9	ユースケース：多分冊用の目次、索引を作成する.....	282
3.6 節	マルチメディアの埋め込み.....	286
3.6.1	埋め込むメディア形式を明示する・埋め込みを制御する.....	286
3.6.2	代替のポスター画像を指定する.....	288
3.6.3	メディア再生時の音量を制御する.....	288
3.6.4	マルチメディア注釈の再生を制御する.....	289
3.6.5	リッチメディア注釈で別ウインドウのサイズを指定する.....	290
3.6.6	リッチメディア注釈のスキンを制御する.....	290
3.6.7	リッチメディア注釈の有効化タイミングを制御する.....	291
3.6.8	Flash メディアを制御する.....	292
3.6.9	ユースケース：メディア代替のポスター画像を印刷する.....	293
3.7 節	PDF 対話フォーム（AcroForm）.....	294
3.7.1	フォームフィールドを作成する.....	294
3.7.2	各フィールドの操作権限、状態を指定する.....	296
3.7.3	テキストフィールドを指定する.....	298
3.7.4	プッシュボタンを指定する.....	302

3.7.5	選択フィールドを指定する.....	305
3.7.6	電子署名フィールドを制御する.....	309
3.7.7	フォームアクションを指定する.....	310
3.7.8	PDF 全体のフォームへの入力と署名を不許可にする.....	312
3.8 節	タグ付き PDF を出力する.....	314
3.8.1	PDF タグを記述、マッピングする.....	314
3.8.2	PDF 2.0 以降に対応したタグ付き PDF かチェックする.....	315
3.8.3	画像、リンクの代替テキストを記述する.....	316
3.8.4	構造要素にスクリプトを指定する.....	317
3.8.5	表組でヘッダとセルの対応を明示する.....	317
3.8.6	表組の要約を記述する.....	318
3.8.7	正確なテキスト情報を含む PDF を出力する.....	318
3.8.8	スクリーンリーダーのテキストアクセスを制限する.....	319
第 4 章	その他の拡張.....	321
4.1 節	印刷.....	322
4.1.1	用紙の端から仕上り寸法までの距離を指定する.....	322
4.1.2	塗り足しの幅を指定する.....	323
4.1.3	印刷マーク（トンボ）を表示する.....	323
4.1.4	オーバープリントを指定する.....	326
4.1.5	PDF で印刷の不許可、印刷可能な解像度を指定する.....	328
4.1.6	Windows 版の印刷出力でプリンタの動作を指定する.....	328
4.1.7	ユースケース：背表紙を含む表紙にトンボを付加する.....	330
4.1.8	ユースケース：塗り足しを含めたツメを配置する.....	334
4.2 節	ウォーターマーク（透かし）を指定する.....	337
4.3 節	文書内の相対 URI の基点を指定する.....	340
4.4 節	オプション設定を FO に記述する.....	341
4.5 節	エラーメッセージ表示を抑制する.....	342

4.6 節 バーコードについての拡張.....	343
本書中の拡張仕様一覧.....	346

# 第1章

## ショウケース

## 1.1 節 文書の種類と拡張のユースケース

AH XSL Formatter は XSL 1.1 仕様の多くに準拠しています。また XSL 2.0 のドラフト仕様や CSS3 で議論されている仕様、ほかユーザの要望をベースとした多くの拡張仕様を備えています。代表的な文書種類と、その組版に利用可能な拡張の例について紹介します。

### 雑誌、パンフレット

写真画像の解像度制御や複雑な段組、テキストやブロックの自在な修飾が求められます。

- スポットカラー指定
- 複数種類の段組
- テキストのオーバーフローへの細かな制御

### 文芸

伝記や小説などでは、本文の細やかな配置、ルビや注などの補足要素、特定の箇所へのイラスト配置などが求められます。

- 字間、行間、段落の細かな制御
- イニシャルレター
- ルビ、圈点
- 細かなフロート配置

### 論文誌

論文誌では決まったページ数に可能な限り文量を詰めたり、豊富な記号類や美しい数式組版などが求められます。

- 字間、行間、段落の細かな制御
- 異体字制御
- 多数のスキプトのサポート
- MathML による数式組版
- SVG、CGM などベクタ画像の挿入、出力
- 細かなフロート配置

### 帳票

PDF ビューアから入力可能なフォームの付加された帳票を自在なレイアウトで出力可能です。

- PDF 対話フォーム
- ブロック、テキストの多様な修飾

## ハードウェア、ソフトウェアマニュアル

マニュアル文書では、長大なページ数やさまざまな図表に対応した組版が求められます。

- 語彙の分割制御
- SVG、CGM などベクタ画像の挿入、出力
- 細かなフロート配置
- 1 ページに収まらない表
- サイズの異なる図表、ページ挿入
- 目次、索引の表記
- 多分冊出力制御
- 動画、ファイルの埋め込み、添付

この章では、AH XSL Formatter 拡張仕様を利用して組版した文書の例を紹介します。

## 1.2 節 観光パンフレット風のページデザイン

大きな背景画像を使用しつつ、テキストの可読性を確保したページを作成します。

見開きの背景ページと袋文字の見出し、透過を含む背景の本文区画、段組に対応したフロートによる画像を含みます [図 1.2-1] (p.22)。



図 1.2-1 観光パンフレット風のページデザイン

### 見開きのページレイアウトを指定する

```
<!-- 見開きページマスタが参照する単純ページマスタ -->
<fo:layout-master-set>
  <!-- 左ページマスタ -->
  <fo:simple-page-master master-name="left"
    page-width="210mm" page-height="295mm"
    margin-left="0mm">
    <fo:region-before region-name="leftSection" extent="20mm"/>
    <fo:region-body
      margin-top="65mm" margin-bottom="20mm"
      margin-left="20mm" margin-right="30mm"
      padding="4mm" column-count="2" column-gap="24pt"
      .../>
  </fo:simple-page-master>
```

```

<!-- 右ページマスタ -->
<fo:simple-page-master master-name="right"
  page-width="210mm" page-height="295mm"
  margin-right="0mm">
  <fo:region-before region-name="rightSection" extent="20mm" />
  <fo:region-body
    margin-top="65mm" margin-bottom="20mm"
    margin-left="30mm" margin-right="20mm"
    padding="4mm" column-count="2" column-gap="24pt"
    .../>
</fo:simple-page-master>

<!-- 見開きページマスタ -->
<axf:spread-page-master master-name="spreadpm"
  left-page-master-reference="left"
  right-page-master-reference="right" ...>
  ...
</axf:spread-page-master>

<!-- ページシーケンスが参照するマスタ -->
<fo:page-sequence-master master-name="psm">
  <fo:repeatable-page-master-reference
    master-reference="spreadpm" />
  <fo:repeatable-page-master-alternatives>
    <fo:conditional-page-master-reference
      master-reference="left" odd-or-even="even"/>
    <fo:conditional-page-master-reference
      master-reference="right" odd-or-even="odd"/>
  </fo:repeatable-page-master-alternatives>
</fo:page-sequence-master>
</fo:layout-master-set>
<!-- PDFビューアに見開き表示を指定 -->
<fo:declarations>
  <axf:document-info name="pagelayout" value="TwoPageLeft" />
</fo:declarations>
<fo:page-sequence master-reference="psm"
  initial-page-number="2" force-page-count="end-on-odd">
  ...
</fo:page-sequence>

```

左ページと右ページでヘッダのスタイルを変更するため、left-page-master-reference、right-page-master-reference プロパティにそれぞれ別のページマスタを指定しています。

見開き区画と重ならないよう、本文区画上側のマージンを広めに指定しています。

見開きページだけですべてのページを構成可能とは限らないため、ページシーケンスマスタに見開きページと、左右のページマスタをそれぞれ指定しています。ページシーケンスはこのページシーケンスマスタを参照します。

PDF ビューアで開いたとき見開きで表示されるように、`<axf:document-info>` 要素で `name="pagelayout"`、`value="TwoPageLeft"` を指定しました。

## 見開きの背景画像を指定する

```
<!-- 見開きページマスタ -->
<fo:layout-master-set>
  ...
  <axf:spread-page-master master-name="spreadpm"
    left-page-master-reference="left"
    right-page-master-reference="right"
    background-image="showcase_brochure_2.jpeg"
    background-content-height="scale-to-fit"
    background-position-vertical="bottom"
    axf:background-image-resolution="normal"
    background-repeat="no-repeat">
    ...
  </axf:spread-page-master>
  ...
</fo:layout-master-set>
```

`<axf:spread-page-master>` へ `background-image` プロパティで画像を指定することで、見開きページの背景とします。広大な風景写真を使うパンフレットなどに有用です。

また、背景画像の解像度を `axf:background-image-resolution` プロパティへ指定しています。

## 見開きの見出しを袋文字にする

```
<!-- 見開き区画 -->
<fo:layout-master-set>
  ...
  <axf:spread-page-master master-name="spreadpm" ...>
    <axf:spread-region region-name="chapter"
      margin="20mm" margin-bottom="250mm" />
    ...
  </axf:spread-page-master>
  ...
</fo:layout-master-set>
<fo:page-sequence master-reference="psm"
  <!-- 見開き区画のフロー -->
  <fo:flow flow-name="chapter">
    <fo:block-container height="100%"
      axf:text-stroke="2pt white"
      color="cmyk(80%, 30%, 10%, 70%)"
      font-size="86pt" font-weight="bold"
      text-align="center"
      font-family="sans-serif">
      <fo:block padding="3mm"
```

```

    >海でXSL-FO体験</fo:block>
  </fo:block-container>
</fo:flow>
...
</fo:page-sequence>

```

見出しを見開き要素として本文の上に配置しました。flow-name に見開き区画を指定しています。テキストが背景で見づらくなってしまうため、axf:text-stroke プロパティで文字に輪郭線を付加し袋文字にしました。

## 本文区画の背景色に透過度を設定する

```

<!-- 本文区画の背景色に透過度を設定する -->
<fo:simple-page-master master-name="left"
  ...>
  ...
  <fo:region-body ...
    background-color="rgba(200, 200, 180,0.8)" />
</fo:simple-page-master>

```

背景画像を映しつつテキストの可読性を確保することを考えます。袋文字や、背景と区別しやすい文字色の指定の他に、薄い色をテキストの下に置く方法が考えられます。<fo:region-body>の background-color プロパティへ、rgba() で半透明の色を指定しています。

## 画像を下側に揃える

```

<!-- 画像を含むフロート -->
<fo:float axf:float-y="bottom">
  <fo:block text-indent="0" text-align="center"
    space-before="1.0em" space-after="1.0em">
    <fo:external-graphic
      src="showcase_brochure_1.png"
      width="48mm" content-width="scale-to-fit"/>
  </fo:block>
  <fo:block text-indent="0" text-align="center"
    space-before="1.0em" space-after="1.0em">
    <fo:external-graphic src="showcase_brochure_3.png"
      width="48mm" content-width="scale-to-fit" />
  </fo:block>
</fo:float>

```

axf:float-y プロパティへ bottom を指定することで、フロート配置が下側に揃えられました。

この FO では画像をひとつのフロート内にまとめ、それぞれのブロックに space-\* プロパティでアキを指定しています。画像をそれぞれ独立したフロートの子に記述すれば、axf:float-float-margin-\* プロパティなどでも画像間のアキを調整可能です。

## 1.3 節 縦組小説風のページデザイン

和文で縦組を中心にしたページを作成します。

複数の書字方向、行グリッド、ルビ、傍注、約物のアキ、数字の書式などを組み合わせます [図 1.3-1] (p.26)。

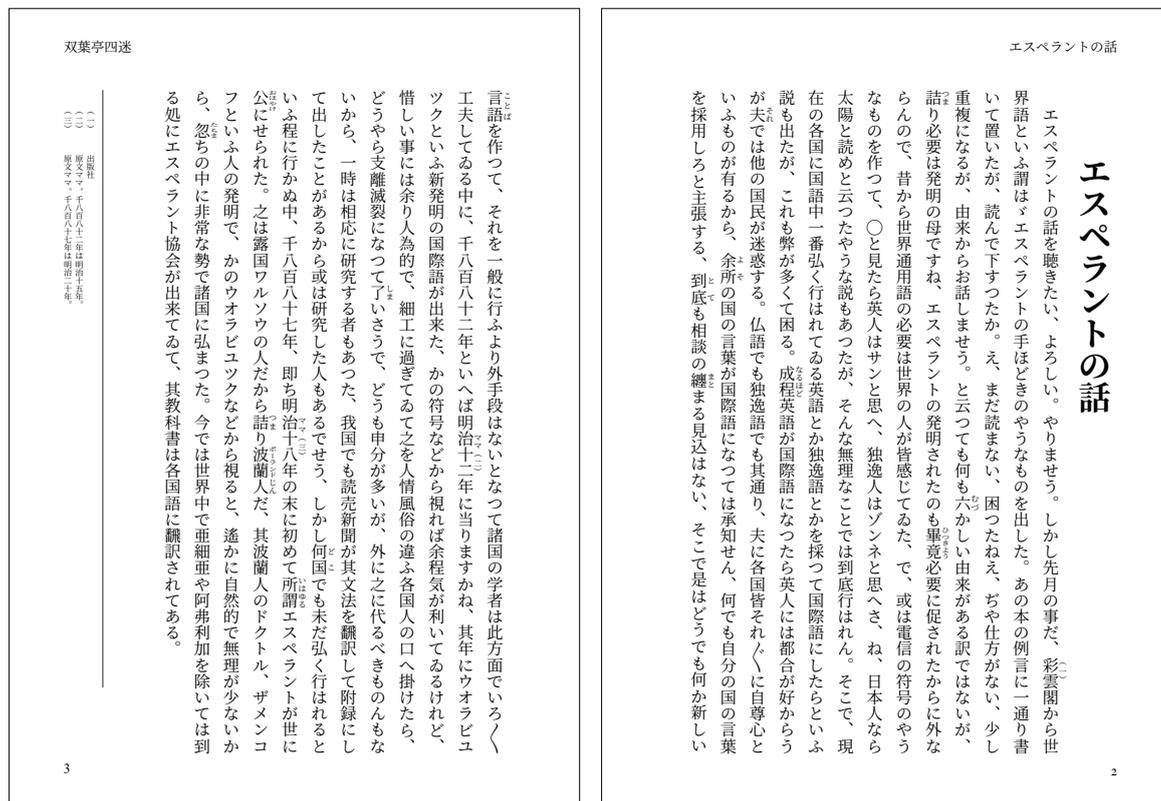


図 1.3-1 縦組小説風のページデザイン

### A6 縦組の基本版面を指定する

```
<!-- 左右のページマスタ -->
<fo:root
  font-size="12q" line-height="1.8" ... >
  <fo:layout-master-set>
    <fo:simple-page-master master-name="left"
      size="A6 portrait" writing-mode="tb-rl">
      <fo:region-start region-name="left-header"
        margin-left="5mm" extent="12mm" />
      <fo:region-end region-name="left-footer"
        extent="10mm"/>
      <fo:region-body margin-top="15mm" margin-bottom="10mm"
        margin-left="10mm" margin-right="14mm"
        axf:footnote-position="odd-page"/>
```

```

</fo:simple-page-master>
<fo:simple-page-master master-name="right"
  size="A6 portrait" writing-mode="tb-rl">
  <fo:region-start region-name="right-header"
    margin-right="5mm" extent="12mm" />
  <fo:region-end region-name="right-footer"
    extent="9mm"/>
  <fo:region-body margin-top="15mm" margin-bottom="10mm"
    margin-left="14mm" margin-right="10mm"
    axf:footnote-position="odd-page"/>
</fo:simple-page-master>
<fo:page-sequence-master master-name="main">
  <fo:repeatable-page-master-alternatives>
    <fo:conditional-page-master-reference
      master-reference="left" odd-or-even="odd" />
    <fo:conditional-page-master-reference
      master-reference="right" odd-or-even="even" />
  </fo:repeatable-page-master-alternatives>
</fo:page-sequence-master>
</fo:layout-master-set>
...
</fo:root>

```

ページサイズ指定のショートハンドである `size` プロパティに、和文縦組の文庫本を意識し A6 判を指定しています。

和文ではフォントサイズの単位に級数 (q) を使うと版面計算がしやすくなることもあります。この FO ではルビや傍注の合印など、行の間に表示されるオブジェクトが頻出するため、行間を大きめにしています。<fo:root>に本文用のフォントサイズと行の高さを指定しています。フロー内の本文箇所では `axf:baseline-grid="root"` として、この値を行グリッドに適用します。

左右で柱の置き方、マージンの取り方を変更するため、ふたつのページマスタを用意し、ページシーケンスマスタに割り当てを記述しています。

## 傍注の合印を本文の横に表示する

```

...
しかし先月の事だ、<fo:inline-container
  alignment-baseline="central"
  block-progression-dimension="0rem"
  text-indent="0" line-stacking-strategy="font-height"
  start-indent="0" end-indent="0" text-align="center" ...>
<fo:block
  axf:baseline-block-snap="before"
  axf:baseline-grid="none" text-indent="0pt"
  font-size="0.5rem" line-height="1"
  space-before.conditionality="retain"
  space-before="-1rem" >
<fo:footnote>

```

```

<fo:inline alignment-baseline="before-edge"
  font-size="0.5rem"> ( <axf:footnote-number xml:id="saiunkaku" /> )
</fo:inline>...
</fo:footnote>
</fo:block>
</fo:inline-container>...

```

合印はインライン領域を取るため、ベースラインを移動しても合印の幅（高さ）のアキが前後のテキスト間に生じます。アキを生じさせず合印を本文の横へ表示するために、インラインコンテナを利用しています。

## 合印が版面をはみ出さないように調整する

```

<!-- 合印が版面をはみ出さないように調整する -->
しかし先月の事だ、<fo:inline-container ...
  axf:inline-overflow-align="end">
...
</fo:inline-container>...

```

インラインコンテナで本文の横に配置した合印ですが、このインラインコンテナの箇所で行分割が生じるとき、本文の領域をはみ出す可能性があります。

`axf:inline-overflow-align="end"`を指定し、行分割で合印がはみ出す可能性がある箇所ではページ下部側で本文の領域に収まるように調整します。

## 傍注で独立した行グリッドを指定する

```

<!-- 傍注で独立した行グリッド -->
<fo:footnote>
...
<fo:footnote-body>
  <fo:list-block ...
    axf:baseline-grid="new" axf:baseline-block-snap="after">
    ...
  </fo:list-block>
</fo:footnote-body>
</fo:footnote>

```

見出し行や著者名など、サイズが異なる要素を含む行の配置を合わせるために行グリッドを設定しています。

注のように、フォントサイズや行の高さが異なった段落内では、本文を基準にした行グリッドに揃えるのは望ましくありません。`axf:baseline-grid="new"`として別の行グリッドを設定しています。

傍注の合印のためにインラインコンテナ中で `fo:block` を利用しています。

本文の行や見出しはルート要素に指定した行グリッドに合わせられます。しかし、フォントサイズや行の高さが異なる段落を持つ傍注では、本文と同じ行グリッドでは不自然な行間が発生しま

す。そこで、行グリッドを無効化するか、別の行グリッドを設定することで不自然な行間を回避します。ここでは `axf:baseline-grid="new"` で傍注に新たな行グリッドを設定しています。

注の段落全体を本文の行グリッドと調整するときは `axf:baseline-block-snap` プロパティで制御します。`<fo:footnote-body>` の子のブロックで指定を行います。

## 入力文字の変換

```
<!-- -->
<fo:flow flow-name="xsl-region-body">
  <fo:wrapper ...
    axf:number-transform="kansuji" axf:kansuji-style="readable"
    text-transform="fullsize-kana" font-variant="jis04" >
    ...
  </fo:wrapper>
</fo:flow>
```

元のテキスト中のひらがな、カタカナの小文字を大文字に変更するため、`text-transform="fullsize-kana"` を指定します。

本文の数字表記を漢数字にするため、`axf:number-transform="kansuji"` を指定します。漢数字の書式は「十」や「〇」を表示するため `axf:kansuji-style="readable"` を指定します。

数字の書式変換は柱のページ番号などには適用しないとして、`<fo:flow>` 直下の `<fo:wrapper>` に指定しています。

## 入力文字の置換を一部に適用する

```
<!-- 特定のテキストを置換する -->
<fo:block ...
  axf:text-replace="はば は&#x309E; それぞれ それ&#x3032; いろいろ いろ&#x3031;"
  >エスペラントの話を聴きたい、よろしい。やりませう。...</fo:block>
```

`axf:text-replace` により、「くの字点」やひらがな繰り返し記号などキーボードから直接入力することが難しい文字を、元テキストでは使用せず組版時の置換が可能です。

## 和文約物のアキを調整する

```
<!-- 和文約物のアキを調整する -->
<fo:wrapper ... axf:punctuation-trim="adjacent">
  <fo:block ...>エスペラントの話...</fo:block>
</fo:wrapper>
```

和文約物のアキを縮めるため、`axf:punctuation-trim` プロパティを指定しています。ここでは `adjacent` の指定で隣接する全角括弧類を詰め処理しています。

## 傍注の罫線を start 側に揃える

```
<!-- 傍注の罫線を指定する -->
<fo:static-content flow-name="xsl-footnote-separator">
  <fo:block text-align="left"
    line-height="1">
    <fo:leader leader-pattern="rule" leader-length="90%"
      leader-alignment="start"
      rule-thickness="0.3pt" rule-style="solid" />
  </fo:block>
</fo:static-content>
```

この FO では、傍注は奇数ページ本文区画の after 側です。本文との間の罫線は `xsl-footnote-separator` 領域に記述します。

罫線を `<fo:leader>` で引くとき、長さは `leader-length` プロパティへ指定します。90%で本文区画の9割となるよう指定しました。 `leader-alignment="start"` とし、リーダーの端を `start` に揃えています。

## 親文字をはみ出すルビを三分ルビにする

```
<!-- 親文字をはみ出すルビを三分ルビにする -->
...発明されたのも<axf:ruby
  ruby-condense="66.6%"
  ruby-font-size="0.5em"
  ruby-minimum-font-size="0.3rem">
<axf:ruby-base>畢竟</axf:ruby-base>
<axf:ruby-text
  font-variant="ruby">ひっきょう</axf:ruby-text>
</axf:ruby>...
```

単純にルビ幅を狭めるのであれば `axf:ruby-font-stretch` を指定しますが、「親文字をはみ出すときのみ幅を狭めたい」というときには `axf:ruby-condense` を指定します。三分ルビは親文字の1/3の高さ（幅）ですが、`axf:ruby-condense` はルビに対しての比率指定なので、ルビのフォントサイズを考慮します。

この FO には登場しませんが、`axf:ruby-condense` を行っても親文字をはみ出すルビについては `axf:ruby-overhang`、`axf:ruby-limit-overhang` プロパティで隣の文字への掛かり方や量の制御が可能です。

## 1.4 節 自由配置の多い雑誌風のページデザイン

ブロックコンテナの絶対配置や修飾を組み合わせることで、DTP ソフトで制作する雑誌のようなデザインも可能です [図 1.4-1] (p.31)。

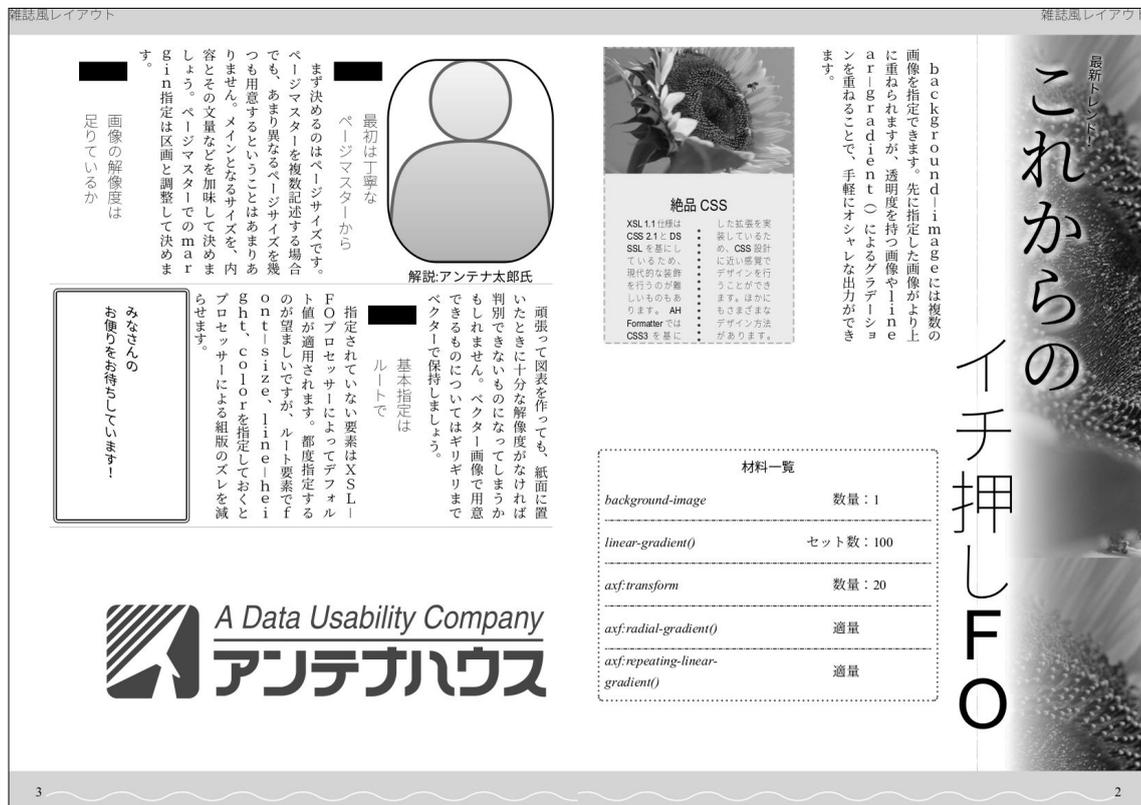


図 1.4-1 自由配置の多い雑誌風のページデザイン

### 無地から画像へのグラデーションをする背景を指定する

```
<!-- ページへの背景指定 -->
<fo:simple-page-master master-name="right"
  size="A4 portrait"
  background-image="linear-gradient(to right,cmyka(0,0,0,0), 1.0) 75% ,
  rgba(0,0,0,.0) 100%), url(/background.jpg) ">
  ...
</fo:simple-page-master>
```

ページマスターに `background-image` プロパティで背景画像を指定し、ページ全体に背景を適用しています。`background-image` プロパティには複数の画像が指定可能で、先に指定したものがより上に描画されます。`linear-gradient()` によるグラデーションを透明度のある色で重ねました。75% まで は完全な白色、そこから透明へグラデーションをかけて、下の画像を表示しています。

## 左右で異なるレイアウトを指定する

```
<!-- 左ページのレイアウト -->
<fo:simple-page-master master-name="left"
  size="A4 portrait">
  <fo:region-start extent="10mm" />
  <fo:region-end extent="15mm" />
  <fo:region-body margin-top="15mm"
    margin-bottom="18mm" margin-left="15mm" margin-right="10mm"
    column-count="3" column-gap="6mm" axf:column-rule-style="solid"
    axf:column-rule-color="cmyk(0.3,0.3,0,0.0)"
    axf:column-rule-width="0.3pt"/>
</fo:simple-page-master>
```

段組の変更などであれば複数の本文区画を用意し、`flow-name` の指定で切り換えた方がページマスタの数は少なくなります。

この FO では、最初のページはブロックコンテナの絶対配置でオブジェクトを配置し、次のページから流し込みが可能な内容を続けています。

## 指定した領域をはみ出すテキストを詰め込む

```
<!-- 右ページ左上の横組-->
<fo:block-container width="14em" height="22rem">
  <fo:block-container height="10em" ...>
    ...
  </fo:block-container>
  ...
  <fo:block-container overflow="condense"
    axf:overflow-condense="font-stretch"
    height="22rem - 10em"
    axf:overflow-limit-block="22rem - 10em - 16pt - 2pt - 0.5pt"
    ...
    margin="4mm">...

  <fo:block ...>絶品CSS</fo:block>
  <fo:block ...>XSL 1.1仕様はCSS 2.1とDSSSLを基にして...
  </fo:block>
</fo:block-container>
</fo:block-container>
```

高さ、幅まで指定したブロックコンテナでは、子要素の内容がそのサイズを超過することがあります。このはみ出した内容の制御方法はいくつかありますが、フォント幅を圧縮して詰め込むよう指定しました。

## 見出しを修飾する

```
<!-- 見出しの修飾 -->
<fo:block-container margin-top="0.5em"
  absolute-position="absolute" top="0mm" right="0"
  height="100vh">
  <fo:wrapper color="cmyk(80, 10,10, 0)"
    font-family="sans-serif">
    <fo:block
      axf:transform="rotate(5)"
      axf:transform-origin="top"
      text-shadow="1pt 1pt 3pt cmyk(0,0,0,0.0),
        -1pt -1pt 3pt cmyk(0,0,0,0)"

      >最新トレンド!</fo:block>
    <fo:block font-size="72pt"
      line-height="1" font-weight="100"
      text-shadow="4pt 4pt 5pt cmyk(0,0,0,0.0),
        -4pt -4pt 5pt cmyk(0,0,0,0)"
      axf:tab-stops="eol"
      axf:text-orientation="upright"
      >
      <fo:inline
        font-family="serif" font-weight="normal"
        >これからの</fo:inline
        ><fo:inline axf:indent-here="-1em"
        /><axf:tab/>
        イチ押しFO
      </fo:block>
    </fo:wrapper>
  </fo:block-container>
```

メインの見出しの横に、小さめのサイズのテキストを傾けて表示しています。axf:transform プロパティによる変形の基点は axf:transform-origin プロパティで指定可能です。この FO では変形の基点を top top に変更しています。

この見出しでは text-shadow プロパティでテキストを強調しています。背景と同じ白色で文字の周りをぼかしています。

## ボーダーの点線の下に背景色を表示する

```
<!-- 背景の適用範囲の変更 -->
<fo:block-container axf:background-clip="border-box"
  writing-mode="lr-tb"
  axf:border-radius="3pt"
  border="1.5pt dashed cmyk(10, 0.4, 0.4, 0)"
```

```
background-color="cmyk(0.0, 0.071, 0.298,0.0)"

absolute-position="absolute"
width="14em" height="22rem">
</fo:block-container>
```

axf:background-clip="border-box"を指定し、ボーダーの破線の下に、このブロックの背景を表示させています。

## 縦組のテキスト中の欧文を全角文字に変換する

```
<!-- writing-mode="tb-rl"-->
<fo:wrapper ...
  text-transform="fullwidth-if-vertical">
  ...
  <fo:block>background-imageには複数の画像を指定できます。...
  </fo:block>
</fo:wrapper>
```

テキストを縦組にするとき、欧文の単語を全角文字に置き換えて正立させるとします。text-transform="fullwidth-if-vertical"はテキストが縦組のとき欧文を全角文字に変換します。

## 1.5 節 欧文雑誌風のページデザイン

複数の段組と段抜きのフロートを組み合わせたページを作成します。

段間罫、フロート配置の細かな指定と段抜き、行グリッド、イニシャルキャップ、装飾行の表示抑制、ブロックコンテナによる段組を組み合わせています [図 1.5-1] (p.35)。

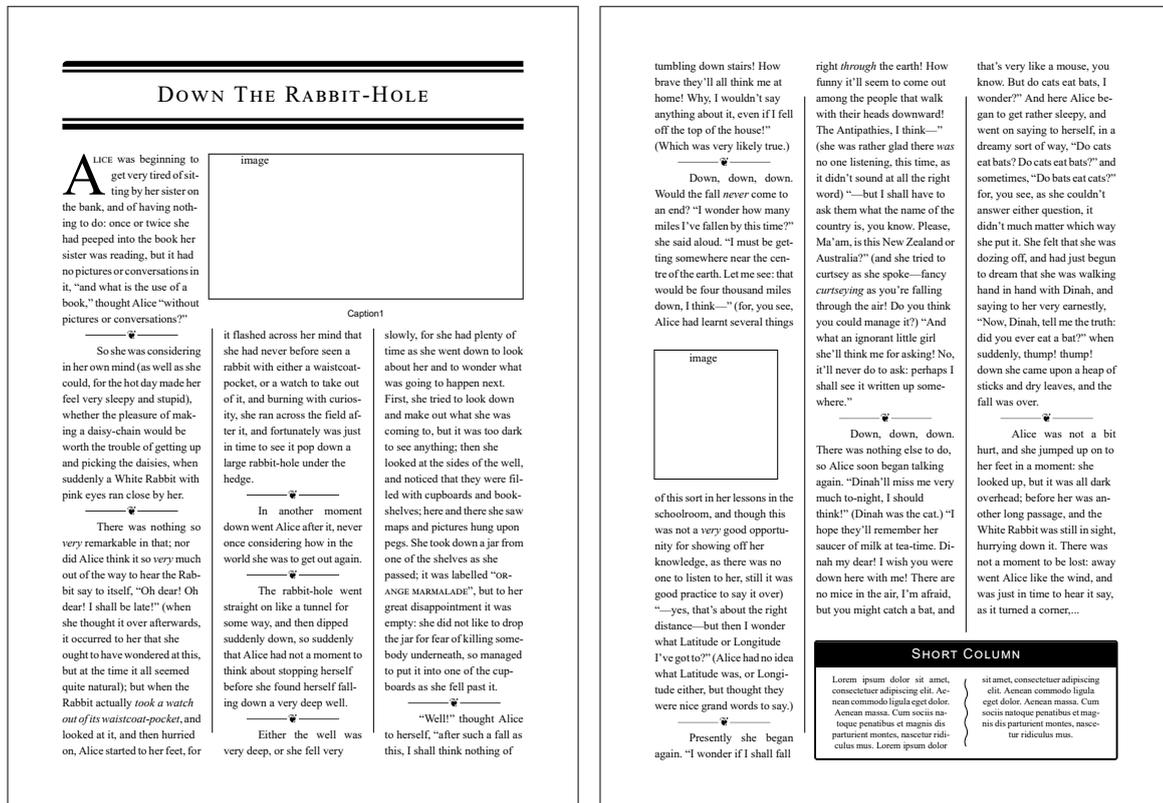


図 1.5-1 欧文雑誌風のページデザイン

### 段組と段間罫を指定する

```
<!-- 本文区画に段組と段間罫を指定する -->
<fo:root xml:lang="en" ... axf:line-break="line">
...
<fo:layout-master-set>
  <fo:simple-page-master ...>
    <fo:region-body ...
      column-count="3" column-gap="8mm"
      axf:column-rule-style="solid"
      axf:column-rule-color="rgb-icc(0, 0, 0, #CMYK,
        0.0, 0.0, 0.0, 1.0)"
      axf:column-rule-width="1pt" axf:column-rule-length="90%"
      axf:column-rule-align="center" axf:column-rule-display="all"/>
    ...
```

```
</fo:simple-page-master ...>
</fo:layout-master-set>
...
</fo:root>
```

3 段組にしつつ行長を確保するため、`column-gap="8mm"`と狭く指定し、`axf:column-rule-*`によって段間罫を指定し、段間を強調しています。段間罫を上端から下端まで引いてしまうと区切られた印象が強くなりすぎるため、`axf:column-rule-length="90%"`とし、端を空けています。`axf:column-rule-align="center"`で均等割り付けにしています。`axf:column-rule-display="all"`の指定で、本文がない段にも段間罫を引きます。

## 見出しを修飾する

```
<!-- 段抜き見出しの修飾 -->
<fo:block span="all"
  border-top-style="double" border-bottom-style="double"
  border-color="black" border-width="10pt"
  axf:border-top-left-radius="5pt"
  axf:border-double-thickness="4pt 2 1"
  space-after="1rem" space-after.conditionality="retain"
  letter-spacing="2pt" font-size="2rem" dominant-baseline="central"
  font-variant="small-caps"
  text-align="center">Down The Rabbit-Hole</fo:block>
```

段抜きで見出しを配置しています。ブロックの修飾、次の要素とのアキ、見出し文字列のスタイル指定を行っています。

二重線は `border-style="double"` を指定します。二重線の調整は `axf:border-double-thickness` プロパティで指定します。`axf:border-top-left-radius` プロパティによって、角丸を左上のみに指定しました。

見出しと本文が近過ぎないように、`space-after`、`space-after.conditionality="retain"` を指定しました。

見出し文は `font-variant="small-caps"` でスモールキャピタルにします。フォントが字形を持つのであれば `titling-caps` を指定してもよいでしょう。中央揃えでフォントサイズを本文サイズの3倍とした `letter-spacing="1.2pt"` で字間を取っています。

## 本文段落の指定

```
<!-- 本文段落のレイアウト指定 -->
<fo:root ...
  font-size="12pt" line-height="1.4">
  ...
  <fo:page-sequence ...>
    <fo:wrapper
      text-indent="3rem" text-align="left"
```

```

    axf:text-align-first="end"
    axf:baseline-grid="root" axf:avoid-widow-words="true"
  hyphenate="true"
  axf:hyphenate-hyphenated-word="false"
    hyphenation-remain-character-count="3"
    hyphenation-push-character-count="3"
    line-stacking-strategy="font-height"
    axf:ligature-mode="latin">
  <fo:block>...</fo:block>
<fo:page-sequence ...>
</fo:root>

```

本文段落に共通するスタイルを<fo:wrapper>に記述しています。文書の大半が本文で構成されるため、axf:baseline-grid="root"で、行グリッドをルート要素へ指定した本文のフォントサイズと行の高さに揃えます。

段落自体は text-align="left" を指定した左寄せのラグ組です。段落開始行のインデントに text-indent="3rem" を指定しています。このインデントを強調するため、段落開始行は axf:text-align-first="justify" によりインデントの後ろから行末まで均等割り付けをしており、右端にアキを持ちません。子のフロートなどで1行のテキストがあるとき、ここで指定されている axf:text-align-first="justify" を継承してしまうと text-align="center" などよりも優先されます。

行長が短いため、hyphenate="true" を指定して単語途中の行分割を許容しています。元からハイフンの含まれる語は axf:hyphenate-hyphenated-word="false" の指定によりハイフネーションを行いません。hyphenation-push-character-count、hyphenation-remain-character-count プロパティは AH XSL Formatter による拡張の auto を指定しています。この指定でハイフネーション可能な字数は、2文字かハイフネーション例外辞書の指定値を利用します。

axf:ligature-mode="latin" で欧文のリガチャが可能などときは行うように指定しています。

## 注

複数段組では行の長さをあまり取れないため、axf:line-break="bpi1" による段落単位でのテキスト配置最適化の効果は限定的となります。

## ドロップキャップを指定する

```

<!-- 最初の段落にドロップキャップを指定する -->
<fo:wrapper text-indent="3rem" ...>
  <fo:block axf:baseline-block-snap="center"
    axf:initial-letters="3 drop"
    text-indent="0rem"
    axf:initial-letters-first-line-head-height="cap-height"
    axf:initial-letters-end-indent="-1rem 0.5rem"
    axf:text-align-first="justify">
  <fo:inline
    font-variant="small-caps">Alice</fo:inline> was beginning to ...

```

```
</fo:block>...
</fo:wrapper>
```

axf:initial-letters="3 drop"で3行分の高さのドロップキャップを指定しています。<fo:wrapper>にtext-indent="3rem"を指定しているため子のブロックでは本文の段落開始行がインデントされますが、この段落ではドロップキャップ指定だけにするため、text-indent="0rem"を指定します。

最初の単語をスモールキャピタルにしましたが、続く語は通常の小文字です。ドロップキャップの高さを最初の行のcap-heightに合わせるため、axf:initial-letters-first-line-head-height="cap-height"を指定しました。

ドロップキャップされた「A」の字形と、先頭行の見た目上のアキが広いと、axf:initial-letters-end-indent="-1rem 0.5rem"で最初の行を1rem左に詰め、2行目以降は逆に0.5rem空けました。この調整の分、最初の行末が左に寄りすぎてみえるためaxf:text-align-first="justify"で均等割り付けにしています。

## 本文段落間の装飾

```
<!-- 段落間に装飾ブロックを挿入する -->
<fo:block>...large rabbit-hole under the hedge.
</fo:block>
<fo:block keep-with-previous="always" text-align="center"
  text-indent="0" axf:text-align-first="center"
  font-family="fantasy">
  <fo:leader leader-pattern="rule"
    leader-length="50% - 0.5em - 2em"
    rule-thickness="0.3pt"
    baseline-shift="0.3em"
    rule-style="solid" color="k100"/>
  &#x2766;
  <fo:leader ... />
</fo:block>
<fo:block>In another...
</fo:block>
```

段落間に装飾の行を挟むことにします。通常のテキストは配置しないため、この<fo:block>のtext-indentプロパティの値をリセットしています。

keep-with-previous="always"を指定することで、常に上の段落の終了位置に配置され、ページや段の先頭にこの装飾の行が現れないようにします。

### 注

ページ開始時の先頭要素のみ表示を抑制するのであれば、axf:suppress-if-first-on-page="true"をそのブロックに指定することで、ブロックのページ先頭での表示を抑制します。

## フロートと見出しの位置を調整する

```
<!-- フロート位置の調整 -->
<fo:float
  axf:float-x="inside" axf:float-y="before"
  axf:float-wrap="wrap" axf:float-margin-x="0.8rem 0"
  axf:float-margin-y="0.5rlh"
  axf:float-offset-y="6rem + 1.5rlh"
  axf:float-reference="multicol">
  ...
<fo:block-container
  axf:baseline-block-snap="before" axf:baseline-grid="none"
  ... width="4gr - 0.8rem">
  <fo:block text-indent="0" ...
    text-align="center"
    font-family="sans-serif">Caption1</fo:block>
</fo:block-container>
</fo:float>
```

見出しは流し込みの最初の要素として置いたため、前方フロート `axf:float-y="before"` を指定したとき、そのままでは見出しと衝突します。 `axf:float-offset-y` でフロートの開始位置をずらし、本文のブロック進行方向の開始位置と揃えています。見出しが `6rem`、見出しの後のアキが `1.5rlh` なのでこの値は `6rem + 1.5rlh` と指定しています。

`axf:float-reference="multicol"` で全段以外の段抜きを可能にしています。 `gr` は段抜きの単位で、各段と段間の幅を順に加算する特殊な単位です。ブロックコンテナに記述しているように `4gr - 0.8rem` のようにほかの長さとの計算を含む指定が可能です。

`axf:float-reference="multicol"` では `axf:float-wrap="skip"` を指定すると、フロートがまたがっていない段でも本文のブロックがフロートの高さを除けてしまいます。

## 本文に挟まるフロートを指定する

```
<!-- 段内に収まるフロート -->
<fo:float
  axf:baseline-block-snap="center" axf:baseline-grid="none"
  axf:float-x="inside" axf:float-y="center"
  axf:float-margin-y="0.7rlh" axf:float-margin-x="6mm"
  axf:float-reference="multicol">
  <fo:block-container width="1gr - 6mm"
    height="8rlh" border="1pt solid" ...>
    <fo:block-container width="34mm" ... >
    ...
  </fo:block-container>
  ...
```

```
</fo:block-container>
</fo:float>
```

1段に収まるサイズのフロートでは、本文がフロートによって区切られたようにみせないため、また本文の可読性を下げないため図表に対し左右のマージンを大きく取りました。

`axf:float-reference="multicol"`を指定しています。`axf:float-reference="column"`のとき、`axf:float-x="inside"`を指定しても、フロートはページの内側に配置されないためです。

見出し下のフロートと同じ理由で `axf:float-wrap="wrap"`を指定しています。段内でのテキストの回り込みを避けるため `axf:float-margin-x` プロパティでマージンを指定しています。

## 本文と異なる段組を指定する

```
<!-- コラム欄のフロート -->
<fo:float axf:float-margin-y="0.5rlh"
  axf:baseline-block-snap="before"
  axf:float-x="outside" axf:float-y="bottom"
  axf:float-reference="multicol"
  axf:float-offset-y="3mm">
  <fo:block-container text-indent="0"
    width="3gr" border="2pt solid"
    axf:border-radius="3pt"
      line-height="1"
      line-stacking-strategy="max-height"
      axf:baseline-grid="none">
    <fo:block
      padding-top="0.35rlh" padding-bottom="0.35rlh"
      font-family="sans-serif"
      background-color="cmyk(0, 0, 0, 1.0)"
      color="cmyk(0, 0, 0, 0)"
      text-align="center" font-size="1.2rem">
      <fo:inline letter-spacing="1pt"
        font-variant="small-caps"
        >Short Column</fo:inline>
    </fo:block>
  <fo:block-container margin="3mm">
    column-count="2" axf:column-gap="5mm"
    axf:column-rule-style="wave"
    axf:column-rule-width="3pt"
    axf:column-rule-length="90%"
    axf:column-rule-align="center"
    axf:column-fill="balance"
      <fo:wrapper font-size="0.8rem" line-height="1.2">
        <fo:block text-indent="0" axf:baseline-grid="new"
          line-stacking-strategy="max-height">...
        </fo:block>
      </fo:wrapper>
  </fo:block-container>
```

```
</fo:block-container>  
</fo:float>
```

axf:float-y="bottom"を指定したフロートの子としてブロックコンテナに2段組を指定しています。

この見出しブロックは、段組のブロックコンテナの兄弟として記述しています。ブロックコンテナによる段組の中で span="all"を指定して段抜きにすることも可能ですが、枠線と見出しブロックの背景色を同じにしています。

本文とはっきり区別するため、フォントサイズを小さくし、段間罫のスタイルを wave にしています。axf:baseline-grid="new"を指定し、行グリッドを新しく指定しました。

## 1.6 節 論文誌風のページデザイン

段組と脚注の配置、フロートの表示位置を意識したページを作成します。

段ごとの脚注、脚注項目のインライン表示、書式変更、細かなフロート配置などを指定します [図 1.6-1] (p.42)。

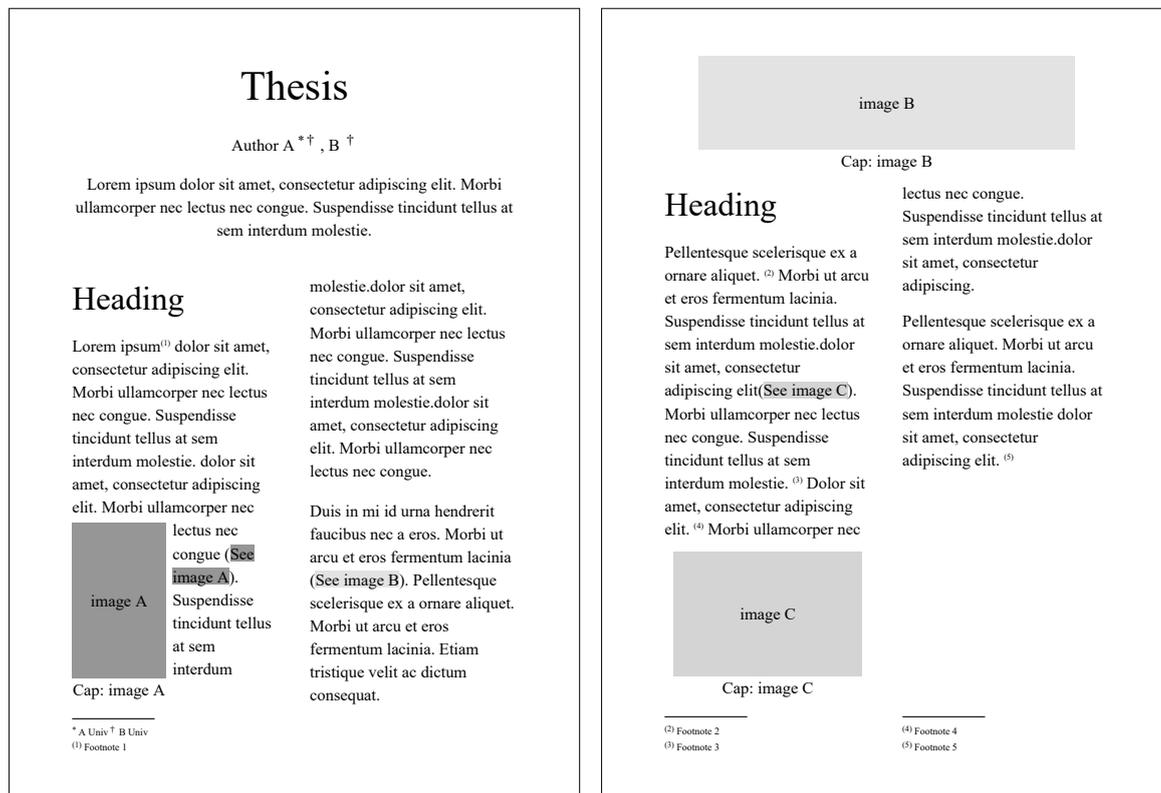


図 1.6-1 論文誌風のページデザイン

### 段ごとの脚注

```
<!-- 段ごとの脚注 -->
<fo:region-body ...
  column-count="2" column-gap="10mm"
  axf:footnote-position="column" />
...
</fo:region-body>
```

段組の脚注は、未指定のとき全段抜きで配置されます。axf:footnote-position="column"で段ごとの脚注を指定しています。

## 数字をシンボルに変更する

```
<!-- 番号書式をシンボルに変更する -->
<fo:declarations>
  <axf:counter-style name="cs-01" system="symbolic" symbols="* †">
</fo:declarations>
...
<fo:page-sequence ...>
  ...
  <fo:block ...>
    ...
    <fo:footnote ...>
      ...
      <fo:footnote-body>
        <fo:block>
          <!-- cs-01の書式により、1が"*"に置換される-->
          <fo:inline baseline-shift="super"
            axf:number-transform="cs-01"
            font-size="0.8em">1</fo:inline>
          ...
        </fo:block>
      </fo:footnote-body>
    </fo:footnote>
  ...
</fo:block>
...
</fo:page-sequence>
```

一部の脚注の合印にシンボルを利用します。直接シンボルを記述するのではなく AH XSL Formatter が数字をシンボルに変換します。数字の書式変換規則を `axf:counter-style` プロパティで用意し、この書式を、著者所属の注の書式として `axf:number-transform` プロパティで指定しています。

XSLT などによる前処理では数をカウントする処理を書くのみとなります。

`system="symbolic"` では、指定されたシンボルを順に利用します。最後のシンボルの次は、また最初のシンボルから利用しますが、前の順よりひとつシンボルを多く繰り返して表示します。この FO では「`symbols="* †"`」と記述しているため、「\*」、「†」、「\*\*」、「††」、「\*\*\*」..... という順に表示されます。

## タイトル部分の注を本文の注と同じ箇所に表示させる

```
<!-- タイトル部 -->
<fo:block ... span="all">
  <fo:block ...>Thesis</fo:block>
  <fo:block ...>Author A
  <fo:inline baseline-shift="super"
```

```

        font-size="0.8em">1</fo:inline>
    Author B
    <fo:inline ...>2</fo:inline>
</fo:block>
</fo:block>
...
<!-- 本文開始 -->
<fo:block line-height="0">
...
</fo:block>
<!-- 見出しブロック -->
<fo:block>
...
</fo:block>

```

XSL 1.1 仕様では、脚注はそのアンカが存在する参照エリア内で表示されるため、段抜きのブロックで指定した脚注はそのブロック内で表示されることになります。段抜きで著者所属を脚注とするとき、段抜き内で<fo:footnote>を使用するとタイトル部の下に脚注が表示されてしまいます。

段組の脚注として表示するために、タイトル部では上付き文字としてシンボルを表記し、段組が開始する本文の最初の位置で、アンカの合印の中身が空要素で、<fo:footnote-body>に著者所属の情報を格納した高さ 0 のブロックを作成しました。

## ▲ 注意事項

表示に都合を合わせた記述のため、タグ付き PDF として出力するときは axf:tag プロパティなどで調整を行うとよいでしょう。

## 一部の脚注をインライン表記にする

```

<!-- 著者所属の注ブロック-->
<fo:footnote axf:footnote-stacking="inline">
...
</fo:footnote>

```

未指定のとき、<fo:footnote-body>はそれぞれブロックとして扱われるため、別の注は別の行に表示されます。axf:footnote-stacking="inline"を指定するとこれをインライン要素とします。<fo:footnote>ごとに指定し、同じ脚注領域でインラインの脚注とブロックの脚注の混在が可能です。

## 図表を段の左寄せページ下部に配置する

```

<!-- 段の左寄せフロート -->
<fo:float axf:float-x="start"
    axf:float-reference="column"
    axf:float-y="after"

```

```
axf:float-move="auto-next"
...
>...</fo:float>
```

図表のフロート配置を `axf:float-x="start"` で左側、`axf:float-y="after"` でページ下部に指定します。配置を片側に揃えることで、図表によって文章が遮られないようにしています。

## マージンが十分でないとき、テキストを回りこませず図表を中央に配置する

```
<!-- マージンの最低値を指定する -->
<fo:float ...
  axf:float-wrap="wrap" axf:float-min-wrap-x="4em"
  axf:float-margin-x="0.3em" axf:float-margin-y="0.1em"
  axf:float-centering-x="auto" >
  ...
</fo:float>
```

`axf:float-wrap="wrap"` でテキストが回りこむように指定します。ただし、図表のサイズによっては回りこんだテキストが1行に1文字など、見づらくなってしまうときがあるでしょう。これを回避するため、回りこむ幅が十分にあるかを `axf:float-min-wrap-x` で指定します。また、回りこむ幅が足りないとき中央に配置するよう、`axf:float-centering-x="auto"` を指定しています。回りこみ幅が足りないときに中央寄せすることが決まっているのなら、`axf:float-centering-x` に数値やパーセントを指定すれば `axf:float-min-wrap-x` は必要ありません。

## 大きめの図表を上部中央に段抜きで配置する

```
<!-- 大きめの図表を上部中央段抜きで配置 -->
<fo:float
  axf:float-reference="page"
  axf:float-y="before" axf:float-x="center"
  axf:float-wrap="skip"
  axf:float-move="next"
  axf:float-margin-y="1em">
  ...
  <fo:block ...>Cap: Image B</fo:block>
</fo:float>
```

サイズが大きい図表については段抜きで配置しました。`axf:float-reference="page"` を指定しています。`axf:float-x` プロパティへ `center`、`axf:float-y` プロパティへ `before` を指定し、上部中央に配置します。

`axf:float-move` プロパティの指定を `auto-next` ではなく `next` にし、アンカのある箇所より後に図表を配置しました。

テキストの回りこみはしないことを、`axf:float-wrap="skip"` で明示しています。

## 1.7 節 数学参考書風のページデザイン

MathML 3.0 で数式を記述し、数学の参考書などの組版も行えます [図 1.7-1] (p.46)。

**問題 1 因数分解**

次の整式の種類項をまとめなさい。

①  $3x - x^2 + 4x + 5 + 2x^2$

②  $2x^2 + 3xy + 2x + 2xy + 3y^2 + 2y + 2x + 3y + 2$

**回答**

①  $3x - x^2 + 4x + 5 + 2x^2$   
 $= (-1 + 2)x^2 + (3 + 4)x + 5$   
 $= x^2 + 7x + 5$

②  $2x^2 + 3xy + 2x + 2xy + 3y^2 + 2y + 2x + 3y + 2$   
 $= 2x^2 + (3 + 2)xy + 3y^2 + (2 + 2)x + (2 + 3)y + 2$   
 $= 2x^2 + 5xy + 3y^2 + 4x + 5y + 2$

2x や  $-3ay^2$  のように、文字や数の積として表される式を単項式といいます。単項式の和としてあらわされる式を多項式といいます。単項式と多項式を合わせて整式といいます。

**ポイント**

- ・整式において、文字の部分が同じ項を種類項といいます。
- ・種類項をまとめた整式において、各項の次数のうちで最も高いものをその整式の次数といい、次数が m である整式を m 次式 といいます。2 種類以上の文字を含む整式では、特定の文字だけに注目して次数を考えることもあります。注目した文字を含まない項を定数項といいます。
- ・整式は、特定の文字について、次のように整理することがあります。

降べきの順	多項式を、次数が高いものから順に並べること
昇べきの順	整式の項を次数の低い順に並べること

図 1.7-1 数学参考書風のページデザイン

## MathML でブロック数式を記述する

```
<!-- ブロック数式 -->
<fo:block>
  <fo:instream-foreign-object>
    <math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
      <mn>3</mn><mi>x</mi><mo>-</mo>
      <msup><mrow><mi>x</mi></mrow>
      <mrow><mn>2</mn></mrow></msup>
      ...
    </math>
  </fo:instream-foreign-object>
</fo:block>
```

`<fo:instream-foreign-object>`要素の子へ、MathML を直接記述しています。

`display` プロパティによってブロック数式とインライン数式のどちらなのかを指定しないとき、オプション設定 `mathDisplay` プロパティの指定に従います。`mathmlDisplay="context"` のとき、`display` プロパティが無指定ならば AH XSL Formatter が周囲の要素から判断しますが、可能な限り明示しましょう。

## MathML でインライン数式を記述する

```
<!-- インライン数式 -->
<fo:block>
  <fo:inline space-start="0.25em" space-end="0.25em">
    <fo:instream-foreign-object>
      <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline">
        <mn>2</mn><mi>x</mi>
      </math>
    </fo:instream-foreign-object>
  </fo:inline>や<fo:inline>
  <fo:instream-foreign-object>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      ...
    </math>
  </fo:instream-foreign-object>
  </fo:inline>のように、文字や数の積として表される式を単項式といいます。...
</fo:block>
```

和文のテキスト中にインラインで MathML を記述するとき、数式とテキストのアキはユーザ側が調整します。この FO ではスペースなどを挿入することを避け、`<fo:inline>`へ `space-start`、`space-end` プロパティで指定しています。

MathML そのものについての解説は、W3C Math Home のページや、アンテナハウスから刊行された『MathML 数式組版入門』（道広勇司著，アンテナハウス CAS 電子出版，2017）をご覧ください。AH XSL Formatter の MathML に関連するオプション設定の詳細はマニュアルをご覧ください。

## ボーダーの上に見出しを重ねる

```
<!-- 直前のオブジェクトとの距離のみを指定したブロックコンテナ -->
<fo:block-container space-before="10mm">
  <!-- 見出しのブロックコンテナ -->
  <fo:block-container absolute-position="absolute"
    top="" left=""
    z-index="2">
    inline-progression-dimension="18pt * 4"
    block-progression-dimension="18pt + 2pt"
    <fo:block font-size="18pt" ... >
    回答
  </fo:block>
</fo:block-container>
<!-- 見出し以外の内容を含むブロックコンテナ -->
<fo:block-container space-before="10pt"
  z-index="1">
  ...
</fo:block-container>
</fo:block-container>
```

「回答」の見出しと内容を三つのブロックコンテナに分けています。

最も外側のブロックコンテナに直前のオブジェクトとのアキのみを記述しています。

見出しを含むブロックコンテナは絶対配置とし、その他の内容を含むブロックコンテナの `before` と `start` にボーダーを記述しています。見出しのブロックコンテナをボーダーのブロックコンテナの子にしてしまうと、ボーダーが見出しの上に描画されてしまうため、兄弟として記述し、`z-index` プロパティを指定して、見出しがボーダーの上に重なるようにしました。

同様の見た目を得るほかの方法として、上側のボーダーを表示せず、ブロックの子に見出しとリーダーを並べて記述する方法があります。こちらの方法では、見出しの文字数が多少変動しても、他の箇所の指定を変更せずに済みます。

```
<!-- 見出しとfo:leaderによる上側ボーダー -->
<fo:block>
  <fo:inline background-color="gray" ...>回答</fo:inline>
  <fo:inline color="gray">
    <fo:leader leader-pattern="rule"
      rule-thickness="1pt" rule-style="solid" .../>
  </fo:inline>
</fo:block>
<fo:block
  border-start-width="1pt"
```

```
border-start-color="gray"  
border-start-style="solid">  
...  
</fo:block>
```

## フロートによる傍注

数式の部分にだけ傍注を置くため、サイドフロートとして注を配置しています。数式の横からずれてしまうのは好ましくないため、`axf:float-move="keep"`を指定し、フロートを配置する余白がなければ数式ごと次のページに移動されます。

### 注

---

`<fo:region-start>`や`<fo:region-end>`を傍注として利用するのであれば、脚注の配置を指定する `axf:footnote-position` プロパティが有用です。

---

# 1.8 節 帳票風のページデザイン

ブロックコンテナによる絶対配置、表組、PDF 対話フォームを含む帳票をデザインします [図 1.8-1] (p.50)。

## 物品購入申請書

年 月 日

所属

---

合計金額

氏名 印

---

製品	単価(円)	数量	金額(円)	摘要

備考


Antenna House, Inc.

図 1.8-1 帳票風のページデザイン (Acrobat Reader での表示)

## 項目をブロックコンテナで絶対配置指定する

```
<!-- 項目を決まった位置に配置する -->
<fo:block-container absolute-position="absolute"
  top="0mm" left="90mm" width="80mm" height="10mm"
  ... >
  <fo:block text-align="right" ...>
    <fo:inline>...年...月...日...</fo:inline>
  </fo:block>
</fo:block-container>
```

`absolute-position="absolute"`を指定したブロックコンテナに、上下方向と左右方向のオフセットを指定します。このときの基点はページの端ではなく現在の参照領域となります。

## テキストの書式を指定する

```
<!-- 単価を入力するフィールド -->
<axf:form-field axf:field-name="Tanka001"
  axf:field-type="text" axf:field-default-text=""
  axf:field-multiline="false"
  axf:field-scroll="false"
  axf:field-maxlen="5"
  ...
  axf:field-text-align="right"
  axf:field-format="0 0 0 ' ' true"
  axf:field-format-category="number"/>
```

数値などの入力欄はテキストフィールドとして設定します。書式を指定することで数値として処理しやすくします。「百円」「100」など、フォーマットの混在を避けられます。

`axf:field-format-category="number"`で、入力値は数値に限定されています。`axf:field-format="0 0 0 ' ' true"`は、「小数点以下表示なし」「3桁で区切り」「負数を加工して表示しない」「通貨記号は空白」「通貨記号を前置」となります。小数点揃えなどを考慮する必要がないため、`axf:field-text-align="right"`で右揃えを指定しています。

ここでは表示の書式のみを指定しています。表示上の小数点の四捨五入や通貨記号の単位の対応といったものは値の計算には影響しません。

## 入力した項目の小計、合計を自動で表示する

```
<!-- 個数 -->
<axf:form-field axf:field-name="Count001"
  axf:field-type="text"
  axf:field-default-text=""
  axf:field-maxlen="4"
  axf:field-text-align="right">
```

```

axf:field-format="0 1 0 ' ' true"
axf:field-format-category="number"
width="2em" height="1.2em" ...>
<axf:form-field-event name="OnBlur" axf:action-type="javascript">
  g=11;
  n = this.getField("Count001").value;
  t = this.getField("Tanka001").value;
  if( n != "" && t != ""){
    this.getField("syoukei1").value = n * t;
    var t=0;
    for ( i = 1; i <= g; i++ ) {
      var fieldName="syoukei"+i;
      k=this.getField(fieldName).value;
      if( k != ""){
        t+=k;
      }
    }
    this.getField("goukei").value = t;
  }
</axf:form-field-event>
</axf:form-field>

```

PDF には JavaScript のサブセットを記述可能です。<axf:form-field-event> へ action-type="javascript"を指定して、内容に JavaScript を記述します。ここでは小計を計算して合計を表示させています。

個数のテキストフィールドのイベントとして記述したので、個数が入力されなければ動作しません。動作のタイミングは<axf:form-field-event>の name を OnBlur としています。つまり、別の項目をクリックするなど、この field-name="Coun001"からフォーカスが離れたときに記述した JavaScript が発火します。FO に記述するため一部記号をエスケープして記述します。

入力可能な項目を 11 個としているため、g=11 としています。

## 計算結果のフィールドを読み込み専用に指定する

```

<!-- 小計 -->
<axf:form-field
axf:field-type="text" axf:field-name="syoukei1"
  axf:field-format-category="number"
  axf:field-format="0 0 0 ' ' true"
  ...
  axf:field-maxlen="7"
  axf:field-readonly="true"/>

```

小計や合計金額は入力した個数と金額から求められる値です。計算結果表示のためのフィールドであるので、PDF ビューアでの直接編集は行えないようにします。axf:field-readonly="true"または axf:field-flags="ReadOnly"を指定することで、読み込み専用フィールドとします。

## 複数行テキストの入力フィールドに罫線を引く

`axf:field-multiline="true"`で複数行のテキストフィールドを指定可能です。しかし、この複数行のテキストフィールドの内部は XSL-FO のオブジェクトとしては行としてカウントされないため「行ごとに下線を引く」ように指定しても、フィールドの終わりに下線が引かれるだけです。

```
<!-- 複数行テキストフィールドへ背景を指定 -->
<axf:form-field axf:field-type="text"
  axf:field-name="bikou"
  axf:field-multiline="true"
  height="5lh" width="80vw"
  background-image="repeating-linear-gradient(0 cmyk(0,0,0,0),
    1lh - 0.3pt cmyk(0,0,0,0), 1- 0.3pt cmyk(0,0,0,1.0))"
```

`axf:form-field` プロパティには背景を指定可能です。フィールドに入力するフォントサイズと間隔を合わせて `repeating-linear-gradient()` で罫線のように細くした黒色を繰り返しています。

## 文書をロックする電子署名フィールドを指定する

```
<!-- 電子署名フィールドの指定-->
<axf:form-field axf:field-name="sig"
  axf:field-type="signature"
  axf:field-lock-document="true" .../>
```

`axf:field-lock-document="true"`を指定した電子署名フィールドは、出力後の PDF でこのフィールドに署名を行うと、以後その PDF の編集をロックします。



# 第 2 章

## AH XSL Formatter 拡張 仕様と応用例

## 2.1 節 単位

XSL 1.1 仕様で利用できる単位の他にもさまざまな単位を指定できます。

### 2.1.1 長さの絶対値を利用する

q (級数) など、組版で用いられる単位をそのまま記述可能です。

Unit Name	意味	等価な長さ
jpt	JIS X 8305 で定められている。	1 jpt = 0.3514 mm
q	JIS X 4052、JIS Z 8125 で定められている。級数。	1q = 0.25 mm
emu	English Metric Unit。	$1 \text{ emu} = \frac{1}{91400} \text{ in} = \frac{1}{360000} \text{ cm}$
dd	didot。ヨーロッパのタイポグラフィに由来する単位。	1 dd = 0.01483 in
cc	cicero。ヨーロッパのタイポグラフィに由来する単位。	1 cc = 12 dd

### 2.1.2 長さの相対値を利用する

長さの単位として利用できる相対的な単位の種類も拡張されています。

Unit Name	意味	フォールバック
lh	行の高さを 1 とする。	-
rem	ルート要素の em を 1 とする。	-
rlh	ルート要素の lh を 1 とする。	-
vw	ビューポートの幅を 100 とする。	-
vh	ビューポートの高さを 100 とする。	-
pvw	ページの幅を 100 とする。	-
pvh	ページの高さを 100 とする。	-
pvmax	pvw、pvh の長さのうち値が大きいものと同値。	-
pvmin	pvw、pvh の長さのうち値が小さいものと同値。	-
cap	フォントの cap-height を 1 とする。	フォントの altitude
ex	フォントの x-height を 1 とする。	0.5em
ch	DIGIT ZERO 「0」 (U+0030) の幅を 1 とする。	0.5em
ic	CJK UNIFIED IDEOGRAPH-6C34 「水」 (U+6C34) の幅を 1 とする。	1em
dcem	ドロップキャップされた文字のフォントサイズを 1 とする。	1em
gr	段抜き指定の単位。	-

font-size="12pt"、line-height="1.2"であるような箇所で<length>を 1lh とすると、内部的に 14.4 pt として変換され処理されます。

他の箇所で指定された値を基準とするため、指定できる箇所が限られます。たとえば行の高さを基準とする単位 lh は、line-height プロパティの指定には利用できません。指定できない箇所や、参照する基準の値が決定できないとき、フォールバックの値が利用されます。

gr は、段組のとき、段や段間を 1 として数える特殊な単位です。段組時かつフロートの子要素でのみ指定可能です。

## ビューポートの基準を指定する

「表示域」とも呼びます。ビューポートは vw や vh としては、紙面など固定画面で見られる（画面内でのスクロールなどを考慮しない）現在の領域のことです。XSL と CSS3 で解釈が異なります。オプション設定で基準を決定します。

OptionProperty Name	viewport-length-units-mode
Value	5   6
Initial	6
Applies to	<formatter-settings>

vw、vh の基準をオプション設定 viewport-length-units-mode プロパティで制御します。

5 のときページサイズが基準、6 のときページのマージンを除いた領域サイズが基準です。

6 を指定したとき、ページのマージンを含めた領域を 100 とする単位には、代わりに pvw と pvh が利用可能です。

### 2.1.3 単位の無い「0」を指定する

長さ 0 のときのみ、単位なしを受け付けます。

XSL 1.1 仕様では長さの値には常に単位が必要です。AH XSL Formatter では 0 のときのみ text-indent="0"のように単位なしを受け付けます。長さの計算式中で単位の無い 0 が登場することは不正です。

### 2.1.4 解像度指定に dpi を利用する

dpi による指定が可能なプロパティが存在します。

Unit Name	意味
<dpi>	解像度を表す。数値に dpi を付加した形で記述する。

afx:image-resolution プロパティなどでは、解像度を指定するときに dpi による指定が可能です。afx:image-resolution="300dpi"のように指定します。

## 2.1.5 px 単位の変換を制御する

px による長さの指定は物理的な値に変換する必要があります。

「インチあたりのピクセル数」をオプション設定で指定可能です。

Option Property Name	pxpi
Value	<number>
Initial	96
Applies to	<formatter-settings>

変換の係数を、オプション設定 pxpi プロパティへ、正の実数値で指定します。画像で縦横の長さをピクセル値で持っていて解像度が無いものも、pxpi プロパティの値を基準に変換されます。

## 2.1.6 ページサイズを判型で指定する

ページサイズ指定で、規格化された用紙サイズをその名前指定できます。

XSL 1.1 仕様ではページの幅と高さをそれぞれ長さで指定可能です。しかし、多くは 210×297mm (A4 サイズ) など、印刷に用いるような定形サイズを指定するでしょう。AH XSL Formatter では size に指定可能な値が拡張されており、size プロパティへ判型 (用紙名) を指定可能です。大文字小文字は区別されません。

size	横×縦	規格
A3	297 × 420 mm	ISO 216
A4	210 × 297 mm	ISO 216
A5	148 × 210 mm	ISO 216
A6	105 × 148 mm	ISO 216
B4	250 × 353 mm	ISO 216
ISO-B4	250 × 353 mm	ISO 216
JIS-B4	257 × 364 mm	JIS P 0138
B5	176 × 250 mm	ISO 216
ISO-B5	176 × 250 mm	ISO 216
JIS-B5	182 × 257 mm	JIS P 0138
B6	125 × 176 mm	ISO 216
ISO-B6	125 × 176 mm	ISO 216
JIS-B6	128 × 182 mm	JIS P 0138
Letter	8.5 × 11 in	North American Paper Sizes

size	横×縦	規格
Government-Letter	8 × 10.5 in	North american Paper Sizes
Legal	8.5 × 14 in	North American Paper Sizes
Ledger	17 × 11 in	North American Paper Sizes
Tabloid	11 × 17 in	North American Paper Sizes
Statement	5.5 × 8.5 in	-
Executive	7.25 × 10.5 in	-
Folio	8.5×11in	-
C	17 × 22 in	ANSI Paper Sizes
D	22 × 34 in	ANSI Paper Sizes
E	34 × 44 in	ANSI Paper Sizes
ISO-DL	110 × 220 mm	ISO 269
ISO-C3	324 × 458 mm	ISO 269
ISO-C4	229 × 324 mm	ISO 269
ISO-C5	162 × 229 mm	ISO 269
ISO-C6	114 × 162 mm	ISO 269
Hagaki	100 × 148 mm	日本郵便

これらの指定には portrait と landscape を付けてどちらが幅、高さであるかを明示可能です。無指定あるいは portrait では長辺が高さとなります。landscape では短辺が高さとなります。

```
<!-- A4 横長を指定 -->
<fo:simple-page-master master-name="a4page" size="A4 landscape" >
...</fo:simple-page-master>
```

## 2.1.7 ユースケース：拡張された単位を利用する

### n 行取りを指定する

rlh はルートで指定した行の高さです。ルートで本文のサイズを決定しておくことで「見出しブロックを本文 n 行取りにする」といった指定を簡単に行えます。1rlh + 1rem とすると和文 2 行取り指定が可能です。

### 基本版面に文字数と対応する値を利用する

A6 判縦組（幅 105 mm、高さ 148 mm）で本文区画を縦 40 字横 20 行としてマージンを均等にとる指定は、たとえば次のように指定可能です。

```
<!-- 行の高さを基準に指定する -->
<fo:region-body
  margin-top="74mm - 20rem" margin-bottom="74mm - 20rem"
  margin-left="52.5mm - 10rlh" margin-right="52.5mm - 10rlh"/>
```

実際に入る文字数は `line-stacking-strategy` プロパティなど、ほかの指定に影響を受けます。

## 2.2 節 ページ制御

ページの並びやサイズを制御します。

### 2.2.1 見開きでページをまたぐ要素を配置する

XSL 1.1 仕様にある機能を利用して左右ページで異なるレイアウトの調整を行うには、単純ページマスタをふたつ用意しページシーケンスマスタで偶数ページ、奇数ページそれぞれのページマスタを参照する方法があります。

長辺や短辺で綴じられることを想定し左右ページで異なるレイアウトの見開きページ自体はこの方法で実現しますが、左右ページにまたがって配置される見開き要素を扱うには不足しています。フロートや絶対配置のブロックコンテナを置いて、隣のページに残りの部分は表示されません [図 2.2-1] (p.63)。

Element Name	<axf:spread-page-master>
Property	axf:left-page-master-reference, axf:right-page-master-reference, reference-orientation, writing-mode
Parent	fo:layout-master-set
Element Name	<axf:spread-region>
Property	region-name, height, overflow, reference-orientation, width, writing-mode
Parent	axf:spread-page-master
Property Name	axf:left-page-master-reference, axf:right-page-master-reference
Value	<name>
Initial	なし (値が必須)
Applies to	axf:spread-page-master

<axf:spread-page-master>で見開きページマスタの記述が可能です。単純ページマスタを左右ページに割り当てることで、見開きページマスタとして扱えるようになります。

左ページの割り当ては axf:left-page-master-reference、右ページの割り当ては axf:right-page-master-reference プロパティへ単純ページマスタ名を指定します。

左右ページをまたがる要素を配置する見開き区画を<axf:spread-region>へ記述します。本文区画同様に区画の位置とサイズを指定します。このとき、見開き全体は左右ページを合わせたサイズとなります。<axf:spread-region>に指定する margin-left プロパティは左ページの左端からの、margin-right プロパティは右ページの右端からの距離です。

この見開き区画には、たとえば<fo:footnote>などを子孫に持つことはできません。見開き区画は本文区画と干渉する箇所では本文区画のフローよりも優先されます。

実際の利用ではページシーケンスマスタで左や右のみのページマスタと組み合わせることになるでしょう。

```
<!-- 見開きページと通常のページを組み合わせる -->
<fo:layout-master-set>
  <fo:simple-page-master master-name="left" ... >
    ...
  </fo:simple-page-master>
  <fo:simple-page-master master-name="right" ... >
    ...
  </fo:simple-page-master>
  <axf:spread-page-master master-name="spread"
    left-page-master-reference="left"... >
    right-page-master-reference="right"
  <axf:spread-region region-name="spreadRegion" .../>
</axf:simple-page-master>
<!-- ページマスタの出現条件を指定するページシーケンスマスタ -->
<fo:page-sequence-master master-name="main" >
  <fo:repeatable-page-sequence-master-alternatives>
    <fo:conditional-page-sequence-master-reference
      master-reference="right" page-position="first"/>
    <fo:conditional-page-sequence-master-reference
      master-reference="spread" page-position="rest"/>
    <fo:conditional-page-sequence-master-reference
      master-reference="left" page-position="last"/>
  </fo:repeatable-page-sequence-master-alternatives>
</fo:page-sequence-master>
</fo:layout-master-set>
...
<page-sequence master-reference="main"
  force-page-count="end-on-odd" ...>
  <!-- 見開き要素を表示する区画 -->
  <fo:flow flow-name="spreadRegion">
    <fo:block-container ...>
      ...
    </fo:block-container>
  </fo:flow>
  <!-- 通常の本文区画 -->
  <flow flow-name="xsl-region-body">
    ...
  </flow>
</page-sequence>
<page-sequence master-reference="spread">
  <flow flow-name="xsl-region-body">
    ...
```

</flow>  
</page-sequence>

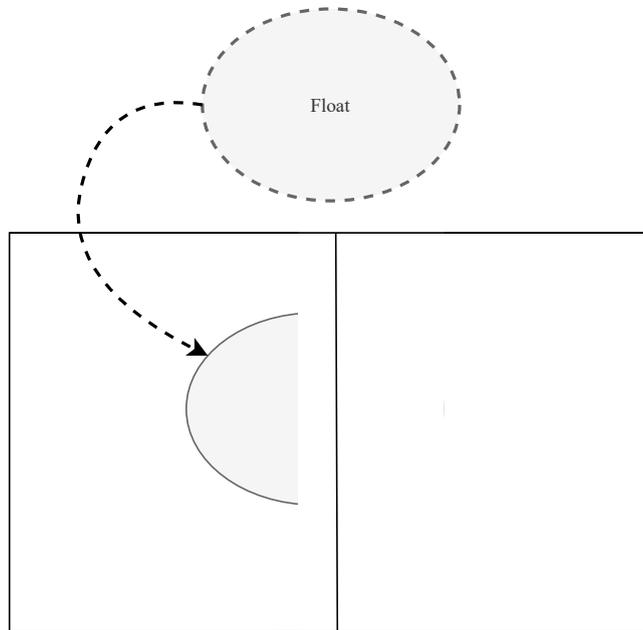


図 2.2-1 オブジェクトはページをまたげない

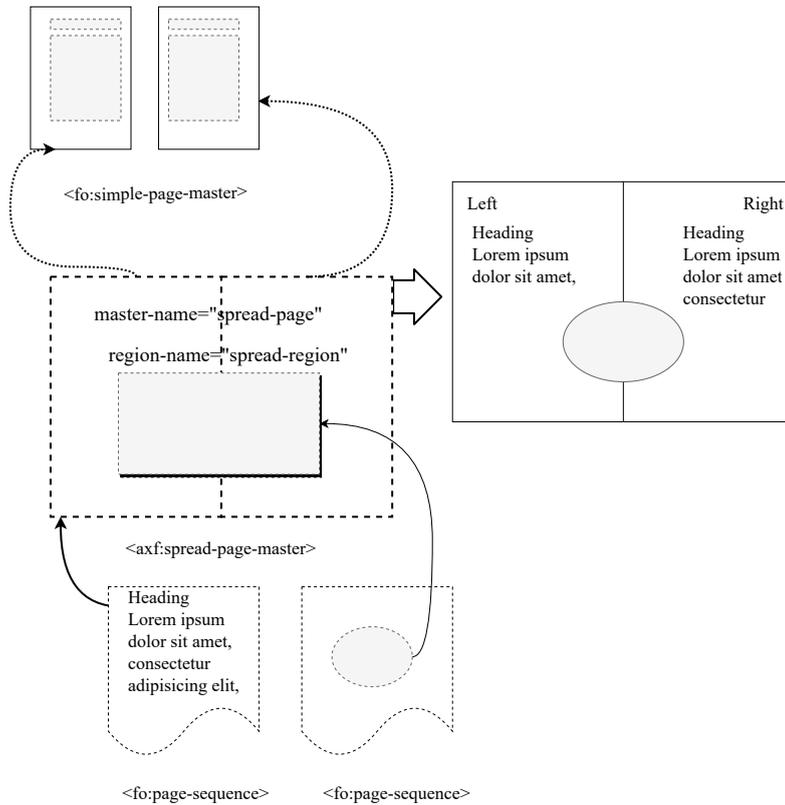


図 2.2-2 <axf:spread-page-master>と<axf:spread-region>による見開きページと見開き区画

## 2.2.2 ページシーケンスを入れ子で記述する

XSL 1.1 仕様ではページサイズを一部のみ切り換えるためには、ページシーケンスを終了し、別のページマスタを参照するページシーケンスを開始、終了し、元のページシーケンスと同じページマスタを参照するページシーケンスを新たに開始し.....という記述が必要になります。FO にそういった指定を行うことが難しいときもあるでしょう。

```
<!-- ページシーケンスを終了して切り換える -->
<fo:layout-master-set>
  <fo:simple-page-master size="A4 portrait" master-name="main">
    ...
  </fo:simple-page-master>
  <fo:simple-page-master size="A3 landscape" master-name="illust">
    ...
  </fo:simple-page-master>
</fo:layout-master-set>
...
<fo:page-sequence master-reference="main">
  ...
</fo:page-sequence>
<fo:page-sequence master-reference="illust">...
</fo:page-sequence>
<fo:page-sequence master-reference="main">...
</fo:page-sequence>
```

AH XSL Formatter では<fo:page-sequence>を入れ子にして記述できるよう拡張されています。

```
<!-- ページシーケンスを入れ子にする -->
<fo:page-sequence master-reference="main">
  ...
  <fo:page-sequence master-reference="illust">...
</fo:page-sequence>
...
</fo:page-sequence>
```

たとえば、ページサイズを変更して図表を挿入したいときに有用です。

## 2.2.3 ページ数を制御する

ページシーケンスのページ数を単純な奇数、偶数よりも細かく制御が可能です。XSL 1.1 仕様から `force-page-count` プロパティの値を拡張しています。

Property Name	force-page-count
Value	auto   even   odd   <b>doubly-even</b>   end-on-even   end-on-odd   end-on-doubly-even   even-document   odd-document

doubly-even-document | [ end-on |  
document]? <number> <number>? | no-force

Initial	auto
Applies to	fo:page-sequence

force-page-count	意味
doubly-even	指定したページシーケンスを強制的に4の倍数にする。
end-on-doubly-even	指定したページシーケンスの最後のページ番号を4の倍数にする。
even-document	文書先頭からカウントしたページ数を強制的に偶数にする。
odd-document	文書先頭からカウントしたページ数を強制的に奇数にする。
doubly-even-document	文書先頭からカウントしたページ数を強制的に4の倍数にする。
[ end-on   document]? <number> <number>?	ひとつめの<number>の倍数にふたつめの<number>を足したページ数にする。

doubly-even と end-on-doubly-even では指定されたページシーケンスを基準に4の倍数とします。通常の even、end-on-even の拡張といえるでしょう。

\*-document では文書先頭からのページ数を調整します。複数のページシーケンスを組み合わせるときに有用です。

使用機会の多い奇数、偶数、4の倍数では対応できないとき、<number>で倍数と残りのページ数を指定できます。先頭に end-on を付けるとそのページシーケンス内での、document を付けると文書先頭からのページ数として調整を行います。

強制的に足されたページは空白ページとして出力されます。

## 2.2.4 ページ出力順を逆順にする

ページシーケンス単位でページ内容を逆順に出力します。

Property Name	axf:reverse-page
Value	true   false
Initial	false
Applies to	fo:page-sequence

axf:reverse-page="true"を指定した<fo:page-sequence>は、ページを逆順に出力します。

このとき、ページ番号の振り方については元のままです。たとえば20ページの文書であれば、出力ページ上は1ページ目に20ページの内容とページ番号「20」が記載されます。

PDF出力のみに対応しています。また、分冊出力には対応していません。

## 2.2.5 ページシーケンスマスタの繰り返しを制御する

ページシーケンスマスタには複数のページマスタの登録が可能です。このとき、`<fo:single-page-master-reference>` で指定したページマスタをすべて使いきる、`<fo:repeatable-page-master-reference>` の `maximum-repeats` プロパティで指定したページ数の上限を超えるなど、出力するページがまだ残っている状態で参照するページマスタが足りなくなってしまうことがあります。

Property Name	axf:repeat-page-sequence-master
Value	true   false
Initial	false
Applies to	fo:page-sequence-master

`axf:repeat-page-sequence-master` プロパティへ `true` を指定すると、選択可能なページマスタがなくなったとき、`<fo:page-sequence-master>` を始めから繰り返します。

```
<!-- 参照するページは10ページ分だが、11ページでfirst、
12ページ以降からother、21ページでfirst..... -->
<fo:page-sequence-master master-name="main"
  axf:repeat-page-sequence-master="true" >
  <fo:single-page-master-reference master-reference="first" />
  <fo:repeat-page-master-reference master-reference="other"
    maximum-repeat="9" />
</fo:page-sequence-master>
```

## 2.2.6 任意の空ページを挿入する

AH XSL Formatter では、`<fo:page-sequence>` を記述するとき、子の `<fo:flow>` の記述を省略した、空の `<fo:page-sequence>` が許容されます。

ブランクページによるページ数調整などに有用です。

ただし、`force-page-count="no-force"` が指定されているときは `<fo:flow>` の記述が必要です。

## 2.2.7 ユースケース：異なるページサイズの図表を挿入する

簡単な記述で異なるサイズのページを記述の途中に含められます。

[図 2.2-3] (p.68) では 1 ページ目と 3 ページ目、2 ページ目と 4 ページ目が同じページサイズとなっています。

### 入れ子のページシーケンスで図表用のページを挿入する

```
<!-- サイズの異なるページマスタ -->
<fo:simple-page-master master-name="NormalPage"
  page-height="257mm" page-width="182mm">...
</fo:simple-page-master>
<fo:simple-page-master master-name="BigPage"
  page-height="257mm" page-width="364mm">...
```

```
</fo:simple-page-master>
...
<!-- 入れ子のページシーケンス -->
<fo:page-sequence master-reference="NormalPage">
...
  <fo:block ...>出典:...</fo:block>
    <fo:page-sequence master-reference="BigPage">
      ...<fo:external-graphic ...>
    </fo:page-sequence>
  <fo:block ...>...</fo:block>
...
</fo:page-sequence>
```

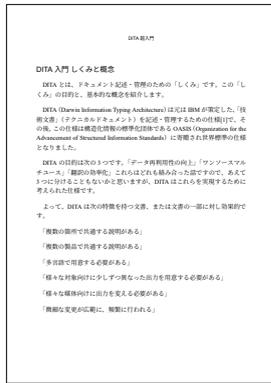
NormalPage を参照するページシーケンスを終了せず、子孫として BigPage を参照するページシーケンスを記述しています。

---

## 注

PDF などデジタル媒体での出力であれば、異なるページサイズの混在はあまり問題になりません。印刷を目的とするとき、2 ページを 1 枚の紙として印刷する都合上、途中に挿入する図が 1 ページのみでも、印刷された紙では裏に空白ページを使うこととなります。異なるサイズのページを複数含むときは、文書中で登場する箇所をあらかじめまとめておくのが良いでしょう。

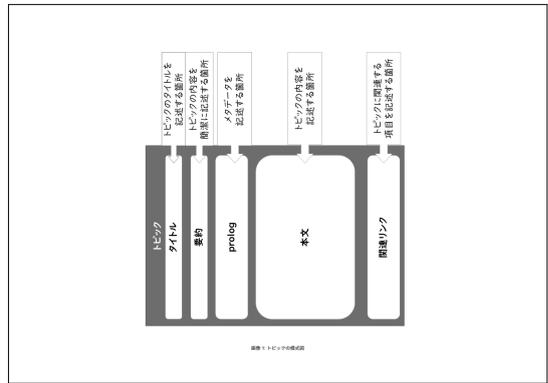
---



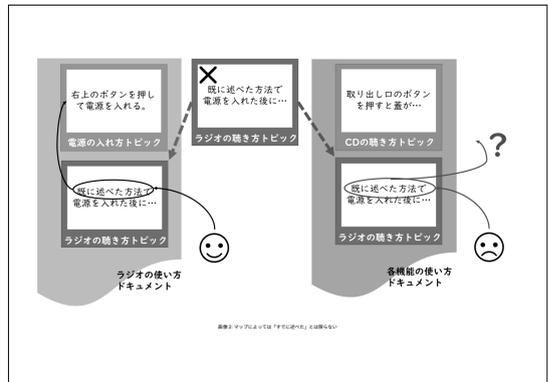
1 ページ



3 ページ



2 ページ



4 ページ

図 2.2-3 図表用のページサイズを途中で挟む

## 2.2.8 ユースケース：PDF のページ順を変更して出力する

AH XSL Formatter では<fo:external-graphic>要素や background-image プロパティに、指定したページの PDF を埋め込み可能です。PDF を新たな FO に埋め込み、印刷用のページ順で改めて出力するなどが可能です。

次の FO では、A4 サイズで 8 ページの冊子 PDF を A3 サイズに貼り付け「1-8」「2-7」、「3-6」「4-5」の組で出力します。

```

<!-- ページマスタへのPDFの埋め込み -->
<fo:layout-master-set>
  <fo:simple-page-master master-name="embed-1"
    background-image="embedded.pdf#page=1"
    background-repeat="no-repeat" >...</fo:simple-page-master>
  <fo:simple-page-master master-name="embed-2"
    background-image="embedded.pdf#page=8"
    ... >...</fo:simple-page-master>
  <fo:simple-page-master master-name="embed-3"
    background-image="embedded.pdf#page=2"
    ... >...</fo:simple-page-master>
  <fo:simple-page-master master-name="embed-4"

```

```

    background-image="embedded.pdf#page=7"
    ... >...</fo:simple-page-master>
<fo:simple-page-master master-name="embed-5"
    background-image="embedded.pdf#page=3"
    ... >...</fo:simple-page-master>
<fo:simple-page-master master-name="embed-6"
    background-image="embedded.pdf#page=7"
    ... >...</fo:simple-page-master>
<fo:simple-page-master master-name="embed-7"
    background-image="embedded.pdf#page=4"... >
    ...</fo:simple-page-master>
<fo:simple-page-master master-name="embed-8"
    background-image="embedded.pdf#page=5"... >
    ...</fo:simple-page-master>
<fo:page-sequence-master master-name="main" ...>
    <fo:single-page-master-reference master-reference="embed-1" />
    <fo:single-page-master-reference master-reference="embed-2" />
    <fo:single-page-master-reference master-reference="embed-3" />
    <fo:single-page-master-reference master-reference="embed-4" />
    <fo:single-page-master-reference master-reference="embed-5" />
    <fo:single-page-master-reference master-reference="embed-6" />
    <fo:single-page-master-reference master-reference="embed-7" />
    <fo:single-page-master-reference master-reference="embed-8" />
</fo:page-sequence-master>
</fo:layout-master-set>
    ...
<fo:page-sequence master-reference="main">
    <fo:flow flow-name="xsl-region-body"></fo:flow>
</fo:page-sequence>

```

## 2.3 節 オブジェクト修飾

拡張された仕様により多彩な修飾が可能です。

ボーダーの修飾が可能なオブジェクトとしては通常ブロックやインラインのほか、段間罫や改訂バーなどがあります。また、リーダーなどでも指定可能なボーダーの種類が拡張されています。

### 2.3.1 ボーダーを修飾する

ボーダースタイルに古い CSS3 のボーダーで規定されていた dot-dash、dot-dot-dash、wave が指定可能な値として拡張されています。

<border-style>	重なるときの優先順位
double	1
solid	2
dashed	3
dot-dash	4
dot-dot-dash	5
dotted	6
ridge	7
outset	8
groove	9
inset	10
wave	11

ボーダースタイルの優先順位      ボーダースタイルにはシステム側で定めた優先順位が存在します。ボーダーの指定が重なるときにどのボーダーを優先して描画するかについてほかの優先順位がないときに、この順位が利用されます。

### 2.3.2 波線の形状を制御する

ボーダースタイルに波線を指定したとき、波の幅や波長を細かく指定します。

Property Name	axf:border-wave-form
Value	<value>[ <value>]
Initial	auto
Applies to	all elements
Inherited	yes

axf:border-wave-form プロパティで波線の形状を指定可能です。<value>は<length>または<number>です。ひとつめが波長の値、ふたつめが波線の幅の指定となります。auto のとき、6 0.125 を指定したとみなされます。<length>では直接的な値、<number>ではボーダー幅に対する比率の指定となります。

[図 2.3-1] (p.71) に border-width="3pt" で axf:border-wave-form プロパティの値を変更した例を示します。

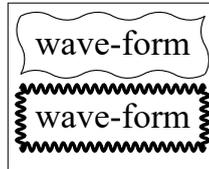


図 2.3-1 既定 ("6 0.125") と "3pt 0.4"

### 2.3.3 二重線の修飾を変更する

二重線の外側の線、内部の間隔、内側の線幅を細かく指定します。

border-style="double" のとき、ボーダーの線として二重線が描画されます。このとき、AH XSL Formatter では XSL 1.1 仕様ではボーダーの外側と内側の線幅それぞれを制御可能です。

Property Name	axf:border-double-thickness
Value	<value>[ <value>[ <value>]]
Initial	1
Applies to	all elements
Inherited	yes

axf:border-double-thickness プロパティにより、二重線のそれぞれの幅が指定可能です。

順に外側の線幅、内部の間隔、内側の線幅の指定となります。<value>は<number>または<length>です。省略されると、直前の値と等しいとみなされます。

外側の線、間隔、内側の線を合わせた幅は border-width プロパティへ指定した値となります。

### 2.3.4 ボーダーの角を丸める

丸め角（丸め罫）を表現します。

表示域を持つオブジェクト領域の形状は基本的に矩形のため、ボーダーの線を引いたり背景色を指定したとき、その角は直角に表示されます。

AH XSL Formatter では [CSS3-Background] Curve Radii: the 'border-radius' properties を基にしたプロパティによって丸め角が可能です。

Property Name	axf:border-radius, axf:border-top-left-radius, axf:border-top-right-radius, axf:border-bottom-left-radius, axf:border-bottom-right-radius
Value	[ <length>   <percentage>][ <length>   <percentage>]?
Initial	0
Applies to	all elements which can have borders
Percentages	border box

axf:border-\*radius プロパティに丸め幅を指定します。axf:border-radius プロパティはショートハンドで、すべての角に対し適用します。

ボーダーと接する円を仮定し、ひとつめの値が start-end 方向の径、ふたつめの値が before-after 方向の径となります [図 2.3-2] (p.72)。指定がひとつのみのとき、同じ値が適用されます。

ボーダーの罫線のほか、ボックスシャドウにも影響します。

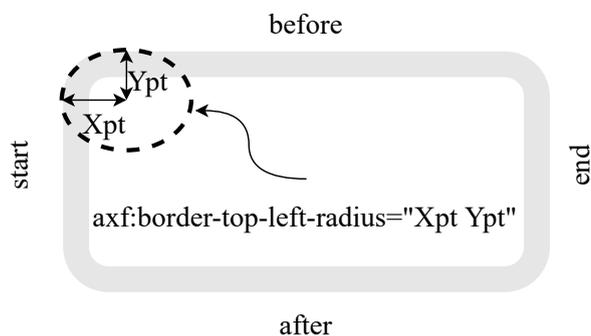


図 2.3-2 丸め角の指定

### 注

ボーダースタイルが wave のとき適用されません。

## 2.3.5 ボーダー接合部の形状を指定する

ボーダーの線同士が交差、接合する箇所の形状を指定します。

Property Name	axf:border-connection-form
Value	wedge   mixed   precedence
Initial	mixed
Applies to	all elements
Inherited	yes

axf:border-connection-form プロパティが対象とする接合部には通常のボーダーの四つの角（右上、右下、左上、左下の角）も含みますが、表のように、ボーダーの角同士が接するようなときにも有用です。

mixed では丁字の交差では矩形、ほかは楔型にします。wedge では十字の交差では X 型、丁字の交差では Y 字型となります。precedence を指定することで、隣接する辺のスタイルが異なるとき、ボーダーの優先順位の高い方を優先して端まで描画します。

border-collapse="collapse"を指定した表では、<fo:table>の axf:border-connection-form プロパティの値が採用され、<fo:table-row>や<fo:table-cell>で指定した値は無効です。

## 2.3.6 対角罫を引く

ブロック要素やインライン要素を斜めに通るようなボーダーを設定します。

Property Name	axf:diagonal-border-color, axf:reverse-diagonal-border-color
Value	<color>
Initial	the value of the 'color' property
Applies to	all elements which can have borders
Inherited	yes
Property Name	axf:diagonal-border-style, axf:reverse-diagonal-border-style
Value	<border-style>
Initial	none
Applies to	all elements which can have borders
Property Name	axf:diagonal-border-width, axf:reverse-diagonal-border-width
Value	<border-width>
Initial	medium
Applies to	all elements which can have borders
Inherited	yes

axf:diagonal-border-\*, axf:reverse-diagonal-border-\* プロパティにより、領域の端から端に斜線を引くことが可能です。領域のサイズがどれくらいになるか分からないもの、表のセルのように分割される可能性があるものに斜線を引くときに有用です。

axf:diagonal-border-\*が before-start の頂点から after-end の頂点、axf:reverse-diagonal-border-\*が before-end の頂点から after-start の頂点への罫線となります。

## 2.3.7 区画のボーダーを表示する

本文区画などでボーダー修飾を指定します。

XSL 1.1 仕様では region-\*のボーダー幅は 0 以外にはできず、線が描画できません。AH XSL Formatter では region-\*でも 0 以上のボーダー幅を指定可能で、ボーダー修飾などの対象となります。

## 2.3.8 オブジェクトにボックスシャドウを付加する

ブロックやインラインに影を付加します。

Property Name	axf:box-shadow
Value	none   [ inset?&&[ <length>{2,4}&& <color>? ] ]#
Initial	none
Applies to	all elements which can have borders

ボーダーを持つようなオブジェクトに対し、axf:box-shadow プロパティでボックスシャドウの付加が可能です。<X のオフセット位置> <Y のオフセット位置> <ぼかし半径> <拡大値> <影の色> を組として指定し、複数のとき COMMA 「,」（U+002C）で区切って指定します。

また、inset を指定することで内側方向にボックスシャドウを描画します。

出力設定で PDF/X や、transparency="false" など透過ができない指定のとき、ぼかしは無視されます。

### 注

axf:border-radius\* プロパティによる角の丸みはボックスシャドウに反映されます。

border-style="wave" を指定しているとき、波の形状は影に反映されません。波の形状の影を表現したいとき、SVG などであらかじめ影のついた波線の枠を作成してください。

## 2.3.9 ブロックを変形する

ブロックを拡大縮小、回転、移動、歪曲します。

Property Name	axf:transform
Value	none   <transform-function>+
Initial	none
Applies to	transformable objects

axf:transform プロパティへ変形のための関数と引数を指定することで、ブロックの変形が可能です。領域の変形は起こらず、内容のみ変形されます [図 2.3-3] (p.75)。

Value	Value Composition
<transform-function>	matrix()   translate()   translateX()   translateY()   scale()   scaleX()   scaleY()   rotate()   skew()   skewX()   skewY()

matrix() と rotate() 以外は、引数がひとつであれば X 方向と Y 方向に同じ値を適用し、引数がふたつであればひとつめが X 方向、ふたつめが Y 方向に適用する値です。また、これらの関数名に \*X、\*Y をつけた関数はその方向のみの変形を行います。

Function Name	<code>matrix()</code>
Argument Value	<code>&lt;number&gt;, &lt;number&gt;, &lt;number&gt;, &lt;number&gt;, &lt;number&gt;, &lt;number&gt;</code>

`translate()`、`skew()`、`scale()` に相当する変形を行列式として指定可能です。

Function Name	<code>translate()</code>
Argument Value	<code>&lt;translation-value&gt;[, &lt;translation-value&gt;]</code>

ブロックを移動します。`translateX()` は X 方向、`translateY()` は Y 方向のみの指定です。

Function Name	<code>scale()</code>
Argument Value	<code>&lt;number&gt;[, &lt;number&gt;]</code>

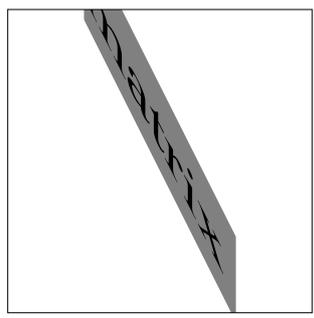
`scale()` はブロックを拡大・縮小します。

Function Name	<code>rotate()</code>
Argument Value	<code>&lt;angle&gt;</code>

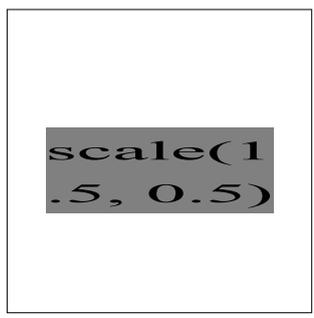
`rotate()` はブロックを回転します。

Function Name	<code>skew()</code>
Argument Value	<code>&lt;angle&gt;[, &lt;angle&gt;]</code>

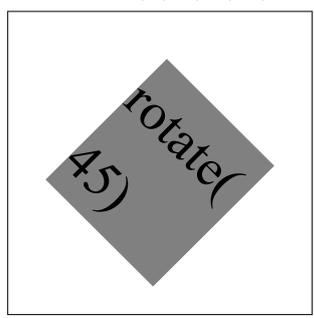
`skew()` はブロックを傾斜変形します。



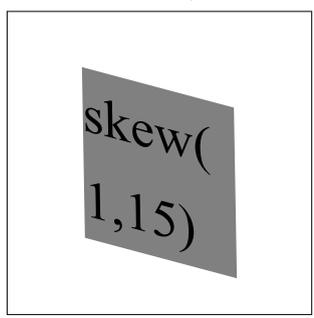
`matrix(1, 2, 0, 1, 0, 0)`



`scale(1.5, 0.5)`



`rotate(45)`



`skew(1, 15)`

図 2.3-3 ブロックの変形

Property Name	axf:transform-origin
Value	[ <percentage>   <length>   left   center   right   top   bottom ]   [ [ <percentage>   <length>   left   center   right ] && [ <percentage>   <length>   top   center   bottom ] ]
Initial	center center
Applies to	transformable objects
Percentages	refer to the size of the bounding box

axf:transform-origin プロパティには変形の基点を指定します。<percentage>と<length>は X 方向、Y 方向のオフセットです。

「あるブロックの囲いを菱形にしたいが文字は変形させたくない」といったときには、たとえばテキストのあるブロックとは別のブロックを用意し、変形後、テキストのあるブロックに位置を重ねることで実現可能です [図 2.3-4] (p.76)。

```

<!-- 菱形の囲いをテキストのブロックに重ねる -->
<fo:block-container ...>
  <fo:block-container width="1cm" height="1cm" >
    <fo:block z-index="1"
      axf:transform="translate(5mm, 5mm), rotate(45)"
      background-color="gray"
      padding-top="0.5cm" padding-bottom="0.5cm">
    </fo:block>
    <fo:block z-index="2"
      axf:transform="translate(5mm, -3.5mm)"
      text-align="center" ...
    > 第...
    </fo:block>
  </fo:block-container>
</fo:block-container>

```



図 2.3-4 菱形の囲いをテキストのブロックに重ねる

#### 注

変形は順に適用されるため、先に位置の移動を指定してから変形を指定すると調整がしやすくなります。

## 2.3.10 行番号を付加する

簡単な構造のブロックや表のセルで、行領域ごとに行番号を付加します。

Property Name	axf:line-number
Value	none   show   hide
Initial	none
Applies to	all block-level-formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes (except for fo:table-column)

axf:line-number プロパティへ show を指定することで行番号を表示し、カウントします。none では表示もカウントも行われません。hide では表示せずにカウントを進めます。

フロートや脚注内では無効です。基本的な制御は<fo:page-sequence>への指定ですが、表の行番号は<fo:table-column>でも制御できる項目が多くあります。

<fo:block>&nbsp;&lt;/fo:block>のような記述もカウントの対象です。

axf:line-number	行番号をカウントする	行番号を表示する
none	-	-
show	✓	✓
hide	✓	-

### 行番号のカウントを制御する

Property Name	axf:line-number-initial
Value	auto   <number>   # <id>
Initial	auto
Applies to	fo:page-sequence, fo:table-column

Property Name	axf:line-number-reset
Value	auto   none   page   column   force
Initial	auto
Applies to	fo:page-sequence, fo:block, fo:table-column

<fo:page-sequence>で、行番号のカウント初期値を axf:line-number-initial プロパティへ指定します。

行番号のカウントをリセットするタイミングを axf:line-number-reset へ指定します。

auto は前のページシーケンスが存在しないとき 1 となり、ほかは初期化を行いません。#<id>は<fo:table-column>のみに指定可能で、指定した id の table-column のカウントを引き継ぎます

force は<fo:block>へのみ指定可能で、指定したブロックで強制的に行番号のカウントをリセットします。

## 行番号の表示を制御する

Property Name	axf:line-number-start
Value	<number>   auto
Initial	auto
Applies to	fo:page-sequence, fo:table-column
Property Name	axf:line-number-interval
Value	<number>   auto
Initial	auto
Applies to	fo:page-sequence, fo:table-column
Property Name	axf:line-number-show
Value	<number>*
Initial	empty
Applies to	fo:page-sequence, fo:table-column

表示する行番号の間隔を axf:line-number-interval プロパティへ指定します。

表示を開始する行番号を axf:line-number-start プロパティへ指定します。

axf:line-number-show プロパティには、axf:line-number-interval などによって出力されない番号でも常に表示する数字を指定します。axf:line-number="hide"のときは表示されません。

## 折り返し時の行番号カウントと表示を抑制する

Property Name	axf:line-number-except-continued-line
Value	true   false
Initial	false
Applies to	all block-level formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes

axf:line-number-except-continued-line プロパティへ true を指定したとき、長い行で折り返しがあつたとき、最初の行のみに行番号を振ります。折り返し後の行（継続行）には行番号は表示されません。false を指定したときは継続行でも行番号が振られます。

たとえばソースコードの例示など、元のテキストの行数を行番号に反映したいときに有用です。

## 2.3.11 行番号のフォントを修飾する

行番号のフォントにブロックと異なる指定をします。

Property Name	<code>axf:line-number-color</code>
Value	<code>&lt;color&gt;</code>
Initial	the value of the 'color' property
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes
Property Name	<code>axf:line-number-font-family</code>
Value	<code>[ &lt;family-name&gt;   &lt;generic-family&gt; ]#</code>
Initial	depends on UA
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes
Property Name	<code>axf:line-number-font-size</code>
Value	<code>&lt;absolute-size&gt;   &lt;relative-size&gt;   &lt;length&gt;   &lt;percentage&gt;</code>
Initial	medium
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes
Percentages	refer to the font size
Property Name	<code>axf:line-number-font-style</code>
Value	<code>normal   italic</code>
Initial	normal
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes

Property Name	<code>axf:line-number-font-weight</code>
Value	<code>normal</code>   <code>bold</code>   <code>bolder</code>   <code>lighter</code>   1... 1000
Initial	<code>normal</code>
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes

`axf:line-number-color` プロパティや `axf:line-number-font-*` プロパティが適用されます。一部の行番号に異なる修飾を指定するには、一度ブロックを終了し別のブロックとして開始する、入れ子のブロックで別のスタイルを指定するなど可能です。

```
<!-- 入れ子で行番号の修飾を変更する -->
<fo:block axf:line-number="show"
  axf:line-number-color="black">
  ...
  <fo:block axf:line-number="show"
    axf:line-number-color="red"
    axf:line-number-font-style="italic" >
    ...
  </fo:block>
</fo:block>
```

## 2.3.12 行番号を修飾する

行番号へテキスト修飾、背景色などを指定します。

### 行番号の書式を指定する

Property Name	<code>axf:line-number-format</code>
Value	<code>&lt;string&gt;</code>
Initial	1
Applies to	<code>fo:page-sequence</code> , <code>fo:table-column</code>
Inherited	yes

行番号の書式を `axf:line-number-format` プロパティへ指定可能です。format プロパティと同様に指定します。

## 行番号に前置文字列を付加する

Property Name	axf:line-number-prefix
Value	<string>
Initial	empty string
Applies to	fo:page-sequence, fo:table-column

行番号の前に出力するテキストを axf:line-number-prefix プロパティへ記述可能です。単純なテキストのみ指定可能です。

## 行番号のテキストを修飾する

Property Name	axf:line-number-text-decoration
Value	same as text-decoration
Initial	none
Applies to	all block level formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes

行番号と axf:line-number-prefix プロパティに記述した内容を、axf:line-number-text-decoration プロパティにより修飾可能です。

## 行番号に背景色を指定する

Property Name	axf:line-number-background-color
Value	<color>
Initial	transparent
Applies to	all block level formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes

行番号領域の背景色を axf:line-number-background-color プロパティへ指定可能です。連続した行に行番号を振るとき、行間の空白にはこの背景色は適用されません。

### 注

行番号が表示される列に対し背景色を置くには、行番号表示と重なる領域の背景に対し背景色を指定するのが良いでしょう。

## 2.3.13 行番号の位置を指定する

行番号の位置を変更、調整します。

Property Name	<code>axf:line-number-position</code>
Value	<code>start</code>   <code>end</code>   <code>inside</code>   <code>outside</code>   <code>alternate</code>
Initial	<code>start</code>
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes

ブロックに対しての行番号の配置を `axf:line-number-position` プロパティへ指定します。  
`alternate` は段組のとき、1 段目は `start`、残りの段では `end` です。

Property Name	<code>axf:line-number-offset</code>
Value	<code>&lt;length&gt;</code>
Initial	<code>0pt</code>
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes

行番号のインライン進行方向のオフセットを `axf:line-number-offset` プロパティへ指定します。

## 2.3.14 行番号領域の幅を指定する・揃えを変更する

行番号の幅を明示し、そのときの揃えを指定します。

Property Name	<code>axf:line-number-width</code>
Value	<code>auto</code>   <code>&lt;length&gt;</code>
Initial	<code>auto</code>
Applies to	all block level formatting objects that are descendants of <code>fo:flow</code> , <code>fo:table-column</code>
Inherited	yes

行番号領域の幅を `axf:line-number-width` プロパティへ指定します。`auto` ではテキスト幅に合わせられます。

Property Name	axf:line-number-text-align
Value	auto   start   center   end   inside   outside   left   right
Initial	auto
Applies to	all block level formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes

行番号のインライン進行方向の揃えを axf:line-number-text-align プロパティへ指定します。axf:line-number-width プロパティが auto でないときに有効です。

Property Name	axf:line-number-display-align
Value	auto   before   center   after
Initial	auto
Applies to	all block level formatting objects that are descendants of fo:flow, fo:table-column
Inherited	yes

行領域内での行番号のブロック進行方向の揃えを axf:line-number-display-align プロパティへ指定します。auto のとき、横組では after、縦組では center とみなされます。

## 2.3.15 行番号の表示を回転する

行番号の表示を回転させます。

Property Name	axf:line-number-orientation
Value	0   90   180   270   -90   -180   -270
Initial	0
Applies to	fo:page-sequence, fo:table-column

axf:line-number-orientation プロパティへ行番号の表示を回転する角度を指定します。指定できる角度は決まっています。

縦組で行番号のみ正立させるときなどに有用です [図 2.3-5] (p.84)。

```
<!-- 縦組で行番号表示を回転させる -->
<fo:page-sequence ...
  axf:line-number-orientation="90"
  axf:line-number-initial="1" axf:line-number-reset="page"
  axf:line-number-font-style="italic" >
<fo:flow>
  <fo:block
    axf:line-number="show"
```

```

axf:line-number-color="cmyk(0,0,0,0.5)"
axf:line-number-font-size="smaller"
axf:line-number-offset="4q"...>エスペラントの話を読みたい...
</fo:block>
</fo:flow>
</fo:page-sequence>

```

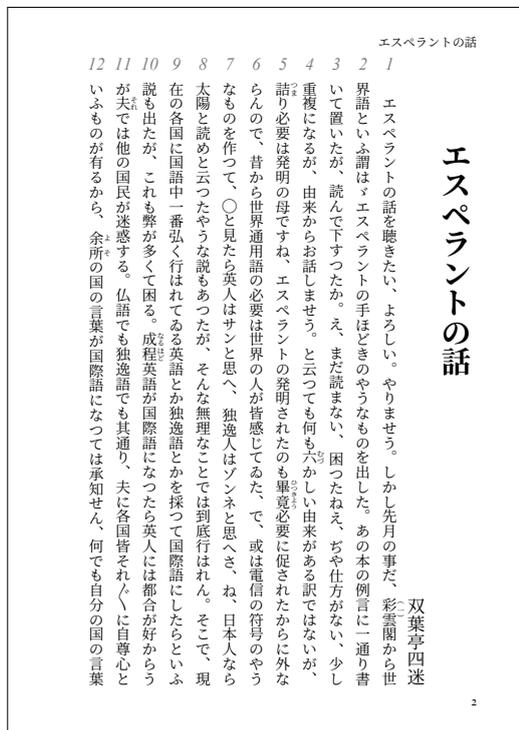


図 2.3-5 縦組で行番号表示を回転させる

## 2.3.16 折り返し行の末尾に文字（行継続マーク）を付与する

折り返しにより行分割が生じた行の末尾に、指定した文字や記号を付加します。

Property Name	axf:line-continued-mark
Value	<string>
Initial	empty string
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

ブロックで axf:line-continued-mark プロパティへ記号を指定することで、行分割が生じた行で、ブロックの外の領域に行継続マークとして表示します。指定が空文字列のときは表示しません。

分割を判定するのが困難でないスクリプトのときのみ有効で、複数種類のスクリプトが混在するブロックや、アラビア語やタイ語などの複雑なスクリプトでは機能しません。

## 2.3.17 行継続マークの位置を調整する

行継続マークが表示される位置を調整します。

Property Name	axf:line-continued-mark-offset
Value	<length>
Initial	0pt
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

axf:line-continued-mark-offset プロパティへ指定した<length>だけ行継続マークのオフセットを移動します。

### 注

ブロックをボーダーで囲うとき、行継続マークをボーダーの内側に表示するのであれば、axf:line-continued-mark-offset プロパティへ負の値を指定します。

## 2.3.18 行継続マークの修飾を指定する

行継続マークのフォントや色を指定します。

Property Name	axf:line-continued-mark-color
Value	<color>
Initial	depends on the current line area
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

Property Name	axf:line-continued-mark-font-family
Value	[ <family-name>   <generic-family> ]#
Initial	depends on the current line area
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

Property Name	<code>axf:line-continued-mark-font-size</code>
Value	<code>&lt;absolute-size&gt;</code>   <code>&lt;relative-size&gt;</code>   <code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>
Initial	depends on the current line area
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes
Percentages	refer to the font size

---

Property Name	<code>axf:line-continued-mark-font-style</code>
Value	<code>normal</code>   <code>italic</code>
Initial	depends on the current line area
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

---

Property Name	<code>axf:line-continued-mark-font-weight</code>
Value	<code>normal</code>   <code>bold</code>   <code>bolder</code>   <code>lighter</code>   <code>1... 1000</code>
Initial	depends on the current line area
Applies to	all block level formatting objects that are descendants of fo:flow
Inherited	yes

## 2.3.19 ブロックに付加する改訂バーに文字や画像を置く

改訂バーの子にフロートを記述し、文字や画像を表示します。

XSL 1.1 仕様における改訂バーは`<fo:change-bar-begin>`と`<fo:change-bar-end>`に挟まれたブロック要素の横に、ページ、段をまたいで表示されるバーを制御するオブジェクトです。このバーにはボーダー修飾が適用可能です。

AH XSL Formatter では`<fo:change-bar-begin>`の子要素としてフロートを配置可能に拡張されています。改訂バーの位置が、この子フロート要素の参照領域の基点となります。

### 注

`change-bar-offset` プロパティを変更したとき、この基点も合わせて変更されます。

`change-bar-placement="alternate"`のとき、段組での改訂バーの位置はページ上の表示位置で決定します。バーにフロートを追従させるには `axf:float-x="alternate"`とします。バーの配置に追従するため、通常フロートでの `alternate` とは動作が異なります。

## 2.3.20 下線、上線、打ち消し線を修飾する

テキストに引く線の修飾を細かく指定します。

text-decoration プロパティへ、underline で下線、upperline で上線、line-through で打ち消し線を指定可能です。XSL 1.1 仕様では、これらの線の修飾を指定する仕様はありません。

Property Name	axf:text-line-color
Value	<color>
Initial	the value of 'color' property
Applies to	all elements with and generated content with textual content
Property Name	axf:text-line-style
Value	<border-style>
Initial	solid
Applies to	all elements with and generated content with textual content
Property Name	axf:text-line-width
Value	auto   <border-width>
Initial	auto
Applies to	all elements with and generated content with textual content

axf:text-line-color プロパティで線色、axf:text-line-style プロパティでボーダーのスタイル、axf:text-line-width プロパティで線幅を指定します。

### 注

下線、上線、打ち消し線について共通で設定されます。それぞれの線で異なる修飾を同じテキストに設定するときは、インライン要素などで入れ子にし、別々に指定するとよいでしょう。

axf:text-line-style へ none は指定できません。「下線を一部だけ消したい」というときは text-decoration="no-underline" の指定などを検討しましょう。

打ち消し線を赤い二重線にする指定例を次に示します [図 2.3-6] (p.88)。

```
<!-- 赤い打ち消し二重線を引く -->
<fo:inline text-decoration="line-through"
  axf:text-line-style="double"
  axf:text-line-color="red"
  axf:text-line-width="0.4em">...</fo:inline>
```

text-line-style プロパティへ double を指定したとき、ボーダーの外側と内側で線の太さを調整する axf:border-double-thickness プロパティなどは利用できません。

~~セミナーは5月26日に開催されます。~~  
セミナーは中止になりました。

図 2.3-6 赤い打ち消し二重線を引く

## 2.3.21 テキストに引く線の位置を制御する

下線や上線の配置を変更します。

Property Name	axf:text-underline-position
Value	auto   [[ before-edge   alphabetic   after-edge ]   [ <percentage>   <length> ]]
Initial	auto
Applies to	all elements with and generated content with textual content
Inherited	yes
Percentages	refers to the line height of the parent area

下線の位置を、axf:text-underline-position プロパティへ指定可能です。auto では自動的に調整されます。

axf:text-underline-position	意味
alphabetic	下線の幅の上端をベースラインに配置。
before-edge	下線の幅の下端を before-edge に配置。
after-edge	下線の幅の上端を after-edge に配置。
<length>   <percentage>	指定した量だけ下線の配置をずらす。

配置とオフセットを両方指定したとき、オフセットは下線の幅の中心からの距離となります。

### 注

縦組のとき、axf:text-underline-position="auto" の下線の配置は axf:vertical-underline-side プロパティで決定されます。このときのオフセットは右側の傍線では左とは逆方向にずらされます。axf:text-underline-position="1pt" のとき、左側の傍線では 1pt 左方向へ、右側傍線では 1pt 右方向へずらされます。

axf:text-underline-position プロパティへ値を指定したとき、上線は下線と反対向きに移動します。たとえば axf:text-underline-position="before-edge" が指定されたとき、text-decoration="overline" はテキストの after-edge に配置されます。

打ち消し線 (text-decoration="line-through") のとき、axf:text-underline-position の値は配置に影響しません。

## 2.3.22 縦組テキストの傍線方向を制御する

縦組のとき下線は傍線として扱われますが、このとき `before` と `after` のどちら側に配置するかを制御します。

Property Name	<code>axf:vertical-underline-side</code>
Value	<code>left</code>   <code>right</code>   <code>depend-on-language</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	<code>vertical-underline-side</code>
Value	<code>left</code>   <code>right</code>   <code>depend-on-language</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	<formatter-settings>

`axf:text-underline-position="auto"` のとき、縦組での傍線をどちらに付けるかを `axf:vertical-underline-side` プロパティで制御可能です。 `text-decolation="upperline"` はこの指定と反対側になります。

`axf:vertical-underline-side` プロパティへ `auto` を指定しているときはオプション設定 `vertical-underline-side` プロパティで決定します。

`vertical-underline-side` プロパティも `auto` のとき、AH XSL Formatter は言語が日本語 (`jpn`)、韓国語 (`kor`) のとき右側、その他の言語では左側への傍線配置を初期値にします。指定されていないときはオプション設定 `default-CJK` プロパティの値によって決定します。 `axf:vertical-underline-side="depend-on-language"` のときも同様です。

明示的に指定するユースケースとして、たとえば「縦組の和文に欧文のフレーズを挟み、その傍線をすべて右側に配置したいとき」 `axf:vertical-underline-side="right"` で実現可能です [図 2.3-7] (p.89)。このとき、欧文を正立させていても下線は右側に配置されます。

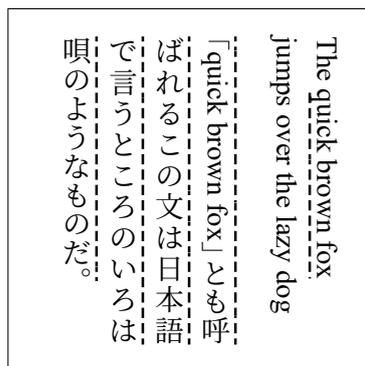


図 2.3-7 縦組での欧文の傍線の例

## 2.3.23 テキストに影を付加する、ぼかしを制御する

テキストに影を付加し、そのぼかしを制御します。

`text-shadow` プロパティにより、テキストへ影の付加が可能です。XSL 1.1 仕様では継承できないプロパティですが、AH XSL Formatter では継承可能に拡張されています。また、適用対象の要素も拡張されています。

Property Name	<code>text-shadow</code>
Value	<code>none</code>   [ <code>&lt;length&gt;{2,3}&amp;&amp; &lt;color&gt;?]</code> #
Initial	<code>none</code>
Applies to	<b>all elements which can have borders</b>
Inherited	<code>yes</code>

`none` のとき影は付加されません。

`<水平方向の影のオフセット>` `<垂直方向の影のオフセット>` `<ぼかし半径>` `<影の色>` を組として、複数の影の付加が可能です。

ぼかし処理を行うとき、テキストの影は透明度を含むラスタ画像形式になります。透明度を含められない出力では、ぼかし処理は行われません。

Option Property Name	<code>textshadow-resolution-minimum-dpi</code>
Value	<code>&lt;integer&gt;</code>
Initial	<code>108</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

Option Property Name	<code>textshadow-resolution-pixel-per-em</code>
Value	<code>&lt;integer&gt;</code>
Initial	<code>100</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

Option Property Name	<code>textshadow-blur-cannot-embed-font</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定で `text-shadow` プロパティのぼかし処理について制御可能です。

`textshadow-resolution-pixel-per-em` プロパティへぼかし処理のときの解像度を指定します。値はフォントサイズ一辺あたりの画素数で指定します。

`textshadow-resolution-minimum-dpi` プロパティへ、ぼかし処理のときの最小 dpi を指定可能です。`textshadow-resolution-pixel-per-em` プロパティから算出された dpi がこの最小 dpi よりも小さいとき、こちらが優先されます。

`textshadow-blur-cannot-embed-font` は埋め込み不可のフォントが指定された箇所での `text-shadow` のぼかし処理を行うかを制御します。true でぼかし処理が行われます。

## 2.3.24 ユースケース：ニューモーフィズムデザインの修飾

ニューモーフィズムデザインのブロック修飾を行います。

複数の種類の影を使用して、要素を単に浮いたように見せるのではなく凹凸として表現します [図 2.3-8] (p.91)。ボーダーの `inset` や `outset` でも凹凸は表現が可能ですが、ボーダーをぼかして表現するのであれば `axf:box-shadow` プロパティが有効です。

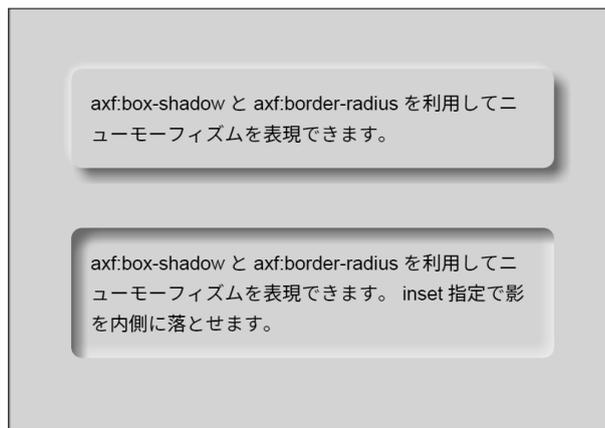


図 2.3-8 ニューモーフィズムデザインのブロック修飾

### ブロックを手前に突き出させる

```
<!-- ブロックを手前に突き出させる -->
<fo:block
  axf:box-shadow="-3px -3px 3px rgba(255,255,255,0.5),
    5px 5px 5px rgba(0,0,0,0.5)"
  margin="10mm" padding="1em"
  axf:border-radius="5pt">
  ...
</fo:block>
```

光源が左上として、要素が突き出ているとします。左上に明るい影、右下に暗い影を配置します。左上は反射光を表現するために、オフセットに負値、色に背景色よりも明るい色を指定します。

`axf:box-shadow` プロパティへ複数の組を指定できるため、続けて、を置いて右下の影を指定しています。背景が画像などであれば、影の色を透過色にすると影が馴染みやすくなります。

影を他のオブジェクトと位置を被らせないようにマージンを取りました。



```

border-end-style="solid" border-start-style="solid"
border-before-style="solid" border-after-style="solid"
border-start-color="gray" border-end-color="gray"
border-before-color="gray" border-after-color="gray"
border-start-width="1.8pt" border-end-width="1.8pt"
border-before-width="1.8pt" border-after-width="1.8pt"

axf:background-clip="border-box"
axf:border-top-left-radius="5pt" axf:border-bottom-right-radius="5pt"
background-image="linear-gradient(to right,gray 10%,
    silver 10%)">
...
</fo:block-container>

```

背景とボーダーのためのブロックコンテナを記述しています。ソースコードを記述するブロックを、このブロックの入れ子にして分離しておくことで、ソースコードの途中で行の処理を変更したときにも柔軟に対応可能にします。widthを本文領域よりやや狭く取りました。

背景を2色に塗り分けるため、background-image プロパティへ linear-gradient() 関数を指定しています。

ページ分割時に見分けがつきやすいように、ブロック左上のボーダーを axf:border-top-left-radius="5pt" で、ブロック右下のボーダーを axf:border-bottom-right="5pt" で丸めています。ページ分割の箇所ではボーダーは描画されないため、丸い角があるのはソースコードの先頭と末尾ということになります。

## キャプションをソースコードの右上に表示する

```

<!-- 背景とボーダーのブロックコンテナ -->
<fo:block-container ...>
<!-- キャプションのブロックコンテナ -->
<fo:block-container
  z-index="10" absolute-position="absolute"
  top="0.2em" right="0">
  <fo:block line-height="1"
    margin-top="0pt" font-size="xx-small"
    text-align="right">
    <fo:inline padding="0.2em"
      background-color="gray" color="white">
      Hello World.go preamble</fo:inline>
    </fo:block>
  </fo:block-container>
...
</fo:block-container>

```

ソースコードのキャプションを右上に表示するため、ブロックコンテナの絶対配置を利用します。ソースコードの幅を指定したブロックコンテナの子要素として絶対配置（`absolute-position="absolute"`）のブロックコンテナを置いたとき、`top="0"`、`right="0"`の点が右端となります。

ブロックコンテナの絶対配置は他のオブジェクトと重なって表示されます。`z-index="10"`で通常のオブジェクトより上に重なるように指定しました。

インライン要素に背景色を指定し、`padding="0.2em"`で背景色の適用範囲を広げています。この分、先に指定した `top="0"` を `top="0.2em"` に下げています。

## ソースコードに行番号と折り返し記号を表示する

```
<!-- ソースコードを記述するブロック -->
<fo:page-sequence ...
  axf:line-number-reset="none" axf:line-number-prefix="line.">
  ...
  <fo:block axf:line-number="show"
    axf:line-number-color="white"
    axf:line-number-font-size="xx-small" axf:line-number-font-style="italic"
    axf:line-number-font-family="serif" axf:line-number-offset="-7mm"
    axf:line-continued-mark="&#x23CE;"
    axf:line-continued-mark-font-size="x-small"
    axf:line-continued-mark-font-style="italic"
    axf:line-continued-mark-font-family="fantasy"
    axf:line-continued-mark-offset="3pt"
    axf:line-continued-mark-color="black"

    font-family="monospace" font-size="smaller" line-height="1.4"
    white-space-treatment="preserve"
    linefeed-treatment="preserve"
    wrap-option="wrap"
    margin-left="9mm"
    padding-top="0.2em" padding-bottom="0.2em">package main
import (
  "fmt"
)
...</fo:block>
</fo:page-sequence>
```

`white-space-treatment` に `preserve`、`linefeed-treatment` に `preserve`、`wrap-option` に `wrap` でスペース、改行の維持と折り返し表示にします。

行番号の開始番号は `axf:line-number-initial` で指定します。`axf:line-number-reset="none"` を指定しているので、行番号は基本的に継続されます。行番号をリセットしたソースコードを表示したいのであれば、`<fo:block>` で `axf:line-number-reset="force"` を指定します。`axf:line-number-prefix` に文字列を指定し、行番号の前に「`lines.`」と表示しました。「前置文字列を変えた行番号を複数使い分けたい」というときにはページシーケンスを切り換えます。

`axf:line-number="show"`によって、ソースコードを記述したブロックの横に行番号を表示しています。`axf:line-number-font-*`で行番号のフォントについて指定しています。

行番号表示の初期位置はブロックの外になりますが、`axf:line-number-offset` プロパティによって、ブロック内部に移動しています。

`axf:line-continued-mark` にテキストを指定することで、行の折り返しがあるとき、このテキストを表示します。例では`☺`を指定しました。`axf:line-continued-mark-font-*`でこの記号に使用するフォントを指定します。

## 注

行番号表示は折り返した行でも行番号のカウントが進みます。

## ソースコードの途中で行番号の修飾を変更する

```
import (<fo:block axf:line-number-color="black"
axf:line-number-background-color="white"
font-family="monospace" "fmt"></fo:block></fo:block>
```

入れ子のブロックを記述し、行番号の文字色と背景色を変更しました。

## 2.3.26 ユースケース：吹き出し会話を作成する

`axf:transform` プロパティとリストブロックを組み合わせ、吹き出し会話風の修飾を行います [図 2.3-10] (p.95)。

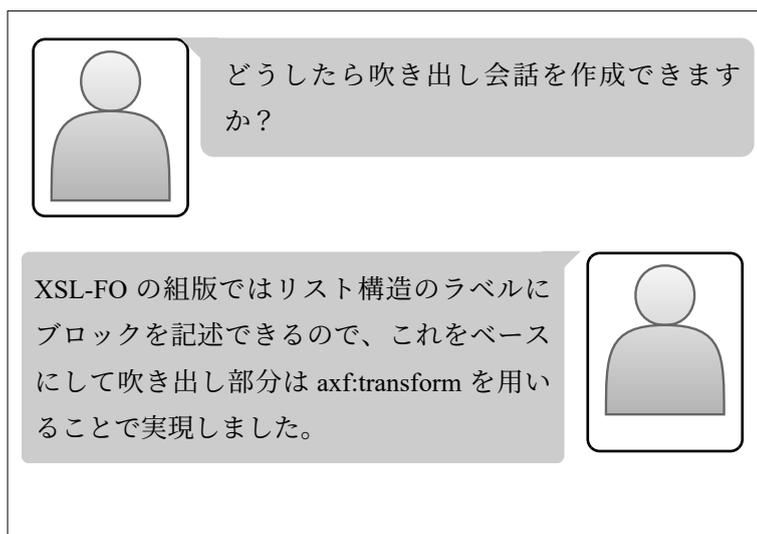


図 2.3-10 吹き出し会話を作成する

## 右側がラベルのリストを作成する

```
<!-- リストブロックの記述 -->
<fo:list-block
  space-before="0.4em" space-end="0.4em">
  <!-- 左がラベルのアイテム-->
  <fo:list-item space-after="0.5em">
    <fo:list-item-label
      start-indent="0mm" end-indent="100vw - 2cm">
      ...
    </fo:list-item-label>
    <fo:list-item-body
      start-indent="2.5cm" end-indent="0">
      ...
    </fo:list-item-body>
  </fo:list-item>
  <!-- 右がラベルのアイテム-->
  <fo:list-item>
    <fo:list-item-label
      start-indent="100vw - 2cm" end-indent="0">
      ...
    </fo:list-item-label>
    <fo:list-item-body
      start-indent="0" end-indent="2.5cm">
      ...
    </fo:list-item-body>
  </fo:list-item>
</fo:list-block>
```

リスト構造はラベル部分のブロックと紐付いたような構造を記述するのに有用です。

リストブロックは start 側にラベル、end 側にボディという配置を想定しているため、右にラベルを配置するときには label-end() や body-start() にインデントの計算を任せることは難しくなります。

ラベルに指定する画像サイズを 2cm であると仮定し、アイテムのラベルとボディの start-indent、end-indent を計算しています。

ラベルの開始位置は、リストブロックの幅とみなせる 100vw を利用し 100vw - 2cm を指定しました。

## ラベルに画像を配置する

```
<!-- ラベル内に外部画像を指定する -->
<fo:list-item-label ...>
  <fo:block axf:border-radius="5pt" border="solid 1pt black">
    <fo:external-graphic src="url(./image.jpg)"
      width="2cm" height="2cm" content-width="scale-to-fit"/>
  </fo:block>
</fo:list-item-label>
```

```
</fo:block>
...
</fo:list-item-label>
```

`<fo:external-graphic>`に `width="2cm"`、`content-width="scale-to-fit"`を指定しました。画像は幅 2 cm に調整されます。

## ブロックコンテナを変形して吹き出しを作る

```
<!-- リストのラベル -->
<fo:list-item-label ...>
  <fo:block ...>
    <fo:external-graphic .../>
  </fo:block>
  <fo:block-container background-color="cmyk(0,0,0,0.2)"
    axf:transform="translate(1.8cm, -2.38cm) skewX(45) scaleX(0.5)"
    width="1cm" height="0.8rem" />
</fo:list-item-label ...>
```

サイズを指定したブロックコンテナを `axf:transform` プロパティで変形し、平行四辺形の吹き出し部分を作成します。ラベル側に吹き出しのオブジェクトを記述したため、ボディが長くなっても吹き出し位置の調整は必要ありません。

`translate()` で位置を移動し、`skewX()` で X 方向に傾きをつけ、`scaleX()` で傾いた分の幅を調整しています。

`<fo:list-item-body>`のブロックと背景色を合わせ、吹き出しを表現しました。吹き出し部をブロックコンテナの変形ではなく SVG などで記述することで、より多様な吹き出しが可能です。

## 2.3.27 ユースケース：改訂バーにリビジョン番号を付加する

改訂バーにその部分のリビジョン番号を示すテキストを付加し、一緒に表示します。

改訂バーにはバーのスタイルや色の指定が可能ですが、ほかに情報を付加したいときがあるでしょう。本文を阻害しないようリビジョン番号の情報を追加します [図 2.3-11] (p.100)。

### 改訂バーとフロートの配置

```
<!-- 改訂バーとフロート -->
<fo:block>...複数用意されることになります。</fo:block>
<fo:block>
  <fo:change-bar-begin ...
    change-bar-class="tr0101"
    change-bar-placement="alternate"... >
    <fo:float axf:float-x="alternate" ... >
      <fo:block ...>R.1</fo:block>
    </fo:float>
```

```
</fo:change-bar-begin>
WordのようなDTPソフトでドキュメントを作る場合、...という仕組みもあります。...
<fo:change-bar-end change-bar-class="tr0101" />
</fo:block>
```

改訂バーの配置は `change-bar-position` プロパティで指定可能ですが、`<fo:change-bar-begin>`の子フロートの配置は別に指定します。このフロートの配置の基点は改訂バーの配置に従います。

改訂バーの外側にフロートを配置するとき、改訂バーの配置指定が `start` や `end` など固定方向であれば、フロートの子としてブロックコンテナを絶対配置で配置する方法が簡便です。

しかし、`change-bar-position="alternate"`のようにページ上の位置によって改訂バーの配置が異なるとき、絶対配置では対応できません。そこでフロートに `axf:float-x="alternate"`を指定すると、改訂バーに合わせた配置を行います。

## 改訂バーの子フロートのフォントを長体にする

```
<!-- コンデンス体があれば利用する -->
<fo:declarations>
  <axf:formatter-config
    xmlns:axe="http://www.antennahouse.com/names/XSL/Settings">
    <axe:font-settings font-stretch-mode="6" />
  </axf:formatter-config>
</fo:declarations>
...
<fo:page-sequence ...>
  <fo:flow ...>
    ...
    <fo:block>...複数用意されることになります。</fo:block>
    <fo:block>
      <fo:change-bar-begin ...>
        <fo:float ... >
          <fo:block font-family="Bahnschrift" font-stretch="semi-condensed"
            ... >R.1
          </fo:block>
        </fo:float>
      </fo:change-bar-begin>Wordのような...
    </fo:block>
    ...
  </fo:flow>
</fo:page-sequence>
```

改訂バーを本文領域の外、行外に配置します。複数の改訂バーが配置されることを考慮し、改訂バーに付随するテキストは長体のフォントを使用するように指定しました。AH XSL Formatter では `font-stretch` プロパティの指定に `condensed` を指定したとき、フォントファミリーに `condensed` が存在すればそれを考慮したフォント選択、変形を行います。

## 改訂バーのオフセットを指定する

```
<!-- 改訂バーとフロートの位置を指定する -->
<fo:change-bar-begin change-bar-class="tr0101"
  change-bar-placement="alternate" change-bar-width="4pt"
  change-bar-offset="10pt - 4pt">
  <fo:float
    axf:float-x="alternate" axf:float-offset-x="-42pt" >
    <fo:block ...>R.1</fo:block>
  </fo:float>
</fo:change-bar-begin>
```

改訂バーとフロートによる画像やテキストが重なって表示しないようにするため、オフセットを変更します。改訂バー同士が重ならないように `change-bar-offset` プロパティを指定します。フロートのオフセットは `axf:float-offset-x` プロパティで変更します。

`change-bar-placement="alternate"` や `axf:float-x="alternate"` のとき、オフセットはそれぞれ配置方向からの変量となります。

## DITA とは

DITA とは、XML の技術を応用した、マニュアルや書籍等のコンテンツを執筆するための仕様です。この仕様に則って書かれたコンテンツから印刷物（PDF）や HTML、Web ヘルプ等の配布可能な媒体を効率よく作成することを目的としています。DITA はコンテンツを最小単位に切り分けたトピックファイル、そしてコンテンツの構造を決めるマップファイルで構成されており、ドキュメントのコンポーネント化、システム化を志向しています。

## トピックファイル

トピックファイルとは、1つの話題のみ書かれたファイルです。たとえば「電源の入れ方」とか「電源の切り方」といった話題毎にトピックファイルを作成することになります。そのため、トピックファイルは必然的に複数用意されることとなります。

Word のような DTP ソフトでドキュメントを作成する場合、一般的には1つのファイルに全ての話題を詰め込むこととなりますが、DITA の場合、話題毎にトピックファイルに分割します。その際、ドキュメントを単に分割すればいいというわけではなく、情報構造をあらかじめ整理することが極めて重要です。執筆する話題に合わせて、concept（概要）、task（手順）、reference（参考書類）、troubleshooting（トラブルシューティング）等のトピックタイプ（情報タイプ）が標準で用意されています。これらの情報タイプだけではどうしてもうま

く表現できない（あるいは執筆する際にもっと強い制約が欲しい）場合のために特殊化という仕組みもあります。

## マップファイル

マップファイルはドキュメントの構造を決める役割を持っています。具体的には、トピックファイルの出力順やトピック間の階層構造です（目次に相当する情報と言ってもいいでしょう）。トピックファイルとマップファイルは完全に分離されている為、マップファイルを変更する事で、コンテンツであるトピックファイルに手を加えることなくドキュメントの構造を変化させることができます。

R.1

R.2

R.2 R.1

図 2.3-11 改訂バーにリビジョン番号を付加する

## 2.4 節 段組

ページ内で異なる段組の箇所や段間の修飾、部分的な段抜きの指定が可能です。

### 2.4.1 同じフローに異なる段組を混在させる

ブロックコンテナに段組を指定し、参照する区画を変えずに複数の段組を混在させます。

XSL 1.1 仕様では、段数を指定するための `column-count` や段間のアキを指定する `column-gap` プロパティは `<fo:region-body>` へ指定します。

AH XSL Formatter では `<fo:block-container>` へ `column-count` プロパティを指定可能です。これを利用し、ひとつのフロー内で複数種類の段組が可能です。

```
<!-- 異なる段組の混在の例 -->
<fo:page-layout-master>
  <fo:simple-page-master ...>
    ... <!-- 本文2段組 -->
    <fo:region-body
      column-count="2" column-gap="8mm" ... />
  </fo:simple-page-master>
  ...
</fo:page-layout-master>
<fo:page-sequence ...>
  ...
  <fo:flow ...>
    <!-- 2段組 -->
    <fo:block
      <fo:block-container
        span="all"
        column-count="3" column-gap="3mm" ...>
      <!-- 3段組 -->
      <fo:block ...>...</fo:block>
    </fo:block-container>
    <!-- 2段組 -->
    ...
  </fo:flow>
</fo:page-sequence>
```

### 2.4.2 段ごとのコンテンツの高さを揃える

先頭の段から段を埋めるようにコンテンツを配置していくと、最終段はほかの段よりも余白が生じることがあるでしょう。段ごとのコンテンツの量を調整し、段の高さをある程度揃えるよう指定します。

Property Name	<code>axf:column-fill</code>
Value	<code>auto</code>   <code>balance</code>   <code>balance=att</code>
Initial	<code>balance</code>
Applies to	multi-column elements

`axf:column-fill` プロパティは [CSS3-Multicol] ‘column-fill’を基にしたプロパティです。balance を指定したとき、コンテンツの高さがその段組内で同じ程度になるように調整（バランス）されます。段組が複数ページ続くとき、最終ページだけがバランスの対象となります。

`auto` では、`span="all"`の段抜き要素によって段が遮られているときのみ、段抜きの前でバランスされます。

[CSS3-Multicol] ‘column-fill’では、このほかに `balance-all` がありますが、AH XSL Formatter 拡張仕様では対応していません。

## 2.4.3 段間罫を引く

段間への罫線（段間罫）を指定します。

Property Name	<code>axf:column-rule-color</code>
Value	<code>&lt;color&gt;</code>
Initial	the value of the ‘color’ property
Applies to	<code>fo:region-body</code> , <code>fo:block-container</code>

Property Name	<code>axf:column-rule-style</code>
Value	<code>&lt;border-style&gt;</code>
Initial	<code>none</code>
Applies to	<code>fo:region-body</code> , <code>fo:block-container</code>

Property Name	<code>axf:column-rule-width</code>
Value	<code>&lt;border-width&gt;</code>
Initial	<code>medium</code>
Applies to	<code>fo:region-body</code> , <code>fo:block-container</code>

`axf:column-rule*` プロパティによる段間罫は、`column-gap` プロパティによって指定される段間の中央に引かれます。

## 2.4.4 段間罫の長さ と揃えを指定する

段の高さはパディングボックスの内側で計算されます。段間罫の長さを段の高さとは別に指定します。また、罫線の長さを変えたとき、端をブロック進行方向のどこへ揃えるかを指定します。

Property Name	axf:column-rule-length
Value	<length>   <percentage>
Initial	100%
Applies to	fo:region-body, fo:block-container
Percentages	refer to the height of the column

段間罫の長さを axf:column-rule-length へ指定可能です。段抜き要素によって段が途中で終わっているときは、その部分で 100%とみなされます。

Property Name	axf:column-rule-align
Value	before   center   after
Initial	center
Applies to	fo:region-body, fo:block-container

段間罫の長さを axf:column-rule-length プロパティで 100%以外に指定したとき、段間罫を before 側、after 側、center のどこへ揃えるかを axf:column-rule-align で制御可能です。

## 2.4.5 段の内容が存在しない箇所で段間罫を引く

段間罫を引くかどうか、コンテンツの有無によって動作を制御します。

段組では、最終ページなど、コンテンツが段組の最後の段まで満たないときがあります。

Property Name	axf:column-rule-display
Value	gap   end   all
Initial	gap
Applies to	fo:region-body, fo:block-container

段間罫が指定されているとき、コンテンツの存在しない（段間とならない）段にも段間罫を引くかどうかを axf:column-rule-display プロパティで指定可能です。

axf:column-rule-display	意味
gap	段間が存在するときのみ段間罫を引く。
end	もともと end 側の段を除き、段の end 側に段間罫を引く。
all	段の内容の有無にかかわらず段間罫を引く。

## 2.4.6 段によって要素の表示位置を変更する

段組で、段によって配置や揃えが変わる指定値が存在します。

改訂バーや行番号の配置を指定する \*-position プロパティへ alternate を指定したとき、段組では段によって配置が変わります。

横 3 段組であれば左から start、start、end となります。偶数段でも均等に分かれるのではなく、横 4 段組であれば start、start、start、end となります。

フロートの X 方向の配置を指定する `axf:float-x` プロパティで指定可能な `alternate` は中央に要素を揃えるときがあり、ブロック修飾などのオブジェクトでの `alternate` の配置とは動作が異なります。改訂バーでは自身の `alternate` 配置と子フロートの `alternate` 配置がありますが、段組では改訂バーの配置に合わせて調整されます。

## 2.5 節 表組

表内要素の配置方法、分割などを柔軟に制御します。

### 2.5.1 ページ、段分割でヘッダ、フッタの動作を変更する

ページや段で表が分割される時のヘッダ、フッタの動作制御が拡張されています。

1 段であるなら改ページ、段組内で表を用いるのであれば改ページや改段といったタイミングで、表の要素は分割される可能性があります。

Property Name	table-omit-header-at-break, table-omit-footer-at-break
Value	true   false   column
Initial	false
Applies to	fo:table

表の分割が起こるとき、そのヘッダ、フッタをどう表示するかは、それぞれ `table-omit-header-at-break`、`table-omit-footer-at-break` プロパティへ指定可能です。

XSL 1.1 仕様では、ページ分割時に表示を省略するか (`true`) しないか (`false`) のみが指定可能です。

AH XSL Formatter は段組での制御が追加されています。`column` を指定したとき、表のヘッダ、フッタは、段分割では省略し、ページ分割では表示します。

### 2.5.2 表のヘッダ、フッタの注の繰り返しを制御する

表分割のとき、表のヘッダ、フッタに記述された注を分割の前後で繰り返すかを指定します。

表のヘッダ、フッタを分割の前後で繰り返すかは `table-omit-*-at-break` プロパティの指定で制御可能です。分割前後で表のヘッダ、フッタを表示するとき、AH XSL Formatter ではそこで記述されている注を繰り返すかどうかを制御可能です。

Property Name	axf:repeat-footnote-in-table-header
Value	true   false
Initial	true
Applies to	fo:table-header
Inherited	yes

Property Name	axf:repeat-footnote-in-table-footer
Value	true   false
Initial	true
Applies to	fo:table-footer
Inherited	yes

ページ、段分割時で表のヘッダ、フッタを表示するとき、ヘッダ、フッタ内の脚注が分割前後で繰り返し表示されるかを axf:repeat-footnote-in-table-\* プロパティへ指定可能です。

それぞれ true で繰り返し表示、false で非表示となります。

ユースケースとしては、段組で段ごとに表のヘッダまたはフッタを表示するとき axf:repeat-footnote-in-table-\*="false" を指定しておくことで、同じ脚注をページ内に繰り返し表示することを回避します。

### 注

axf:footnote-position="column" により段ごとに脚注があるとき、axf:suppress-duplicate-footnote="true" は同じ段内でのみ注の重複表示を抑制します。このとき、ページ内では同じ注が複数回表示されることがあります。

## 2.5.3 セル分割時のコンテンツを制御する

セルの途中で分割が生じるとき、分割後のセルの内容を変更します。

同じ row でほかのセルが分割、または number-rows-spanned で複数の row にまたがったセルで分割が起こるとき、分割されたセルの後ろ側は何も表示されません。

AH XSL Formatter では、ページ、段分割前後でのセルの表示制御が可能です。また、セルの分割が起こるときのみ、別の要素を分割後のセルに表示させることも可能です。

Property Name	axf:repeat-cell-content-at-break
Value	true   false
Initial	false
Applies to	fo:table-cell

axf:repeat-cell-content-at-break プロパティへ、セルの分割時、セルのコンテンツを分割の前で繰り返し表示するかを指定可能です。

axf:repeat-cell-content-at-break="true" のとき、セルのコンテンツをコピーし、分割後のページ、段のセルにも表示します。

Element Name	<axf:table-cell-repeated-marker>
Parent	fo:table-cell

<axf:cell-repeated-marker>に表示させたいコンテンツを記述し、<fo:table-cell>の最初の子要素として配置すると、axf:repeat-cell-content-at-break="true"のとき、記述した内容が分割後のセルに表示されます。

<axf:cell-repeated-marker>の内容は<fo:table-cell>のほかの子要素の内容とは独立しています。セルが分割される時、分割された前側のセルに「列タイトル」、後ろ側のセルに「列タイトル (続き)」と表示したいとします。<axf:cell-repeated-marker>の子の記述では「列タイトル」を含め、「列タイトル (続き)」を記述します。

```
<!-- セル分割時、後ろのセルに「列タイトル (続き)」を表示する -->
<fo:table-cell ... number-rows-spanned="2"
  axf:repeat-cell-content-at-break="true">
  <axf:table-cell-repeated-marker>
    <fo:block>列タイトル (続き) </fo:block>
  </axf:table-cell-repeated-marker>
  <fo:block>列タイトル</fo:block>
</fo:table-cell>
```

## 2.5.4 幅の自動計算に利用する行数を制限する

表の行数が多いときなどに、表幅の自動計算に制限をかけたり、より行数が多いときに自動計算で考慮する行数を増やします。

<fo:table>要素の table-layout プロパティでは、表のカラム幅を固定 (fixed) にするか自動計算 (auto) にするかを指定します。fixed を指定すると組版が高速になります。ただし、このとき内容が指定された幅を超えると、セルの幅をはみ出して表示することがあります。あらかじめセルの内容とその幅が指定できるときに利用するとよいでしょう。

table-layout="auto"では、XSL-FO プロセッサが表の内容を調べてそれぞれの幅を算出します。ひとつの行だけでなく複数の行を調べた上での幅を算出するため、調べる行数が増えるほど計算に時間を要します。詳細はマニュアルをご覧ください。

自動計算のために内容を調べるカラムの行数は AH XSL Formatter の初期値は 100 行ですが、この値の変更が可能です。

Property Name	axf:table-auto-layout-limit
Value	auto   <integer>
Initial	auto
Applies to	fo:table
Inherited	yes

Option Property Name	table-auto-layout-limit
Value	<integer>
Initial	100
Applies to	<formatter-settings>

表組での幅の自動計算に利用する行数をオプション設定 `axf:table-auto-layout-limit` プロパティへ指定します。`axf:table-auto-layout-limit="auto"` のとき、オプション設定 `table-auto-layout-limit` プロパティへ指定した値となります。

## 2.5.5 セル内でのテキストの揃えを制御する

`<fo:table-cell>`要素で `text-align` プロパティへ<string>を指定すると、各行のテキストは指定した文字の位置に揃えられます。

<string>の指定は主に数値表示を小数点の位置に揃えるときなどに有用ですが、このとき、揃えた文字列全体をどちらに揃えるかについては XSL 1.1 仕様に規定がありません。AH XSL Formatter では end 側に揃えますが、この動作は変更が可能です。

Property Name	axf:text-align-string
Value	start   center   end   inside   outside   left   right
Initial	end
Applies to	fo:table-cell
Inherited	yes

`axf:text-align-string` プロパティへ揃え位置を指定することで、`text-align="<string>"` で <string>の位置に揃えられた各行のテキスト全体が揃えられます。

## 2.5.6 セル内のテキストをブロック進行方向で均等割り付けにする

Property Name	display-align
Value	auto   before   center   after   justify
Initial	auto
Applies to	block-level objects
Inherited	yes

`<fo:table-cell>`で `display-align` プロパティへ `justify` を指定することで、セル内のテキストをブロック進行方向に均等割り付けにします。`rowspan` で高さに差があるセルのテキストや、縦横両方向の見出しとなるセルなどで有用です。

## 2.5.7 rowspan したセルがある表の高さを揃える

表に行をまたぐ (rowspan された) セルがあるとき、この結合されたセルを含む各行のセルの高さは同じになるとは限りません。

Option Property Name	justify-rowspan-height
Value	true   false
Initial	false
Applies to	<formatter-settings>

オプション設定 justify-rowspan-height プロパティへ true を指定したとき、結合セルを含む行の高さがある程度同じになるよう調整されます。

高さを揃える対象のセルの内容によっても、どの程度揃えられるかが変わります。高さが指定されている行やセル、文字方向の回転したセルが存在する表では、その行より後ろだけが均等になるよう処理されます。

また、高さの指定されている<fo:table>も処理対象です。

## 2.5.8 表のヘッダ、フッタで複数行を含むマーカを取り出す

表組のヘッダ、フッタで参照するマーカが複数の行を含むとき、参照する最大行数を指定することで、参照する行数に由来する望ましくない表示を回避します。

たとえばヘッダに rowspan による大項目と、より細かい小項目があるようなとき、複数行を含むマーカを記述し、取り出すことがあるでしょう。AH XSL Formatter では表組を描画するとき、それぞれの行は行数に関連した管理が行われています。

<fo:marker>には複数の<fo:table-row>が含まれます。そのマーカを参照し取り出すときに、管理上の行数に矛盾が生じることがあります。

Property Name	axf:retrieve-table-rows
Value	<integer>
Initial	1
Applies to	fo:retrieve-table-marker

<fo:retrieve-table-marker>が取り出す最大行数を axf:retrieve-table-rows プロパティへ指定可能です。指定された最大行数を基準にすることで、行数に関連する管理上の矛盾を回避し、正しく表示します。

```
<!-- 複数行を含むマーカを取り出す -->
<fo:table>
  <fo:table-header>
    <fo:retrieve-table-marker retrieve-class-name="multiple"
      axf:retrieve-table-rows="3"/>
```

```

</fo:table-header>
...
<fo:marker marker-class-name="multiple">
  <fo:table-row>...</fo:table-row>
  <fo:table-row>...</fo:table-row>
  <fo:table-row>...</fo:table-row>
</fo:marker>
...
</fo:table>

```

## 2.5.9 ユースケース：修飾した表を作成する

### 対角罫、縦組の見出し

対角罫のユースケースとして、見出しがふたつの方向にある表組を作成しました。[図 2.5-1] (p.112)

```

<!-- 行をまたぐセルの高さを揃える -->
<fo:declarations>
  <axf:formatter-config
    xmlns:axs="http://www.antennahouse.com/names/XSL/Settings">
    <axs:formatter-settings justify-rowspan-height="true" />
  </axf:formatter-config>
</fo:declarations>

```

rowspan がある表なので、セルの高さをある程度揃えるためにオプション設定 justify-rowspan-height="true"を指定しています。

```

<!-- 対角罫を含む見出しと縦組の見出し -->
<fo:table padding="4mm" table-layout="fixed"
  table-omit-header-at-break="page"
  display-align="center" border="1pt solid k100">
<fo:table-column
  column-number="1" border="0.4pt solid k100" margin="2mm" />
<fo:table-column
  column-number="2" column-width="3rem" border="0.4pt solid k100" />
...
<fo:table-header
  border="0.4pt solid k100">
  <fo:table-row height="7rem">
    <fo:table-cell padding="3mm"
      axf:diagonal-border-style="dashed"
      axf:diagonal-border-width="0.4pt"
      axf:diagonal-border-color="k100"
      axf:scope="both">
      <fo:block text-align="end">Name</fo:block>
      <fo:block text-align="start">License</fo:block>
    </fo:table-cell>
  </fo:table-row>
</fo:table-header>

```

```

</fo:table-cell>
<fo:table-cell >
  <fo:block-container writing-mode="tb-rl"
    block-progression-dimension="3rem">
    <fo:block>アンテナ太郎</fo:block>
  </fo:block-container>
</fo:table-cell>
...
</fo:table-row>
</fo:table-header>
...
</fo:table>

```

`display-align="justify"`によって、セルの縦方向の揃えを均等揃えに指定しました。

`<fo:table-column>`要素の幅指定を `column-width` プロパティで指定します。また、セルの内部で正立させたブロックコンテナの幅を指定するために `block-progression-dimension` プロパティを指定しています。

左上のセルは縦方向、横方向両方の見出しであることを示すために対角罫と揃え位置の変更を指定しました。内容セルでも対角罫を利用するため、見出しでは `axf:diagonal-border-style="dashed"` として破線にしています。どちら方向の見出しかを見やすくすることを意図し、対角罫に加え、縦方向と横方向の見出しの縦方向の揃え位置を `axf:display-align="justify"` に指定しています。

```

<!-- セルに両方向の対角罫を指定する -->
<fo:table>
  <fo:table-body>
    <fo:table-row height="2em" padding="3mm" border="0.3pt solid k100">
      <fo:table-cell text-align="left" margin="3mm"
        padding-left="3mm" padding-right="3mm">
        <fo:block>AH XSL Formatter</fo:block>
      </fo:table-cell>
      <fo:table-cell number-rows-spanned="2"
        axf:diagonal-border-width="0.3pt"
        axf:diagonal-border-style="solid"
        axf:diagonal-border-color="k100"
        axf:reverse-diagonal-border-width="0.3pt"
        axf:reverse-diagonal-border-style="solid"
        axf:reverse-diagonal-border-color="k100">
        <fo:block> </fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>
          <fo:character font-family="fantasy" character="&#x2713;" />
        </fo:block>
      </fo:table-cell>
    ...
  </fo:table-row>

```

```
...
</fo:table-body>
</fo:table>
```

axf:diagonal-border-\*, axf:reverse-diagonal-border-\*プロパティへの指定により、セルに「×」印を描画しました。このFOでは、number-rows-spanned="2"を指定し、2列をまたぐセルにしています。

## 注

rowspan や colspan を含むような複雑な表組で意図した背景色を適用するには、セルごとに背景色の指定を行うのが確実でしょう。こうした結合セルの内部で「背景色をふたつに分けたい」といったときには、たとえば、結合セルの背景画像へ linear-gradient() による色指定を行う方法があります。

<fo:character character="&#x2713;" />でチェックマークを表しています。

Name	アンテナ太郎	アンテナ花子	John Smith	Jane Doe
License	×	×	×	×
AH XSL Formatter	×	✓	✓	×
AH CSS Formatter	×	×	✓	✓
AH Formatter	✓	×	×	×

図 2.5-1 対角罫、縦組の見出し

## 数値を小数点で揃えた表

数値が並ぶ表を作成しました [図 2.5-2] (p.113)。

```
<!-- 数値を小数点で揃えた表 -->
<fo:table ... table-omit-header-at-break="false">
  <fo:table-column border="0.3pt solid k100"/>
  <fo:table-column border="0.3pt solid k100"/>
  <fo:table-column border="0.3pt solid k100"/>
  <fo:table-header
  >
    <fo:table-row text-align="center"
      border="0.3pt solid k100">
      <fo:table-cell padding="1mm">
        <fo:block>製品名</fo:block>
      </fo:table-cell>
      <fo:table-cell display-align="justify" ...>
```

```

    <fo:block>処理1</fo:block>実行時間（秒） </fo:block>
  </fo:table-cell>
  ...
</fo:table-row>
</fo:table-header>
<fo:table-body>
<fo:table-row background-color="cmyk(0,0,0,0.3)">
  <fo:table-cell padding="1mm" font-size="80%">
    <fo:block>製品A</fo:block>
  </fo:table-cell>
  <fo:table-cell padding="1mm"
    text-align=".'" axf:text-align-string="end">
    <fo:block font-variant="tabular-nums">
      800.00</fo:block>
    </fo:table-cell>...
</fo:table-row>
<fo:table-row background-color="cmyk(0,0,0,0.1)">
  <fo:table-cell ...>
    <fo:block>製品B</fo:block>
  </fo:table-cell>
  ...
</fo:table-row>
...
</fo:table-body>
</fo:table>

```

セルで text-align プロパティへ<string>を指定するとき、<string>で揃えたテキスト自体の揃えを指定する axf:text-align-string プロパティは、<string>に小数点などを想定しています。

ブロック進行方向は基本的に display-align="center"として中央に揃えています。内容が2行になるヘッダのセルは display-align="justify"としています。

内容の小数点を揃え位置の基準としたとき、文字の端がバラバラになり隣同士の項目は少し見づらくなってしまうため、項目ごとの罫線を指定しています。横方向の罫線は見出し行のみで、残りは背景色を分けています。

font-variant="tabular-nums"により、数字の並びが整うように設計された字形を明示的に指定しています。

製品名	処理 1 実行時間（秒）	処理 2 実行時間（秒）
製品 A	800.00	8.01
製品 B	10.112	220.3
製品 C	1000.1	1000.118

図 2.5-2 数値を小数点で揃えた表

## 2.6 節 リーダ

リーダーの揃え方をより柔軟に指定します。

### 2.6.1 リーダの揃えを制御する

リーダーの揃え方を制御するプロパティが拡張されています。

AH XSL Formatter では、`leader-alignment` プロパティへ指定可能な値が XSL 1.1 仕様から拡張されています。

Property Name	<code>leader-alignment</code>
Value	<code>none</code>   <code>reference-area</code>   <code>page</code>   <code>start</code>   <code>center</code>   <code>end</code>
Initial	<code>none</code>
Applies to	<code>fo:leader</code>
Inherited	<code>yes</code>

`start`、`center`、`end` のいずれも、`leader-pattern="dots"`または `leader-pattern="use-content"`で内容がテキストのときにのみ、リーダーを指定に揃えます [図 2.6-1] (p.114)。

`page` は要求されないため、AH XSL Formatter では用意していません。

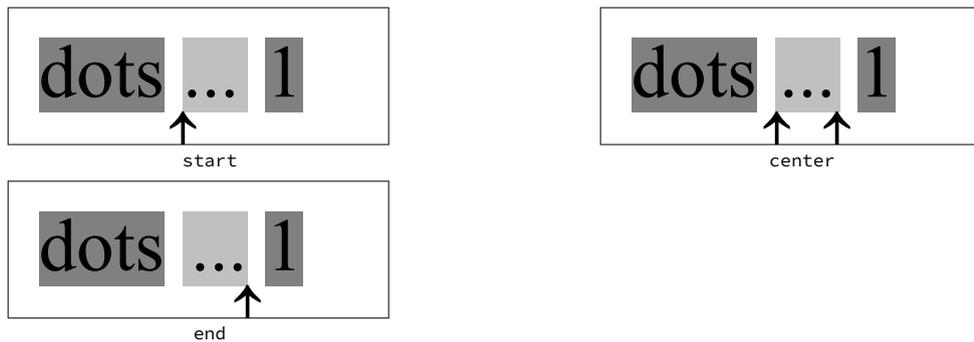


図 2.6-1 leader-alignment

### 2.6.2 リーダ自体を両端揃えにする

いくつかの条件を満たしたとき、リーダー自体の端を揃えます。

`leader-alignment` プロパティはリーダーの端をどう揃えるかについて制御しますが、リーダーに使用している点またはテキストの間隔はそのままです。

Option Property Name	justify-leader
Value	true   false
Initial	false
Applies to	<formatter-settings>

leader-alignment="none"かつ leader-pattern プロパティの指定が dots または use-content のとき、オプション設定 justify-leader プロパティに true を指定すると、条件を満たすリーダを両端にアキがないように均等に揃えます [図 2.6-2] (p.115)。

justify-leader="true"のとき、テキスト間を調整して端のアキを解消します。代わりに、行の異なるリーダ同士で見た目が揃わなくなることがあります。

### 注

justify-leader プロパティの指定は文書全体に影響しますが、それぞれのリーダが leader-alignment="none"でなければ処理に影響しません。「連続する複数行でリーダを記述するときは leader-alignment プロパティへ none 以外を指定する」といった運用も考えられます。

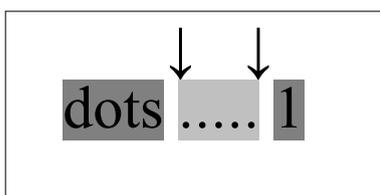


図 2.6-2 リーダを両端揃えにする

## 2.6.3 リーダのある行を均等割り付けにする

リーダが存在する行でのみテキストの揃え方を変更します。

Property Name	axf:leader-expansion
Value	auto   force
Initial	auto
Applies to	fo:block
Inherited	yes

axf:leader-expansion プロパティへ force を指定したとき、リーダが存在する行の揃えを justify にします。

たとえば、目次項目などリーダを挟んだ両端にコンテンツが存在し、端のコンテンツの長さが一定でないときに有用です。ほか、リーダが 2 行にまたがるときや、索引でページ番号が 2 行にまたがるときなどに有用です。

## 2.7 節 リスト

リスト構造について直接拡張する仕様はありませんが、拡張仕様を応用することでより柔軟なリスト表現が可能です。

### 2.7.1 ユースケース：箇条書きを段組にする

段組にした箇条書きで一部を段抜きにします。

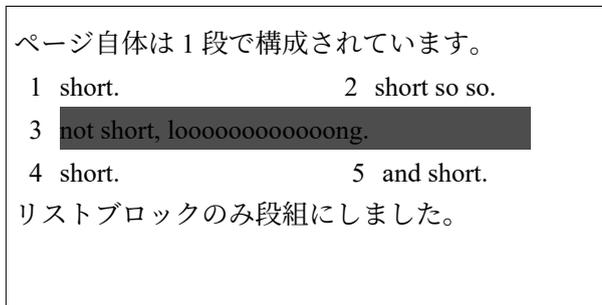


図 2.7-1 箇条書きのみ段組にする

段組をせず箇条書きを記述するとき、項目の長さによっては紙面に大きな空白が生じます。ブロックコンテナでの段組とリストブロックの段抜きを組み合わせ、項目の長さに合わせた箇条書きを作成しました [図 2.7-1] (p.116)

#### 箇条書きのみ段組にする

```
<!-- ブロックコンテナへ段組を指定する -->
<fo:block-container
  column-count="2" column-gap="5mm">
  <fo:list-block
    provisional-distance-between-starts="4mm"
    provisional-label-separation="2mm">
    ...
  </fo:list-block>
</fo:block-container>
```

基本版面が 1 段組、短い項目が並ぶような箇条書きを 2 段組とします。<fo:block-container>に column-count="2" を指定し、<fo:list-block> をその子要素にします。ラベルの開始位置は段間のアキを考慮して指定します。

#### 一部の箇条書きのみ段抜きにする

```
<!-- 段抜きのリストブロック -->
<fo:list-block
  span="all">
```

```

... >
<fo:list-item>
  <fo:list-item-label
    start-indent="2mm"
    end-indent="label-end()">
    <fo:block>3</fo:block>
  </fo:list-item-label>
  <fo:list-item-body
    start-indent="body-start()"
    end-indent="1cm">
    <fo:block background-color="red">not short, ...</fo:block>
  </fo:list-item-body>
</fo:list-item>
</fo:list-block>
<fo:list-block ...>
...
</fo:list-block>

```

一部の箇条書きのみ段抜きとしたいとき、一度<fo:list-block>を終了し、新たな<fo:list-block>に span="all"を指定することで、全段抜きの箇条書きとします。

全段抜きの箇条書きの項目が終了したとき、この<fo:list-block>を終了し再び span="all"のない箇条書きを開始します。

## 2.8 節 グラフィック（色、画像、背景、数式）

多彩な色表現や画像形式、描画方法の制御に対応しています。

### 2.8.1 色を指定する

さまざまな指定方法で目的の色を指定します。

どのように色を表現するかを決めたカラースペースによって、表現可能な色の範囲は異なります。ディスプレイ上で色は Red、Green、Blue（RGB）の加法混色によって表現されますが、インクによる印刷では Cyan、Magenta、Yellow、Keyplate（CMYK）による減法混色によって表現されます。また、特殊なインクなど、通常のカラースペースでは表現できない色をスポットカラーとして指定することもあります。

XSL 1.1 仕様では `<color>` の値は RGB の指定、基本 16 色、`system-color()` 関数、`rgb-icc()` 関数によって指定します。AH XSL Formatter では CMYK や HSL、透明度を含む色など、指定可能な値が拡張されています。

### 2.8.2 色名で指定する

RGB カラースペースでの指定としては `rgb()` 関数による指定、`#RRGGBB` による指定、色名による指定があります。AH XSL Formatter では、指定可能な色名が拡張されています。

基本 16 色	aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
拡張 131 色 と rebeccapurple	aliceblue, antiquewhite, aquamarine, azure, beige, bisque, blanchetalmond, blueviolet, brown, burlywood, cadetblue, chartreuse, chocolate, coral, cornflowerblue, cornsilk, crimson, cyan, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkgrey, darkkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkslategrey, darkturquoise, darkviolet, deeppink, deepskyblue, dimgray, dimgrey, dodgerblue, firebrick, floralwhite, forestgreen, gainsboro, ghostwhite, gold, goldenrod, greenyellow, grey, honeydew, hotpink, indianred, indigo, ivory, khaki, lavender, lavenderblush, lawngreen, lemonchiffon, lightblue, lightcoral, lightcyan, lightgoldenrodyellow, lightgray, lightgreen, lightgrey, lightpink, lightsalmon, lightseagreen, lightskyblue, lightslategray, lightslategrey, lightsteelblue, lightyellow, limegreen, linen, magenta, mediumaquamarine, mediumblue, mediumorchid, mediumpurple, mediumseagreen, mediumslateblue, mediumspringgreen, mediumturquoise, mediumvioletred, midnightblue, mintcream, mistyrose, moccasin, navajowhite, oldlace, olivedrab, orange, orangered, orchid, palegoldenrod, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, peru, pink, plum, powderblue, rosybrown, royalblue, saddlebrown, salmon, sandybrown, seagreen, seashell, sienna, skyblue, slateblue, slategray, slategrey, snow, springgreen, steelblue, tan, thistle, tomato, turquoise, violet, wheat, whitesmoke, yellowgreen, rebeccapurple
28 のシステム色名	ActiveBorder, ActiveCaption, AppWorkspace, Background, ButtonFace, ButtonHighlight, ButtonShadow, ButtonText, CaptionText, GrayText, Highlight, HighlightText, InactiveBorder, InactiveCaption, InactiveCaptionText, InfoBackground, InfoText, Menu, MenuText, ScrollBar, ThreeDDarkShadow, ThreeDFace, ThreeDHighlight, ThreeDLightShadow, ThreeDShadow, Window, WindowFrame, WindowText

基本 16 色 XSL 1.1 仕様で規定されている色です。

拡張 131 色 CSS3 Color で「Extended Color」となっている色です。Extended Color には基本 16 色に入る色も含まれます。

28 のシステム色名 Deprecated になった CSS2 の色名です。

色名による指定は RGB のカラースペースに属します。

## 2.8.3 色を CMYK で指定する

印刷物などへの出力に利用されるカラースペースである CMYK の値で色を指定します。

Function Name	cmk()
Argument Value	<C>, <M>, <Y>, <K>

0.0 から 1.0 または割合で指定します。

ショートハンドとして、k100 を `cmk(0, 0, 0, 100)` を指定するのと同様に扱えます。

`cmk(<C>, <M>, <Y>, <K>)` は `rgb-icc(#CMYK, <C>, <M>, <Y>, <K>)` と等価です。

## 2.8.4 HSL で指定する

RGB のほかに HSL カラースペースでの指定が可能です。

AH XSL Formatter では、Hue (色相)、Saturation (彩度)、Lightness (輝度) で色を表現する HSL カラースペースによる色指定が可能です。

Function Name	hsl()
Argument Value	<H>, <S>, <L>

`hsl()` 関数で指定します。

<H>は単位なしの数値、<S>と<L>は割合または 0.0 から 1.0 の範囲で指定します。

## 2.8.5 透過色を含む色を指定する

それぞれの色指定方法に追加して透明度が指定可能です。

Function Name	rgba()
Argument Value	[ <R>, <G>, <B>   <string> ], <A>
Function Name	hsla()
Argument Value	<H>, <S>, <L>, <A>
Function Name	cmyka()
Argument Value	<C>, <M>, <Y>, <K>, <A>

それぞれ透明度を含めない関数 `rgb()`、`hsl()`、`cmk()` への引数に加え、透明度 `<A>` を指定します。`<A>` の値は 0 から 1 の範囲です。

`rgba()` では、`rgba(green,0.4)` のように色名による指定も可能です。

PDF の出力バージョンやプロファイル仕様によっては透明が使用できないものもあります。

Option Property Name	<code>transparency-color-space</code>
Value	<code>None</code>   <code>DeviceRGB</code>   <code>DeviceCMYK</code>
Initial	<code>DeviceRGB</code>
Applies to	<code>&lt;pdf-settings&gt;</code>

PDF 出力で透過処理を行うときのカラースペースを、オプション設定 `transparency-color-space` プロパティへ指定します。

## 2.8.6 PDF 出力時の透明の変換を制御する

出力形式や PDF の種類によっては、出力時透明を含む色や画像をそのまま出力せず変換します。この動作を制御します。

Option Property Name	<code>transparency</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>&lt;pdf-settings&gt;</code>

PDF へ出力するとき、透明を含む色、画像を非透明へ変換するかどうかをオプション設定 `transparency` プロパティへ指定します。

`true` のとき透明を保持し、`false` であれば変換します。

ただし、`<fo:external-graphic>` などで埋め込む PDF に透明があるときはこれを変換できません。

## 2.8.7 PDF 出力時にカラースペースを変換する

PDF は内部にカラースペースを保持します。PDF 出力時に、FO で指定された色などを、指定したものに換します。

Option Property Name	<code>color-conversion</code>
Value	<code>none</code>   <code>black</code>   <code>gray</code>   <code>all-gray</code>   <code>all-cmyk</code>
Initial	<code>none</code>
Applies to	<code>&lt;pdf-settings&gt;</code>

RGB や CMYK で指定されたカラースペースの色を `DeviceGray` または `DeviceCMYK` に変換する方法を、オプション設定 `color-conversion` プロパティへ指定可能です。

black は RGB の黒、gray は RGB のグレー調に該当する色を DeviceGray として出力します。all-gray はすべての色を DeviceGray に、all-cmyk はすべての RGB を CMYK に変換して出力します。

この変換は、独自描画エンジンを用いてレンダリングされる画像（SVG、CGM、MahML、EMF、WMF）についても有効ですが、それ以外の画像では all-\* のときのみ変換処理されます。

出力する PDF 形式が PDF/X-1a のとき、all-gray または all-cmyk とみなされます。

### 注

埋め込まれた PDF は変換の対象外です。

## 2.8.8 画像のスケール合わせを検知する

画像のサイズが調整されて出力されることが望ましくないとき、組版時に検知し、エラー報告を行います。

主に外部画像の配置では、content-width または content-height プロパティへ scale-\*to-fit が指定されているとき、指定された領域サイズと合っていないければ画像のスケールを変更します。

Option Property Name	issue-scale-to-fit
Value	true   false
Initial	false
Applies to	<formatter-settings>

スケール変更検知の制御をオプション設定 issue-scale-to-fit プロパティへ指定します。true を指定すると、スケール変更が発生したときレベル 1 の報告を行います。

## 2.8.9 ラスタライズされるベクタ画像の解像度を指定する

Windows 版では、PDF に直接格納できない一部のベクタ画像をラスタ画像に変換して格納可能です。

Option Property Name	rasterize-resolution
Value	<integer>
Initial	108
Applies to	<pdf-settings>

オプション設定 rasterize-resolution プロパティへ、変換時のラスタ画像の解像度を 70 から 500 の数値で指定可能です。指定する数値は dpi の値ですが、<integer>なので指定するとき単位は付加しないでください。

## 2.8.10 PDF に直接格納できない画像形式の変換を制御する

PDF が対応していない画像形式は、出力時、対応している形式に AH XSL Formatter が変換して格納します。入力した画像形式の一部の変換を制御します。

PDF に格納可能な画像形式や変換設定の詳細についてはマニュアルをご覧ください。

Option Property Name	color-compression, grayscale-compression
Value	auto   zlib   jpeg   jpeg2000   keeplzw   auto2k   keeplzw2k
Initial	auto
Applies to	<pdf-settings>
Option Property Name	monochrome-compression
Value	ccitt4   ccitt3   runlength   zlib   none
Initial	ccitt4
Applies to	<pdf-settings>

PDF に直接格納できない画像形式が埋め込まれるとき、その画像は PDF と互換性のあるビットマップ形式に変換され、PDF へ格納されます。そのとき、ビットマップ形式の圧縮方法をオプション設定で制御可能です。

カラー画像については color-compression プロパティ、グレースケール画像については grayscale-compression プロパティ、白黒画像については monochrome-compression プロパティで指定します。

color-compression、grayscale-compression プロパティへ auto を指定したとき、それぞれの \*-jpeg-quality に指定した値に従いイメージデータを作成します。この JPEG と ZLIB 圧縮のサイズを比較し小さい方を選択します。

### JPEG へ変換するときの画質を指定する

直接格納できない画像形式を JPEG 形式、JPEG2000 形式へ変換するときの画質指定が可能です。元から JPEG、JPEG2000 である画像には影響しません。

Option Property Name	color-jpeg-quality, grayscale-jpeg-quality
Value	<number>
Initial	80
Applies to	<pdf-settings>

オプション設定でカラー画像であれば color-jpeg-quality、グレースケール画像では grayscale-jpeg-quality プロパティへ指定します。

JPEG2000 として格納するとき、値は圧縮率へ換算されます。

## 2.8.11 画像埋め込みのパススルーを制御する

画像埋め込み時に処理を行うかを制御します。

Option Property Name	pass-through
Value	[ all    -gif    -tiff    -png    -jpeg    -jpeg2000    -jbig2]   [ gif    tiff    png    jpeg    jpeg2000    jbig2]   none
Initial	all
Applies to	<pdf-settings>

画像の埋め込み時、そのまま埋め込めるものは埋め込むパススルーをオプション設定 `pass-through` プロパティで画像種別を指定し制御可能です。パススルーを行いたい画像種別を空白区切りで列挙するか、`all -gif` のように、除外したい画像種別に `-` を付加する指定方法が可能です。ダウンサンプリングの指定と同時に指定されているとき、ダウンサンプリングが優先されます。パススルーを指定していても、事情によりできないときはパススルーをしません。

### 注

たとえば、画像を大量に含むような組版で処理を高速化したいときに有用です。

## 2.8.12 画像、背景画像の解像度を制御する

画像、背景画像の解像度を指定します。

Property Name	axf:image-resolution
Value	normal   [ from-image    <dpi>]
Initial	from-image
Applies to	fo:external-graphic, fo:instream-foreign-object

Property Name	axf:background-image-resolution
Value	[ normal   [ from-image    <dpi>]]#
Initial	from-image
Applies to	fo:simple-page-master, fo:page-sequence, all formatting objects

出力時の画像解像度を `axf:image-resolution` プロパティへ指定します。

オプション設定 `pxpi` プロパティで指定されるシステム既定値に合わせるならば `normal`、画像が持っている解像度を使用するのであれば `from-image`、解像度を指定するのであれば `<dpi>` で指定します。

`axf:image-resolution="from-image <dpi>"`とすると、画像に解像度があればそれに従い、存在しないとき<dpi>にフォールバックします。フォールバックする<dpi>が存在しなければ normal にフォールバックします。

SVG などのベクタ画像では単位なしの数値からの変換に利用されます。

背景画像の解像度は同様に `axf:background-image-resolution` プロパティで制御可能です。背景画像が複数であるとき、値を複数指定すると、背景画像の指定と同じ順に適用されます。

## 2.8.13 画面上での画像のアンチエイリアス処理を制御する

画面上での画像表示について、アンチエイリアス処理を行うかどうかを制御します。

Property Name	<code>axf:image-smoothing</code>
Value	<code>auto</code>   <code>true</code>   <code>false</code>
Initial	<code>auto</code>
Applies to	<code>fo:external-graphic</code> , <code>fo:instream-foreign-object</code>

画像を画面上で表示するとき、アンチエイリアス処理を行うかを `axf:image-smoothing` プロパティへ指定します。

`auto` では、GUI の設定に従います。`true` なら処理を行い、`false` なら行いません。

## 2.8.14 ページマスタ、ページシーケンスへ背景を指定する

AH XSL Formatter ではページマスタやページシーケンスへ背景を指定できるように拡張されています。

ページ全体へ背景を指定したり、ページシーケンスごとに背景を切り換えることが可能です。ページマスタに `background*` プロパティがひとつでも指定してあるとき、ページシーケンスへの背景指定は無視されます。

### 注

ページマスタへ指定した背景は、`axf:bleed` プロパティによって追加した塗り足しの領域にも描画されます。

## 2.8.15 ラスタ画像のダウンサンプリングを制御する

画像のダウンサンプリングの値や処理を行う条件を制御します。

Option Property Name	color-downsampling, grayscale-downsampling, monochrome-downsampling
Value	none   average   bicubic   subsampling
Initial	none
Applies to	<pdf-settings>
Option Property Name	color-downsampling-above-dpi, grayscale-downsampling-above-dpi
Value	<number>
Initial	450
Applies to	<pdf-settings>
Option Property Name	color-downsampling-target-dpi, grayscale-downsampling-target-dpi
Value	<number>
Initial	300
Applies to	<pdf-settings>
Option Property Name	monochrome-downsampling-above-dpi
Value	<number>
Initial	1800
Applies to	<pdf-settings>
Option Property Name	monochrome-downsampling-target-dpi
Value	<number>
Initial	1200
Applies to	<pdf-settings>

ラスタ画像のダウンサンプリング方法をオプション設定\**-downsampling* プロパティへ指定可能です。

*average* はバイリニア法、*bicubic* はバイキュービック法、*subsampling* はニアレストネイバー法です。

ダウンサンプリングを行う解像度の下限はオプション設定\**-downsampling-above-dpi*、ダウンサンプリングする解像度はオプション設定の\**-downsampling-target-dpi* プロパティへ指定します。

カラー、グレイスケール画像と白黒画像 (*monochrome-\**) は初期の下限、対象解像度が大きく異なります。

## 2.8.16 SVG の既定サイズを指定する

SVG 画像を埋め込むときの表示サイズの基準を指定します。

AH XSL Formatter では、SVG 画像を埋め込むとき、SVG のルート要素に `width` と `height` が指定されていない場合は、仕様通り `width="100%"`、`height="100%"` とみなします。この動作の変更が可能です。

Option Property Name	<code>responsive-svg-size</code>
Value	<code>reference</code>   <code>viewBox</code>
Initial	<code>reference</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定 `responsive-svg-size` プロパティへ `viewBox` を指定したとき、SVG のルート要素に `viewBox` が指定されていればそのサイズとみなします。

## 2.8.17 背景にグラデーションを指定する

線形、円形グラデーションを表現する関数を指定可能です。

グラデーションを指定する関数は、`background-image` プロパティへ記述します。

Function Name	<code>linear-gradient()</code>
Argument Value	<code>[[ &lt;angle&gt;   to &lt;side-or-corner&gt;],]? &lt;color-stop&gt;#{2,}</code>
Function Name	<code>radial-gradient()</code>
Argument Value	<code>[[ &lt;shape&gt;    &lt;size&gt;]   [ at &lt;position&gt;],]?   at &lt;position&gt; ,? &lt;color-stop&gt;#{2,}</code>

Value Name	Value Composition
<code>&lt;color-stop&gt;</code>	<code>&lt;color&gt;[ &lt;percentage&gt;   &lt;length&gt;]</code>
<code>&lt;side-or-corner&gt;</code>	<code>[ left   right]    [ top   bottom]</code>
<code>&lt;position&gt;</code>	<code>[ &lt;percentage&gt;   &lt;length&gt;   left   center   right][ &lt;percentage&gt;   &lt;length&gt;   top   center   bottom]?   [[ left   center   right]    [ top   center   bottom]]   inherit</code>
<code>&lt;shape&gt;</code>	<code>circle   ellipse</code>
<code>&lt;size&gt;</code>	<code>closest-side   farthest-side   closest-corner   farthest-corner   [ &lt;length&gt;   [ &lt;length&gt;   &lt;percentage&gt;]{2}]</code>

`linear-graient()` は指定した方向からの線形なグラデーションを指定するための関数です。`<angle>` でグラデーションの向きを指定します。`<side-or-corner>` はキーワードでの指定で、たとえば `to right` であれば左から右方向、`<angle>` の `90deg` 指定に対応します。

<color-stop>は、色と長さまたは割合を指定することで、グラデーションの始点の色と、グラデーションの範囲を決定します。

radial-gradient() は中心から外側に描画されるグラデーションです。この基点の配置は <position> で変更が可能です。引数 <size> はグラデーションの終端を制御します。closest-\* は基点からもっとも近い辺または角を、farthest-\* はもっとも遠い辺または角をグラデーションの終端とします。

## 2.8.18 繰り返すグラデーションを指定する

グラデーションの繰り返しを指定します。

Function Name	repeating-linear-gradient()
Argument Value	[[ <angle>   to <side-or-corner>],]? <color-stop>#
Function Name	repeating-radial-gradient()
Argument Value	[[ <shape>    <size>]   [ at <position>],]?   at <position> ,?] <color-stop>#{2,}

repeating-\*-gradient() も基本は通常のグラデーションと同様に background-image プロパティへ指定します。加えて「指定された <color-stop> が終了するとその <color-stop> を繰り返す」という動作を行います。

<color-stop> 内で 100% のように指定すると、領域に繰り返す余地がないため、通常の \*-gradient() と変わらない動作になります。

### 注

隣り合う <color-stop> に同じ色を指定するとグラデーションにならず単色となります。これを背景画像に利用することで、「同じ割合で繰り返される線」のような修飾も可能です。

## 2.8.19 EPS 画像の処理を制御する

AH XSL Formatter は EPS 画像を処理しません。出力時、外部プロセッサに処理を渡し、変換後の画像を出力に含めることが可能です。

### 注

EPS 画像を処理可能なツールで PDF に変換しておき、PDF として埋め込むことを推奨します。

Option Property Name	EPS-processor
Value	none   distiller   ghostscript
Initial	none
Applies to	<pdf-settings>

オプション設定 EPS-processor プロパティに distiller を指定すると Adobe Distiller を、ghostscript を指定すると GhostScript を外部プロセッサとして利用します。外部プロセッサがインストールされている必要があります。

EPS-processor="distiller"は Windows 版のみ有効な指定で、acrodist.exe を利用します。パスが通っていることを確認してください。これを利用するとき、EPS 画像に次の制約があります。

- PS-Adobe-2.0 以上
- %%BeginProlog、%%EndProlog が含まれている

## Distiller のオプション引数を指定する

Option Property Name	joboptions
Value	<string>
Initial	empty string
Applies to	<pdf-settings>

オプション設定 joboptions プロパティは Distiller を利用して EPS を PDF に変換するとき、Distiller が使用する Adobe PDF Settings File のパスを指定可能です。

相対パスで指定したときは、対象の EPS ファイルの相対パスとみなされます。EPS ファイルの指定が相対パスだったとき不定になってしまうため、なるべく絶対パスで指定してください。

この指定は Windows 版でのみ有効です。

## GhostScript のパスを指定する

EPS-processor="ghostscript"のとき利用される GhostScript は Windows 版では gswin32c.exe または gswin64c.exe、それ以外では gs です。fork() などで起動するため、GPL ライセンスの問題はありません。

Option Property Name	ghostscript
Value	<string>
Initial	empty
Applies to	<pdf-settings>

オプション設定 ghostscript に GhostScript へのフルパスを記述しておくこと、GhostScript に OS 側でパスを通しておこななくても動作します。

## GhostScript のオプション引数を指定する

AH XSL Formatter では EPS を GhostScript で処理するとき、次のパラメータを指定して起動します。

```
-dPDFSETTINGS=/printer
-dNOPAUSE
-dBATCH
-dSAFER
-sDEVICE=pdfwrite
-dDEVICEWIDTHPOINTS=%width
-dDEVICEHEIGHTPOINTS=%height
-dEPSFitPage
-dCompatibilityLevel=1.3
-dAutoRotatePages=/None
-q
-sOutputFile=%temp
-c
.setpdfwrite
-f
%eps
```

%width、%height、%temp、%eps には AH XSL Formatter が値を挿入します。

Option Property Name	gs-options
Value	<string>
Initial	empty
Applies to	<pdf-settings>

GhostScript 起動時のパラメータを、オプション設定 `gs-options` プロパティへ指定することで変更可能です。

パラメータの区切り文字には LINE FEED (U+000A) を用いますが、XML の制約上直接記述ができないため、`&#xA;` のように文字参照で記述します。

### 2.8.20 ユースケース：文書全体の文字色を指定する

ルート要素に指定した色は、実質的な文書全体の既定色として利用されます。

```
<fo:root ... color="k100">...</fo:root>
```

AH XSL Formatter の既定の文字色は RGB の `black` です。FO で既定の文字色を指定したいとき、`<fo:root>` 要素で `color` を指定すると全体の文字色として反映されます。

たとえば `<fo:root color="k100">` と指定すれば、別に `color` を指定していないテキストなどは `k100` で描画されます。

## 2.8.21 ユースケース：画像を円形にくり抜いて表示する

拡張を用いて画像を円形にくり抜いて表示する方法はふたつあります。いずれも画像は背景画像として指定します。

### ボーダーを円形に指定する

```
<fo:block-container  
  width="5cm" height="5cm"  
  background-image="url(image.jpg)"  
  axf:border-radius="50%">  
<fo:block>...</fo:block>  
</fo:block-container>
```

`axf:border-radius="50%"`、つまりすべての径を等しくしたとき、背景画像のある領域は見た目上円形になります。`<fo:block-container>`へサイズを明示的に指定しています。

背景画像の縁にぼかしを行うのであれば、`axf:box-shadow` プロパティに `inset` で指定する方法があります。

### 放射グラデーションを画像に重ねる

```
<fo:block-container width="5cm" height="5cm"  
  background-image="radial-gradient(circle closest-side,  
    rgba(255,255,255,0.0), rgba(255,255,255,0.0), rgba(255,255,255,1.0)),  
  url(image.jpg)" >  
<fo:block>...</fo:block>  
</fo:block-container>
```

透過色を内側の色に指定した `radial-gradient()` を画像に重ねて指定することで画像が円形に表示されます。縁にぼかしを行うのであれば、こちらの方法が簡単です。

## 2.9 節 フロート

フロート配置を XSL 1.1 仕様よりも詳細に制御します。

フロートの配置の基本は [図 2.9-1] (p.131) のようになります。フロートを記述した箇所（アンカ）を基点に、フロートを配置できる箇所を計算し、実際に配置されます。

フロート配置の参照エリアを `axf:float-reference` プロパティへ指定可能です。

領域にフロートを配置できる余白がないとき、アンカの移動について `axf:float-move` プロパティで制御可能です。

配置できる参照領域上の位置を `axf:float-x` と `axf:float-y` プロパティで指定可能です。

そして、フロートを配置したとき、領域にテキストが回り込めるかを `axf:float-wrap` プロパティで指定可能です。

Property Name	<code>axf:float</code>
Value	<code>&lt;float-x&gt;    &lt;float-y&gt;    &lt;float-wrap&gt;    &lt;float-reference&gt;    &lt;float-move&gt;</code>
Initial	none
Applies to	<code>fo:float / floated elements</code>

`axf:float` プロパティはフロート配置関連の拡張プロパティをまとめて指定するショートハンドです。それぞれの値については、個別の項目で説明します。

マージンや回り込みの条件については `axf:float` プロパティでは指定できません。

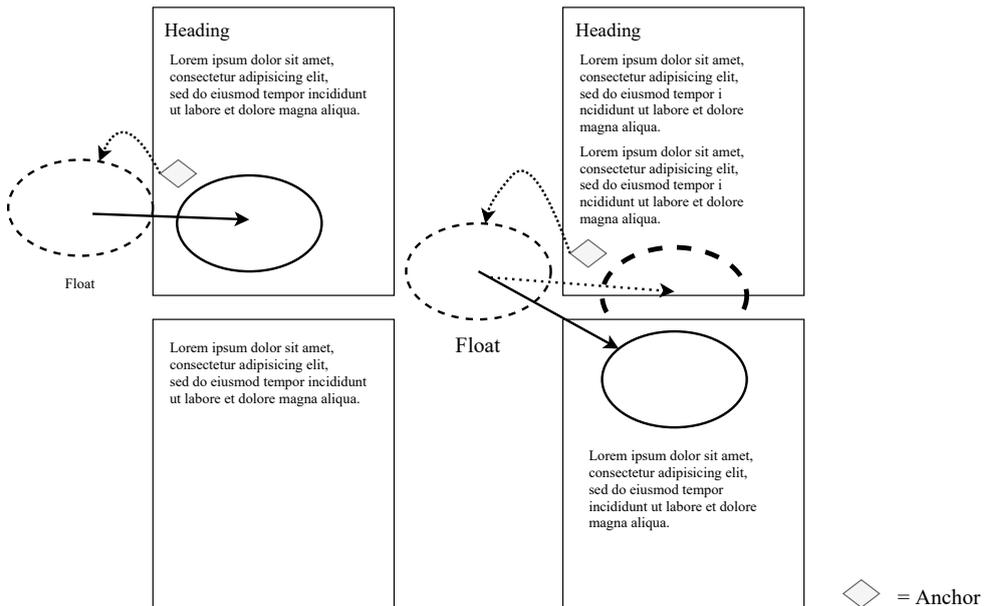


図 2.9-1 余白が十分なとき（左）と十分でないとき（右）

## 2.9.1 フロート配置の参照領域を指定する

フロート配置の参照領域を指定します。

Property Name	axf:float-reference
Value	auto   normal   page   column   multicol
Initial	auto
Applies to	fo:float

フロートの参照領域を axf:float-reference プロパティへ指定可能です [図 2.9-2] (p.132)。

axf:float-reference	意味
auto, normal	現在の参照領域。
page	ページ (本文区画) を参照領域とする。
column	段を参照領域とする。
multicol	段組領域内を参照領域とする。全段以外の段抜きを可能とする。

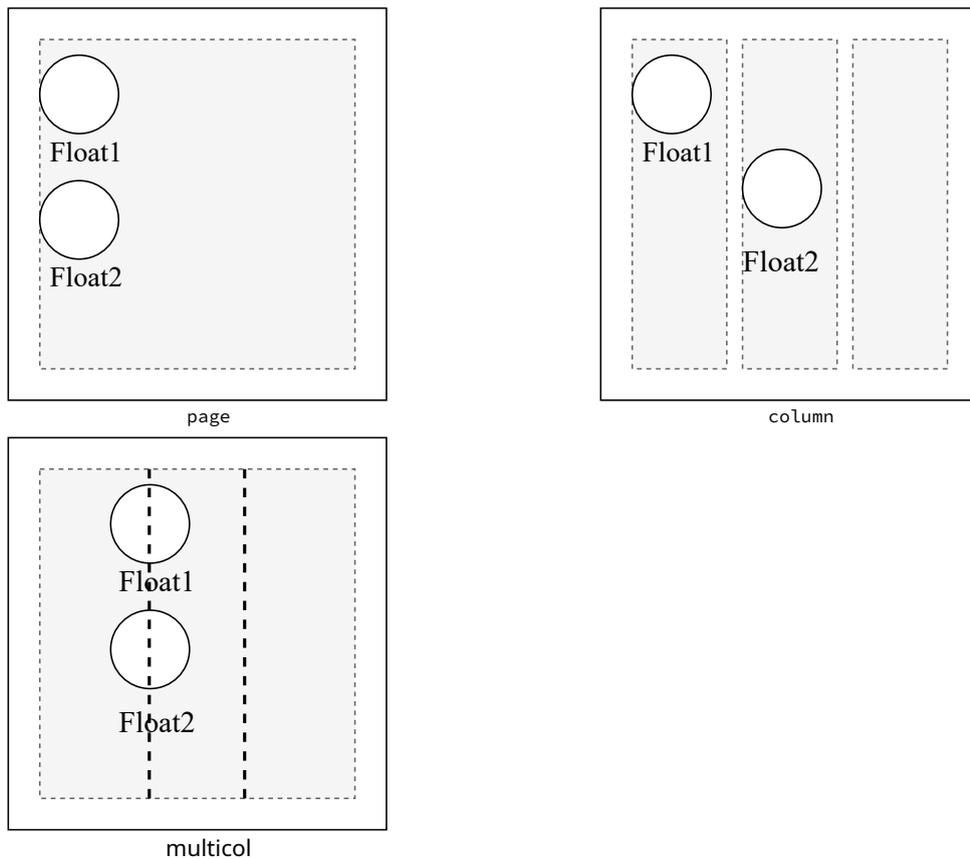


図 2.9-2 axf:float-x="start"での axf:float-reference

column は段を参照領域とするので、3 段組の領域であれば、1 段目、2 段目、3 段目という順に配置候補箇所が動きます。

multicol を指定したときもフロートは段を順に移動しますが、全段以外の段抜きも可能になります。フロートのサイズに gr を利用することで「3 段中 2 段にまたがる」のように指定可能です。auto と normal は現在の参照領域内になります。

### 注

multicol を指定したとき、参照領域のサイズとしては page を指定したときと変わらないため、axf:float-min-wrap-x プロパティなどを指定するときはページを参照領域として計算してください。

## 2.9.2 フロートとアンカを制御する

フロート配置の基点となるアンカを制御します。

フロートのマークアップを行った箇所にアンカ領域が設定され、アンカ領域を基点としたフロート参照領域の余白が十分であればフロートが配置されます。

Property Name	axf:float-move
Value	auto   next   auto-next   auto-move   keep   keep-float   <integer>
Initial	auto
Applies to	fo:float / floated elements

axf:float-move プロパティへ、余白が十分でないとき、フロート配置を次の参照領域に移動するか、アンカ領域を移動するかを指定可能です。

axf:float-move 意味	
auto	axf:float-y プロパティが none なら keep、それ以外では auto-next。
next	次の参照領域。
auto-next	現在の参照領域に余白がなければ、フロートを次の参照領域へ送る。
auto-move	現在の参照領域に余白がなければ、フロートを次の参照領域へ送る。または、フロートを移動せず、アンカとまわりのテキストを次の参照領域へ送る。
keep	余白が足りないとき、アンカの箇所から次の領域へ送る。アンカ領域に keep-with-next="always"を自動的に設定する。
keep-float	余白が足りないとき、アンカの箇所から次の領域へ送る。
<integer>	指定した数値をページ番号とみなし、そのページにフロートを配置する。総ページ数を超える数を指定したとき、ブランクページが挿入される。axf:float-y="none"のときは auto とみなされる。

## 2.9.3 フロート配置を指定する

フロートの、参照領域上の配置を指定します。

`axf:float-x` プロパティへ水平（縦組では垂直）方向、`axf:float-y` プロパティへ垂直（縦組では水平）方向のフロート配置を指定します。それぞれ組方向によって指定可能な値が異なります。

Property Name	<code>axf:float-x</code>
Value	<code>none   start   end   left   right   top   bottom   center   inside   outside   alternate   column-outside</code>
Initial	<code>none</code>
Applies to	<code>fo:float / floated elements</code>

<code>axf:float-x</code>	意味	横組	縦組
<code>none</code>	水平（縦組では垂直）方向ではフロート指定しない。	✓	✓
<code>start</code>	<code>start</code> 側にフロート配置。左から右の横組では <code>left</code> と同じ。	✓	✓
<code>end</code>	<code>end</code> 側にフロート配置。左から右の横組では <code>right</code> と同じ。	✓	✓
<code>left</code>	左側にフロート配置。	✓	-
<code>right</code>	右側にフロート配置。	✓	-
<code>top</code>	上側にフロート配置。	-	✓
<code>bottom</code>	下側にフロート配置。	-	✓
<code>center</code>	水平（縦組では垂直）方向の中央にフロート配置。	✓	✓
<code>inside</code>	ノド側にフロート配置。	✓	-
<code>outside</code>	小口側にフロート配置。	✓	-
<code>alternate</code>	段組の最初の段では <code>end</code> 、最後の段では <code>start</code> 、それ以外では <code>center</code> とみなす。	✓	✓
<code>column-outside</code>	段組の最後の段では <code>end</code> 、それ以外の段では <code>start</code> とみなす。	✓	✓

Property Name	<code>axf:float-y</code>
Value	<code>none   start   end   left   right   top   bottom   center   inside   outside</code>
Initial	<code>none</code>
Applies to	<code>fo:float / floated elements</code>

axf:float-y	意味	横組	縦組
none	垂直（縦組では水平）方向ではフロート配置しない。	✓	✓
before	before 側にフロート配置。左から右の横組では top と同じ。	✓	✓
after	after 側にフロート配置。左から右の横組では bottom と同じ。	✓	✓
left	左側にフロート配置。	-	✓
right	右側にフロート配置。	-	✓
top	上側にフロート配置。	✓	-
bottom	下側にフロート配置。	✓	-
center	垂直（縦組では水平）方向の中央にフロート配置。	✓	✓
inside	ノド側にフロート配置。	✓	✓
outside	小口側にフロート配置。	✓	✓

axf:float-x="none"かつ axf:float-y="none"のとき、フロートは無効です。記述した位置に配置されます。

## 2.9.4 テキストの回り込みを制御する

フロートへのオブジェクトの回り込みと、そのときの条件を細かく制御します。

Property Name	axf:float-wrap
Value	auto   wrap   skip
Initial	auto
Applies to	fo:float / floated elements

axf:float-wrap プロパティへ、テキストの回り込みをするかどうかを指定します。

skip の指定ではテキストは回り込みをせず、wrap の指定では回り込みをします。ただし、段組でフロートが中央あたりにあり、両側にテキストが回り込み可能な幅があるとき、wrap を指定していても skip とみなされテキスト配置はフロートをとばします。

auto では、float-x が none の以外のとき wrap、none のときは skip とします。

## アキによって回り込みや揃えを変更する

Property Name	axf:float-min-wrap-x,axf:float-min-wrap-y
Value	normal   <length>   <percentage>
Initial	normal
Applies to	fo:float / floated elements
Percentages	refer to the size of containing block

Property Name	axf:float-centering-x,axf:float-centering-y
Value	none   auto   <length>   <percentage>
Initial	none
Applies to	fo:float / floated elements
Percentages	refer to the size of containing block

図表が大きく、テキストが1行あたり1文字や2文字しか回り込めず内容が読みづらくなってしまうときがあるでしょう。

フロートにテキストが回りこむための幅の最小値を axf:float-min-wrap-x と axf:float-min-wrap-y プロパティへ指定可能です。normal は 0pt の指定と同じです。

テキストが回り込めない幅や高さだったときのみフロートの配置を中央にするには axf:float-centering-x と axf:float-centering-y プロパティへ指定します。

<length>や<percentage>を指定したとき、テキストが回りこむための幅が指定した値未満なら中央寄せにします。auto は axf:float-min-wrap-\*の対応する幅に足りないとき、フロートを中央に配置します。

## 2.9.5 フロートのマージンを指定する

フロートとテキスト間のマージンや、フロート間のマージンを指定します。

Property Name	axf:float-margin-x, axf:float-margin-y
Value	[ <length>   <percentage>][ <length>   <percentage>]?
Initial	0pt
Applies to	fo:float / floated elements
Percentages	refer to the size of containing block

フロートと回り込むテキスト間のマージンは axf:float-margin-x、axf:float-margin-y プロパティで指定可能です。

axf:float-margin-x プロパティはインライン進行方向、値をふたつ指定するとき最初の値が start、次が end 側のマージンとなります。

axf:float-margin-y プロパティはブロック進行方向、値をふたつ指定すると最初の値が before、次が after 側のマージンとなります。

## フロートとフロートの間のマージンを指定する

フロートとそのほかのオブジェクトとのマージンとは別に、フロート同士が隣接するときのマージンを指定します [図 2.9-3] (p.137)。

Property Name	axf:float-float-margin-x, axf:float-float-margin-y
Value	auto   [[ <length>   <percentage>][ <length>   <percentage>]?)
Initial	auto
Applies to	fo:float / floated elements
Percentages	refer to the size of containing block

axf:float-float-margin-x、axf:float-float-margin-y プロパティへ、それぞれ X 方向、Y 方向で隣接するフロートとのマージンを指定可能です。それぞれ最大値は axf:float-margin-x、axf:float-margin-y プロパティの値となります。

auto のとき、axf:float-margin-x、axf:float-margin-y プロパティの値が利用されます。

[図 2.9-3] (p.137) では B に指定した axf:float-float-margin-y プロパティの値が有効となっています。隣接したフロートのどちらにもフロート間マージンが指定されているとき、一方の値のみが有効です。

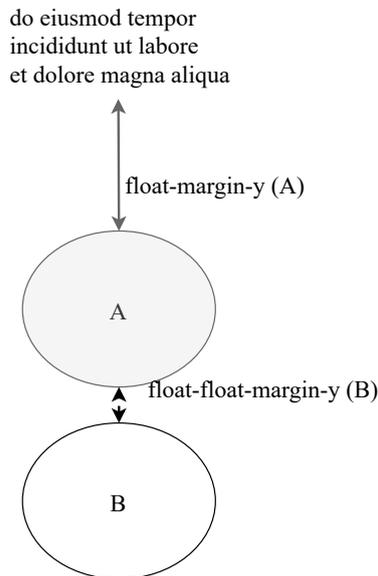


図 2.9-3 float-margin と float-float-margin

## 2.9.6 フロートのオフセット位置を変更する

フロートの配置された位置からのオフセットを指定します。

Property Name	axf:float-offset-x, axf:float-offset-y
Value	<length>   <percentage>
Initial	0pt
Applies to	fo:float / floated elements
Percentages	refer to the size of containing block minus the size of float

`axf:float-offset-x` プロパティへ、フロートの X 方向配置からのオフセットを指定します。配置が `start` なら `end` 側へのオフセット、配置が `end` なら `start` 側へのオフセットとなります。

`axf:float-offset-y` プロパティへ、フロートの Y 方向配置からのオフセットを指定します。配置が `before` なら `after` 側へのオフセット、配置が `after` なら `before` 側へのオフセットとなります。

## 2.10 節 注

注の配置方法や注番号の振り方の制御が可能です。

### 2.10.1 合印、脚注番号を自動化する

ページ、段分割といったタイミングでの脚注番号のカウン制御が可能です。

Element Name	<axf:footnote-number>
Property	id
Parent	descendants of fo:footnote's child fo:inline
Element Name	<axf:footnote-number-citation>
Property	ref-id
Parent	decsendants of fo:footnote-body

<axf:footnote-number>要素で脚注番号を生成、表示します。

<axf:footnote-number-citation>要素で脚注番号を引用します。<axf:footnote-number>で指定した id の値を、<axf:footnote-number-citation>の ref-id プロパティへ指定します。ref-id プロパティには、同じ<fo:footnote>の子孫として記述された<axf:footnote-number>要素の id の値のみ指定可能です。

Property Name	axf:footnote-number-initial
Value	auto   <number>
Initial	auto
Applies to	fo:page-sequence
Property Name	axf:footnote-number-reset
Value	auto   none   page   odd-page   even-page   column
Initial	auto
Applies to	fo:page-sequence

axf:footnote-number-initial プロパティで脚注番号の初期値の数字を、axf:footnote-number-reset プロパティへ初期化のタイミングを指定します。

axf:footnote-number-initial="auto"のとき、前のページシーケンスの脚注番号を引き継ぎます。最初のページシーケンスでは 1 とみなされます。

axf:footnote-number-reset	意味
auto	前のページシーケンスの指定を引き継ぐ。最初のページシーケンスのとき none。
none	リセットしない。
page	改ページでリセット。
odd-page	奇数ページへの改ページでリセット。
even-page	偶数ページへの改ページでリセット。
column	改段でリセット。

脚注番号が axf:footnote-number-reset プロパティへ指定された条件により初期化されたとき、axf:footnote-number-initial プロパティへ指定した値となります。

横組で見開きごとに脚注番号をリセットするのであれば even-page、縦組の見開きごとであれば odd-page となります。

「章単位で脚注番号をリセットしたい」というときは、たとえば axf:footnote-number-initial="1" を指定したページシーケンスを章ごとに切り替えることで実現が可能です。

## 注

脚注領域自体は傍注と脚注などを混在させられますが、axf:footnote-number-\* プロパティは <fo:page-sequence> で指定するため、<axf:footnote-number> による採番はページシーケンスで共通となります。

## 注番号の書式を指定する

Property Name	axf:footnote-number-format
Value	<string>
Initial	1
Applies to	fo:page-sequence

axf:footnote-number-format プロパティへ脚注番号の書式を指定可能です。指定可能な書式については format プロパティと同様です。

## 2.10.2 注の配置を制御する

傍注や段ごとの脚注など、注の配置を指定します。

XSL 1.1 仕様では <fo:footnote> は全段の脚注として扱われます。AH XSL Formatter では脚注配置の細かな制御が可能です。

Property Name	axf:footnote-position
Value	page   odd-page   even-page   start   end   inside   outside   column   start-column   end-column   inside-column   outside-column
Initial	page
Applies to	fo:region-body, fo:footnote

<fo:footnote>による注がページ上のどこに配置されるかを、axf:footnote-position プロパティへ指定します [図 2.10-1] (p.142)。

odd-page、even-page は<fo:region-body>へのみ指定可能です。

## 横組で脚注配置を指定する

axf:footnote-position	横組での意味
page	各ページの本文区画内の after 側に配置。XSL 1.1 仕様仕様の標準的な脚注。
odd-page	奇数ページの本文区画内の after 側に配置。
even-page	偶数ページの本文区画内の after 側に配置。

## 横組で傍注配置を指定する

横組で region-start または region-end に脚注の配置が可能です。横組におけるこの注を、傍注と呼びます [図 2.10-2] (p.143)。

領域に収まらないときは次の領域には送られず、はみ出しが生じます。

axf:footnote-position	横組での意味
start	各ページの region-start 内に配置。
end	各ページの region-end 内に配置。
inside	偶数ページの region-end 内、奇数ページの region-start 内に配置。
outside	偶数ページの region-start 内、奇数ページの region-end 内に配置。

inside はノド寄りの傍注、outside は小口寄りの傍注です。標準的な傍注のときは outside を指定するとよいでしょう。

## 段組での脚注配置を指定する

段ごと、最初の段、最後の段、ページの内側、外側を指定可能です [図 2.10-3] (p.143)。

\*-column は<fo:region-body>へのみ指定可能です。また、これらが指定してあるとき、<fo:footnote>の axf:footnote-position プロパティには column を指定できません。

start、end、inside、outside による傍注の指定は段組でないときと同様です。また、段組でないときの\*-column の指定は、それぞれ対応する傍注として扱われます。

afx:footnote-position	段組での意味
page	段抜きの脚注（初期値）
column	段ごとの脚注
start-column	start 側の段に脚注
end-column	end 側の段に脚注
inside-column	ノド側の段に脚注
outside-column	小口側の段に脚注

## 縦組での注配置を指定する

writing-mode="tb-rl"のときの注の配置を指定します [図 2.10-4] (p.144)。

afx:footnote-position	縦組での意味
page	各ページの本文区画内の after 側に配置。
odd-page	奇数ページの本文区画内の after 側に配置。
even-page	偶数ページの本文区画内の after 側に配置。
start	各ページの region-start 内に配置。縦組での頭注。
end	各ページの region-end 内に配置。縦組での脚注。

AH XSL Formatter ではページに writing-mode="tb-rl"が指定されているとき、注配置の初期値は afx:footnote-position="odd-page"、つまり奇数ページの小口になります。

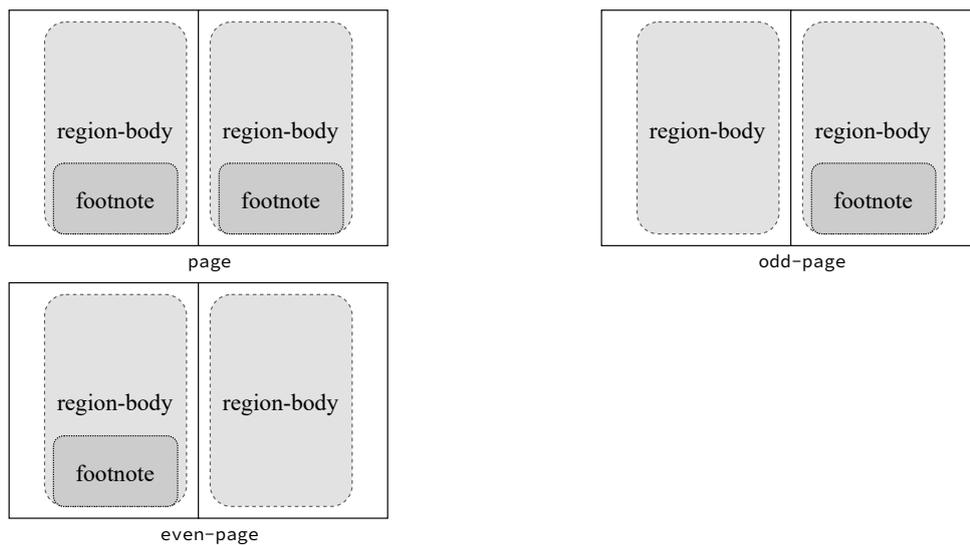


図 2.10-1 横組での注の配置

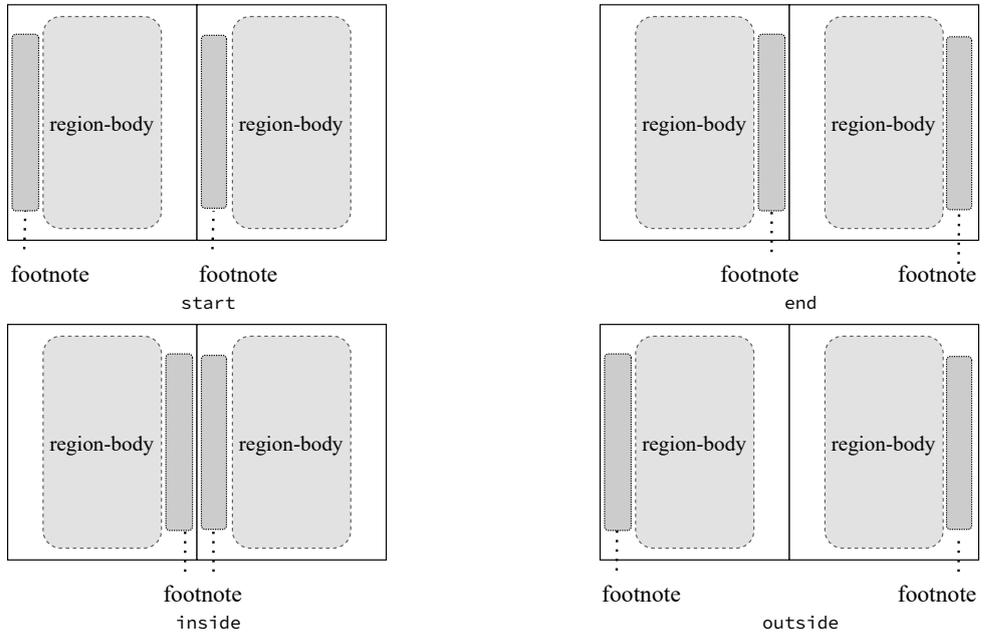


図 2.10-2 横組での傍注の配置

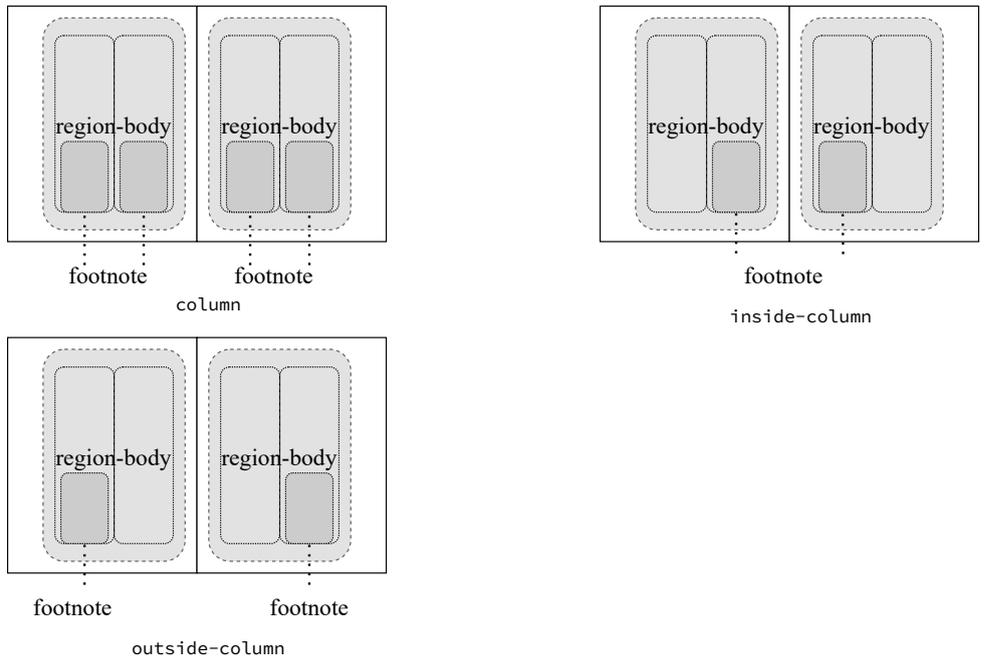


図 2.10-3 段組での注の配置

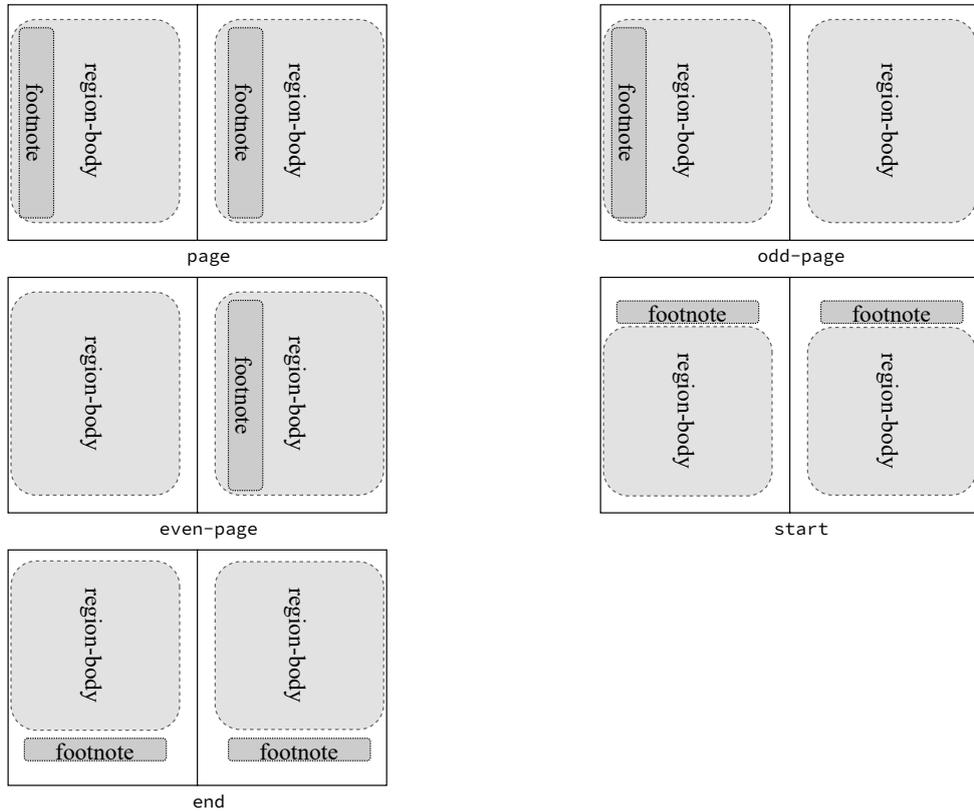


図 2.10-4 縦組での注の配置

### 2.10.3 注の揃えを制御する

注の揃えを指定します。

Property Name	axf:footnote-align
Value	auto   before   after
Initial	auto
Applies to	fo:region-body, fo:footnote

揃えの対象が脚注（または頭注）か傍注かで動作が異なります。

axf:footnote-align	注の位置	意味
auto	傍注	1 段組ではアンカ位置、2 段組以上では before 側へ揃える。
auto	脚注	通常の配置。
before	傍注	常に before 側に揃える。
before	脚注	ページ内文字列の直後に配置。
after	傍注	after 側に揃える。
after	脚注	ページの最後。通常の配置と変化なし。

## 2.10.4 同じ注の表示を抑制する

同じ内容の注が同じページに複数回表示されることを抑制します。

Property Name	axf:suppress-duplicate-footnote
Value	true   false
Initial	false
Applies to	fo:footnote
Inherited	yes

axf:duplicate-footnote プロパティに true を指定すると、同じ注を同じページ内で割り当てるとき、同じ注の内容が表示されることを抑制します。

axf:footnote-position="column"のときは、同じ段内の注についてのみ抑制されます。

「同じ注」かどうかについては、<fo:footnote-body>の内容から判断されます。

## 2.10.5 注の並べ方を制御する

複数の脚注を同じ行に表示します。

XSL 1.1 仕様の標準的な脚注は、それぞれ別のブロックとして扱われるため、注ごとに別の行となります。AH XSL Formatter ではこの動作を変更し、インラインとして扱うことが可能です。

Property Name	axf:footnote-stacking
Value	block   inline
Initial	block
Applies to	fo:region-body, fo:footnote

axf:footnote-stacking プロパティへ inline を指定したとき、注を同じ行に並べ表示します。

## 2.10.6 注とアンカの分離を抑制する

注はそのアンカよりも後ろのページ、段に送られることがあります。注のアンカごと後ろに送るように指定します。

通常、注がその注領域内に入らなくなったとき、注は次の注領域に送られますが、アンカはページ、段を移動しません。このとき、合印とその注が別のページに表示されることがあります。

Property Name	axf:footnote-keep
Value	auto   always   none
Initial	auto
Applies to	fo:region-body, fo:footnote

`axf:footnote-keep` プロパティへ `always` を指定したとき、そのアンカの行以降を次のページに送り、脚注とアンカを同一ページ内に配置します。段組かつ `axf:footnote-position="column"` のとき、脚注とアンカを同じ段内に配置します。

`none` を指定すると、脚注が段内に入りきれないとき、そのアンカの行以降を次のページへ送り、脚注は元のページへ収めようとしています。

## 注

脚注の長さやアンカの位置によっては、本文が3行程度の位置でも改ページが起こることもあります。

## 2.10.7 長い脚注を複数ページ、段に分割する

ページ、段で脚注の分割と、脚注が占める領域の高さの最大値を指定します。

Property Name	<code>axf:footnote-max-height</code>
Value	<code>auto</code>   <code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>
Initial	<code>auto</code>
Applies to	<code>fo:region-body</code>
Percentages	refer to the height of the page
Option Property Name	<code>auto-break-footnote</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

脚注の高さの最大値を、`axf:footnote-max-height` プロパティへ指定します。指定した高さを超えるとき、脚注は次のページへ送られます。

本文がないページ、段では `axf:footnote-max-height` プロパティへの指定は無効です。`axf:footnote-max-height="auto"` のとき、脚注分割はオプション設定 `auto-break-footnote` の指定によって変化します。

<code>axf:footnote-max-height</code> と <code>auto-break-footnote</code>	結果
<code>axf:footnote-max-height="auto"</code> かつ <code>auto-break-footnote="true"</code>	脚注の高さを制限せず、脚注のページ、段分割をする。
<code>axf:footnote-max-height="auto"</code> かつ <code>auto-break-footnote="false"</code>	脚注の高さを制限せず、脚注のページ、段分割をしない。脚注のはみ出しが発生することがある。
<code>axf:footnote-max-height</code> に正の長さか割合を指定	<code>auto-break-footnote</code> の指定は影響しない。脚注の高さを制限する。

## 2.10.8 ユースケース：後注をページ下部に配置する

後注の配置をページ下部に張り付かせるため、フロートにします。

後注自体は特別な拡張をせずとも記述可能ですが、`axf:float-y="bottom"`と組み合わせることで配置をページの下部に限定しました [図 2.10-5] (p.148)。

### 後注をページ下部に配置する

```
<!-- 後注をフロートにまとめる -->
<fo:float axf:float-y="bottom" axf:float-x="none" ...>
  ...
  <fo:block ...>
    <fo:inline font-size="70%"
      baseline-shift="super">(1)</fo:inline>本白書においては、...
  </fo:block>
  ...
</fo:float>
```

### 本文と後注間に罫線を引く

```
<!-- 後注の前に罫線のブロックを記述する -->
<fo:float ...>
  <fo:block keep-with-next="always">
    <fo:leader leader-pattern="rule"
      rule-style="solid" rule-thickness="0.7pt" leader-length="50mm" />
  </fo:block>
  ...
</fo:float>
```

後注の開始位置に罫線を引くため、後注のブロックに `border-top` プロパティなどを指定するか、`<fo:leader>`要素を後注の項目の上に配置します。ボーダーによる罫線は波線や二重線の調整が行えます。`<fo:leader>`では、罫線の長さの調整が容易です。

## はじめに

新型コロナウイルス感染症（COVID-19）は、令和元年12月に中華人民共和国湖北省武漢市での発生が報告され、当初は中華人民共和国を中心に流行を見せていたが、その後世界各地において感染が見られるようになり、令和2年1月30日には、世界保健機関（WHO）が「国際的に懸念される公衆衛生上の緊急事態（PHEIC）」を宣言し、わずか数か月ほどの間に「パンデミック」といわれる世界的な流行となった<sup>(1)</sup>。我が国においては、同年1月15日に最初の感染者が確認された後、4月末までに、約14,000人の感染者、約400人の死亡者が確認されており、4月7日には、新型インフルエンザ等対策特別措置法（平成24年法律第31号。以下「特措法」という。）第32条第1項に基づく緊急事態宣言が行われた。一方で、海外の状況としては、新型コロナウイルス感染症が発生している国は、南極大陸を除く全ての大陸に存在する状況となっており、爆発的な患者の急増のため、強制的な外出禁止の措置や生活必需品以外の店舗閉鎖などを行う、いわゆる「ロックダウン」と呼ばれる強硬な措置を取らざるを得ない国も見られている。わずかな間に世界中に拡大した今回の新型コロナウイルス感染症の流行は、人々の活動が、国境を越え、グローバルにつながっているという現実を明らかにするとともに、この流行の克服のために、早急な診断法や治療法、ワクチンの開発等が求められている現在の状況は、我が国が直面している世界規模の課題解決するためには、科学技術の英知の結果が不可欠であるということを改めて示すこととなった。

我が国では、科学技術基本法（平成7年法律第130号）に基づき、政府が5年ごとに科学技術基本計画を策定し、科学技術政策を総合的かつ計画的に推進している。第5期科学技術基本計画（平成28年1月閣議決定。計画期間は平成28年度から令和2年度まで）においては、ネットワーク化やサイバー空間利用の飛躍的發展といった潮流を踏まえた目指すべき未来社会の姿として、サイバー空間とフィジカル空間（現実空間）の融合により経済・社会的課題を解決し、人々が質の高い生活を送ることのできる人間中心の社会である「Society 5.0」<sup>(2)</sup>を提唱し、世界に先駆けて実現していくこととした。現在、人工知能（AI<sup>(3)</sup>）、量子技術、ゲノム編集を始めとした最先端の新興技術が飛躍的に進展し、科学技術がかつてないスケールで、社会に大きな影響をもたらす状況となるとともに、世界の不確実性はますます高まっている。また、多様性の尊重や包摂性の重視等の様々なニーズへのきめ細やかな対応や、新型コロナウイルス感染症を始めとする新興・再興感染症、地球温暖化等の地球規模課題への対応等、多様化・複雑化する国内外の諸課題の解決に向け

て科学技術への期待は大きい。現在、政府においては次期科学技術基本計画策定に向けた議論が行われているが、「Society 5.0」を提唱した我が国が、最先端の科学技術により社会課題を解決し、世界の持続的発展に貢献するとともに、先端科学技術と社会が調和した、「誰一人取り残さない」社会（インクルーシブ社会）を実現することが急務となっている。

※「令和2年版科学技術白書（文部科学省）」より一部抜粋

<sup>(1)</sup> 本白書においては、おおむね令和2年4月末までに把握した新型コロナウイルス感染症の状況及び対応について述べている。

<sup>(2)</sup> 第5期科学技術基本計画では「必要なもの・サービスを、必要な人に、必要な時に、必要なだけ提供し、社会の様々なニーズにきめ細やかに対応でき、あらゆる人が質の高いサービスが受けられ、年齢、性別、地域、言語といった様々な違いを乗り越え、生き生きと快適に暮らすことのできる社会」とし、「科学技術イノベーション総合戦略2017」（2017年6月閣議決定）では「サイバー空間とフィジカル空間を高度に融合させることにより、地域、年齢、性別、言語等による格差なく、多様なニーズ、潜在的なニーズにきめ細やかに対応したモノやサービスを提供することで経済的發展と社会課題の解決を両立し、人々が快適で活力に満ちた質の高い生活を送ることのできる、人間中心の社会」としている。

<sup>(3)</sup> Artificial Intelligence

図 2.10-5 後注をページ下部に配置する

## 2.11 節 マーカ、相互参照

柱に表示する内容を取得するマーカや相互参照を制御します。

### 2.11.1 マーカ参照の重複を除去する

連続するマーカ参照の内容が重複するとき、一方の表示を抑制します。

たとえば柱でマーカ参照を節タイトルの表示に使用するとき、節タイトルのマーカはページ内にいくつあるか分かりません。同じページに複数あれば「最初の節タイトルから最後の節タイトル」、ひとつの節であれば「節タイトル」を表示したいとします [図 2.11-1] (p.150)。

```
<!-- 「最初の節タイトルから最後の節タイトル」をヘッダに表示する -->
<fo:static-content
  flow-name="xsl-region-before">
  <fo:block ...>
    <fo:retrieve-marker
      retrieve-class-name="heading"
      retrieve-position="first-including-carryover"/>
    から
    <fo:retrieve-marker
      retrieve-class-name="heading"
      retrieve-position="last-starting-within-page" />
  </fo:block>
</fo:static-content>
```

このように、インラインで連続するマーカ参照が、同じ内容のマーカを参照することがあります。上の FO では、同じページの<fo:marker marker-class-name="heading">が現われなければ、「節タイトル 1 から節タイトル 1」のように表示されます。

Property Name	axf:suppress-duplicate-marker-contents
Value	true   false
Initial	false
Applies to	all formatting objects
Inherited	yes

axf:suppress-duplicate-marker-contents プロパティは XSL 2.0 仕様における「coalescing markers」の働きを実現します。true を指定すると、マーカ参照が重複するとき、重複するマーカ参照とその間の要素を除去します。

ページシーケンスで axf:suppress-duplicate-marker-contents="true" を指定します。

```
<!マーカ参照内容の重複表示を抑制する>
<fo:page-sequence ...
  axf:suppress-duplicate-marker-contents="true">
```

```
<fo:static-content flow-name="xsl-region-before">...
  </fo:static-content>
  ...
</fo:page-sequence>
```

適用結果の例を示します [図 2.11-2] (p.150)。

マーカ参照の抑制が機能するためには、次の条件があります。

- 参照するマーカが同じ (retrieve-marker-class の値が共通)
- <fo:retrieve-marker>、または<fo:retrieve-table-marker>がインラインで兄弟の位置
- <fo:marker>がインライン要素

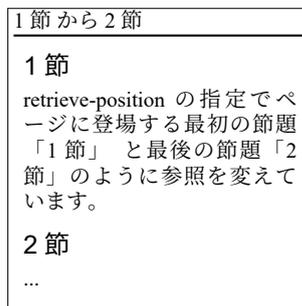


図 2.11-1 「最初の節タイトルから最後の節タイトル」をヘッダに表示する

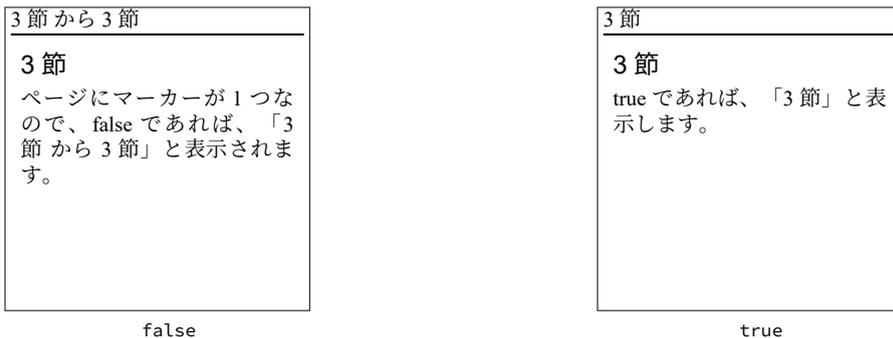


図 2.11-2 axf:suppress-duplicate-marker-contents

## 2.11.2 ページ番号の桁数を仮組で配置する

ページ番号参照が自身より後ろを参照するとき、確定していないページの桁数を仮の値で指定します。

参照するページ番号がそのページよりも後ろであるとき、AH XSL Formatter はその場では仮の桁数分の文字を領域として確保して組版を行い、後でページ番号が確定してから内容を調整します。少なくとも 3 桁を見込むので、1 桁ページの冊子であったり、5 桁ページのマニュアルの参照を行うときに影響が大きくなります。たとえば 10 ページ目で 2000 ページ目を参照するような文がある場合、3 桁を見込んでいると数字ひとつ分の領域のズレを調整することになります。

Property Name	axf:assumed-page-number
Value	<number>
Initial	N/A
Applies to	all formatting objects
Inherited	yes

axf:assumed-page-number プロパティに、仮組に用いるページ番号の指定が可能です。

### 2.11.3 ページ番号を逆順に振る

ページ番号表示をそのページシーケンスの逆側から数えた値にします。

異なるページ進行方向のページシーケンスが混在する FO では、ページ番号はそれぞれの方向で進んでいきます。

Property Name	axf:reverse-page-number
Value	true   false
Initial	false
Applies to	fo:page-number, fo:page-number-citation, fo:page-number-citation-last

axf:reverse-page-number プロパティへ true を指定すると、ページ番号を、そのページシーケンスの最後のページから数えて逆順に振ります。

### 2.11.4 ページ番号に段番号を表示する

段組のとき、ページ番号表示を行うときに段番号を表示します。

Property Name	axf:number-type
Value	page   column   page-and-column
Initial	page
Applies to	fo:page-number, fo:page-number-citation, fo:page-number-citation-last

ページ番号またはその参照を記述するとき、axf:number-type プロパティに column を指定すると、ページ番号ではなく段番号を表示します。

page-and-column を指定すると、ページ番号と段番号をあわせて表示します。

Property Name	axf:column-number-format
Value	<string>
Initial	A
Applies to	fo:page-sequence

段番号の書式は `axf:column-number-format` プロパティで指定可能です。 `axf:number-type="page-and-column"` のとき、ページ番号の書式は `format` で別に指定します。

## 2.11.5 ページ番号表示とは別に物理的なページ数を得る

通常、ページ番号の表示はページシーケンスでの制御指定を反映します。たとえば `initial-page-number="100"` であれば、100 ページからカウントされます。AH XSL Formatter では、複数のページシーケンスで番号をリセットしているようなときでも物理的なページ番号を得られます。

Property Name	<code>axf:physical-page-number</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>fo:page-number</code> , <code>fo:page-number-citation</code> , <code>fo:page-number-citation-last</code>

`axf:physical-page-number` プロパティへ `true` を指定することで、物理的なページ番号の表示、参照が可能です。

印刷用にページの外に通しノンプルを置くのであれば、たとえばページマスタに `axf:crop-area-visibility="visible"` を指定した上で、周辺区画にブロックコンテナの絶対配置で `axf:physical-page-number="true"` を指定したページ番号を記述することで実現します。

総ページ数を得るのであれば、記述する最後のオブジェクトに `id` を振り、その `id` を参照するページ番号参照に `axf:physical-page-number="true"` を指定することで、総ページ数となります。

```
<!-- 通し番号 / 総ページ数 -->
<fo:block>
  <fo:inline>
    <fo:page-number axf:physical-page-number="true" /> /
    <fo:page-number-citation ref-id="lastPage"
      axf:physical-page-number="true"/>
  </fo:inline>
  ...
</fo:block>
<!-- 最終ページ -->
<fo:block id="lastPage">...</fo:block>
```

## 2.11.6 相対ページ番号を得る

「`n` ページ後」のような相対的なページ番号を得ます。

`<fo:page-number*>` 要素は基本的にページシーケンスを基準としたページ番号を生成し参照します。AH XSL Formatter ではページ番号の基点を指定することで、相対的なページ番号を取得可能です。

Property Name	axf:origin-id
Value	<idref>
Initial	none
Applies to	fo:page-number, fo:page-number-citation, fo:page-number-citation-last

axf:origin-id プロパティに<idref>を指定し、その id を持つ要素をページの基点としたページ番号を生成、参照が可能です。生成されるページ番号は「ref-id のページ - origin-id のページ + 1」ですが、ref-id のページが origin-id のページより後ろのとき 0 になります。<fo:page-number>では ref-id の代わりに自身の位置とみなされます。

<fo:page-number-citation>要素は通常、<refid>を指定することで「200 ページで補足します。」のようにページ番号を表示しますが、axf:origin-id プロパティによって相対ページ番号を表現することも可能となります。

「n ページ後」を表現してみましょう。

```
<!-- 通常のページ参照 -->
<fo:block id="here">
  <fo:page-number-citation ref-id="referred-after" />ページで補足します。
  ...
<!-- 相対ページ参照-->
<fo:page-number-citation ref-id="referred"
  axf:origin-id="here" />ページ後で補足します。
</fo:block>
<!-- 参照先-->
<fo:block id="referred-after">...</fo:block>
```

id="referred-after"のページが 220 ページ、id="here"のページが 200 ページだとすると、上の FO の相対ページ参照は処理時に「220 - 200 + 1」となり、「21 ページ後で補足します。」となります。

### 注

ref-id のページより origin-id のページ番号が大きいときには 0 となります。「n ページ前で .....」とするときには axf:origin-id プロパティではない手段を考える必要があります。

## 2.11.7 ページ番号の前置、後置文字列を抑制する

ページシーケンスで指定したページ番号と同時に表示するテキストを、個別に抑制します。

XSL 1.1 仕様ではページ番号やページ番号参照を表示するとき、ページ番号に前置、後置して表示したいテキストをそれぞれ fo:folio-prefix、fo:folio-suffix で指定可能です。これらはページシーケンス単位で付加したい情報をページ番号表示に追加することになります。

「ほかのページシーケンスから参照するときの区別などに利用するが、同じページシーケンス内から参照するときには表示を抑制したい」といった要望があるとき、次の拡張が有用です。

Property Name	axf:suppress-folio-prefix, axf:suppress-folio-suffix
Value	true   false
Initial	false
Applies to	fo:page-number, fo:page-number-citation, fo:page-number-citation-last

ページ番号、ページ番号参照で表示を抑制したいときに `axf:suppress-folio-*=true` を指定します。

適用対象は `<fo:page-number-*>` 要素であるので、「基本的にページ番号に前置・後置文字列が表示されてほしいが、表示を省略したいときに `axf:suppress-folio-*=true` を指定する」などの用途になります。

## 2.11.8 索引で連続するページ参照の表示を制御する

索引で参照するページ番号の数や桁数が多いとき、略記方法を指定します。

索引での連続するページ番号参照についての制御を指定する `merge-sequential-page-numbers` プロパティが拡張されています。

XSL 1.1 仕様にある `merge` を指定すると、索引などでページ番号参照の結果が「2,3,4,5,6,7」と並ぶようなとき、「2-7」のように複数ページをまとめた表記が可能です。このとき2ページのみときも「2-3」のようにまとめられますが、ページ数の桁が増えると「1002-1003」のように字数が増えます。

Property Name	merge-sequential-page-numbers
Value	merge   merge-f   merge-ff   leave-separate
Initial	merge
Applies to	fo:page-index-citation-list
Inherited	yes
Element Name	<code>&lt;axf:index-page-citation-range-f-suffix&gt;</code>
Parent	fo:index-page-citation-list
Element Name	<code>&lt;axf:index-page-citation-range-ff-suffix&gt;</code>
Parent	fo:index-page-citation-list

`merge-sequential-page-numbers="merge-f"` では連続する2ページするとき「2f.」のように表示します。3ページ以上では `merge` と同じ動作をします。

`merge-sequential-page-numbers="merge-ff"` では連続する3ページを「2ff.」のように表示します。2ページでは `merge-f`、4ページ以上では `merge` と同じです。

接尾辞「f.」「ff.」はそれぞれ `<axf:index-page-citation-range-f-suffix>`、`<axf:index-page-citation-range-ff-suffix>` の子として、別のテキストを指定可能です。

merge-sequential-page-numbers	索引で並ぶページ番号	適用結果
leave-separate	1000,1001,1002	1000,1001,1002
merge	1000,1001,1002	1000-1002
<b>merge-f</b>	1000,1001	1000f.
<b>merge-ff</b>	1000,1001,1002	1000ff.

## 2.12 節 段落の始め方、終わり方

段落の開始、終了についての拡張仕様を説明します。

### 2.12.1 イニシャルキャップを指定する

行頭の文字サイズを大きくして段落の先頭であることを強調する、イニシャルキャップが行えます。

段落先頭の文字のフォントサイズを大きくするだけであれば、`<fo:character>`へ大きめのフォントサイズを指定するなどで実現が可能です。しかし段落に文字を沈みこませるには、それぞれの行の高さや続くテキストとの位置関係などをさまざまに計算し、指定しなければなりません。

`axf:initial-letters` プロパティにより、[CSS3-Inline] Initial Letters にあるようなイニシャルキャップの指定が可能です [図 2.12-1] (p.158)。

Property Name	<code>axf:initial-letters</code>
Value	<code>normal</code>   [[ <code>&lt;number&gt;</code>   <code>&lt;length&gt;</code> ] <code>&lt;integer&gt;?</code>   [ <code>&lt;number&gt;</code>   <code>&lt;length&gt;</code> ]]&&[ <code>drop</code>   <code>raise</code> ]] [ <code>adjacent</code>   <code>&lt;integer2&gt;</code> ]?
Initial	<code>normal</code>
Applies to	<code>fo:block</code>

`axf:initial-letters` プロパティによってイニシャルキャップの基本的な制御を指定します。

まず文字サイズを行数または寸法で指定します。寸法での指定では、何行分に相当するのか換算されます。3 を指定すると 1 行目の `cap-height` から 3 行目のベースラインがイニシャルキャップの文字サイズとなります。単位 `dcm` はこの値を基にするため、ここでは指定しないでください。

次にドロップキャップを何行分沈みこませるかを行数で指定します。または `drop` か `raise` を指定します。`drop` ではすべて沈みこむ通常のドロップキャップ、`raise` では 1 行目のベースラインから突き出したレイズドキャップとなります。

ドロップキャップが配置される行より後の行のインデントを `adjacent` または `<integer2>` で指定します。

指定した行数よりも実際の行数が少ないとき、イニシャルキャップを指定したブロックに続くブロックもインデントして配置されます。

Property Name	<code>axf:initial-letters-color</code>
Value	<code>&lt;color&gt;</code>
Initial	the value of the 'color' property
Applies to	<code>fo:block</code>
Inherited	yes

イニシャルキャップの色を、ブロックのほかのテキストの色とは変更したいとき、`axf:initial-letters-color` プロパティを指定します。

Property Name	axf:initial-letters-width
Value	auto   length   <percentage>
Initial	auto
Applies to	fo:block
Inherited	yes
Percentages	refer to the width of containing reference area
Property Name	axf:initial-letters-text-align
Value	start   center   end   left   right
Initial	end
Applies to	fo:block
Inherited	yes

axf:initial-letters-width プロパティへイニシャルキャップの文字幅を指定します。auto ではイニシャルキャップの文字列または画像から計算されます。イニシャルキャップの文字幅が axf:initial-letters-width="auto" で決まる値よりも狭いとき、axf:initial-letters-text-align プロパティでイニシャルキャップの揃えを制御可能です。

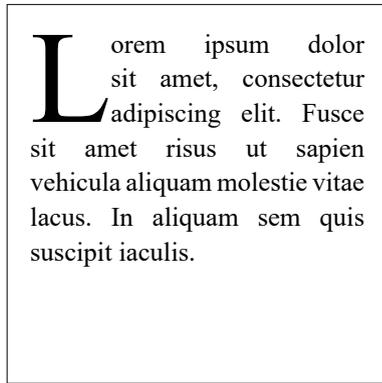
### 注

先頭のオブジェクトにはテキストではなく画像も配置可能です。このとき、出力した PDF からこの画像部分もテキストを読み取らせたければ axf:alttext プロパティが有用です。

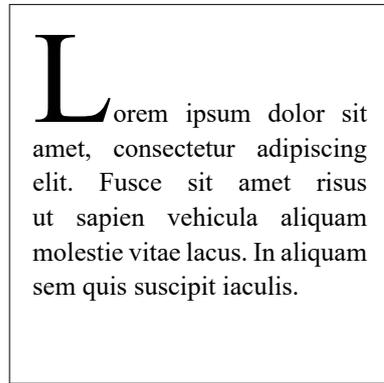
**ドロップキャップ** 大きくした先頭文字列の上端を 1 行目の続くテキストの高さ、下端を 2 行目以降のベースラインに揃えます。

**レイズドキャップ** 大きくした先頭文字列を 1 行目のベースラインに揃えます。

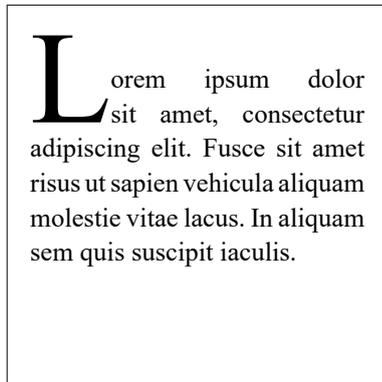
**サンクンキャップ** 大きくした先頭文字列の上端は 1 行目よりも高く、下端は 2 行目以降のベースラインに揃えます。



axf:initial-letters="3 drop"



axf:initial-letters="3 raise"



axf:initial-letters="3 2"

図 2.12-1 イニシャルキャップの例

axf:initial-letters プロパティの指定は位置によっては無視されます。ほか、注意事項についてはマニュアルをご覧ください。

Property Name	axf:initial-letters-first-line-head-height
Value	cap-height   x-height   auto
Initial	auto
Applies to	fo:block
Inherited	yes

ドロップキャップから続く先頭行のテキストの高さを合わせることを考えます。axf:initial-letters-first-line-head-height プロパティはドロップキャップの高さを、最初の行のどの位置に合わせるかについて指定可能です。

cap-height で cap-height に、x-height で x-height に、auto で自動判定となります。自動判定は通常 cap-height に合わせますが、ドロップキャップの次の文字が小文字の small-caps のとき、またはドロップキャップの行数と沈み込む行数が等しいとき x-height に合わせます。

axf:initial-letters="2 2"でブロックの行数が2であれば x-height です。

## 2.12.2 イニシャルキャップへテキストを食い込ませる

イニシャルキャップのある行より後ろのテキストの位置と、イニシャルキャップに続くテキストの位置を調整します。

`axf:initial-letters` プロパティへ `adjacent` を指定するとブロック行内すべての行をインデントするため、イニシャルキャップの文字が突き出して見えます。これをハンギングキャップと呼びます [図 2.12-2] (p.160)。

```
<!-- ハンギングキャップ -->
<fo:block axf:initial-letters="4 drop adjacent">
  Lorem ...
</fo:block>
```

`<integer2>` は、イニシャルキャップの文字の下にある行を指定された行数だけインデントします。`axf:initial-letters="3 drop 1"` と指定したとき、6 行の段落であれば、イニシャルキャップのある 3 行と次の 4 行目はインデントされますが、5 行目以降はインデントされません。

### 注

イニシャルキャップのある段落すべての行をイニシャルキャップ分インデントしてしまうと、次段落以降の指定によっては、インデントの分、この段落だけ凹んでみえてしまうかもしれません。あらかじめこの段落の `start-indent` プロパティを `-1dcm` としたり、次段落以降のインデントを大きく取るといった方法で回避可能です。

Property Name	<code>axf:initial-letters-end-indent</code>
Value	<code>&lt;length&gt;+</code>
Initial	<code>0pt</code>
Applies to	<code>fo:block</code>
Inherited	<code>yes</code>

`axf:initial-letters-end-indent` プロパティへ、イニシャルキャップの `end` 側のアキを指定します。この位置がイニシャルキャップに続くテキストの開始位置となります。値を複数指定したときは行ごとに適用されます。最後の値は対応する行以降に適用されます。

イニシャルキャップが 4 行の高さであれば、`1em 3em 2em` としたとき、1 行目はイニシャルキャップの文字から `1em`、2 行目は `3em`、3 行目が `2em`、4 行目が `2em` となります。

負の値を指定すると、ドロップキャップの文字のある領域へ各行の先頭を食い込ませることが可能です [図 2.12-3] (p.160)。

```
<!-- ドロップキャップへテキストを食い込ませる -->
<fo:block axf:initial-letters="4 drop"
  axf:initial-letters-end-indent="-0.25dcm -0.3dcm 0">
```

```
  Lorem ...  
</fo:block>
```

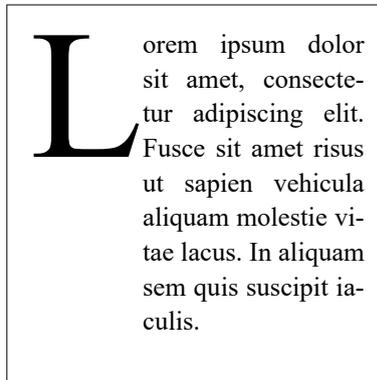


図 2.12-2 ハンギングキャップ

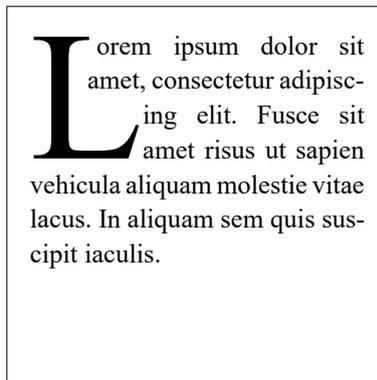


図 2.12-3 ドロップキャップへテキストを食い込ませる

### 2.12.3 イニシャルキャップに先行する約物を制御する

イニシャルキャップを行う段落先頭行の行頭に約物が含まれるときの処理を制御します。

Property Name	axf:initial-letters-leading-punctuation
Value	normal   hide   [ <length>    <color> ]
Initial	normal
Applies to	fo:block
Inherited	yes

axf:initial-letters-leading-punctuation プロパティへ、先頭に登場する約物のフォントサイズと色を指定します。

Property Name	axf:initial-letters-leading-punctuation-position
Value	normal   [[ hang   intrude]&& <length>?]
Initial	normal
Applies to	fo:block
Inherited	yes
Property Name	axf:initial-letters-leading-punctuation-shift
Value	normal   [[ baseline   before   middle   after]    <length>]
Initial	normal
Applies to	fo:block
Inherited	yes

axf:initial-letters-leading-punctuation-position プロパティへ先行する約物のインライン進行方向の配置、axf:initial-letters-leading-punctuation-shift プロパティへブロック進行方向の位置を指定します。

axf:initial-letters-leading-punctuation-position 意味	
normal	とくに何もしない。
hang	先行する約物を start 側へ突き出す。axf:hanging-punctuation と異なり、すべての先行する約物を突き出す。
intrude	イニシャルキャップを行う文字に食い込ませる。

normal 以外の指定では揃え位置とその位置からのシフト量が指定可能です。

axf:initial-letters-leading-punctuation-shift 意味	
normal   <length>	先行する約物は横組ではベースライン、縦組では中央に揃える。<length> のみの指定では normal からのシフト量となる。
baseline <length>	先行する約物を最後の行のベースラインに揃える。
before <length>	先行する約物をイニシャルキャップの before 側に揃える。
middle <length>	先行する約物をイニシャルキャップの中央に揃える。
after <length>	先行する約物をイニシャルキャップの after 側に揃える。

baseline はイニシャルキャップではなく、最後の行のベースラインであること、middle がイニシャルキャップの中央であることに注意してください。

約物を含むイニシャルキャップの例を示します [図 2.12-4] (p.162)。

```
<!-- 先行する約物を先頭行のテキストの位置、サイズで配置する -->
<fo:block ...
  axf:initial-letters="4 drop"
  axf:initial-letters-leading-punctuation="1em"
```

```

axf:initial-letters-leading-punctuation-position="hang"
axf:initial-letters-leading-punctuation-shift="before">
  “Lorem ipsum dolor sit amet, ...
</fo:block>

```

axf:initial-letters-leading-punctuation="1em"でイニシャルキャップのフォントサイズから段落のテキストのフォントサイズにしています。約物を axf:initial-letters-leading-punctuation-position="hang" でブロックから突き出し、axf:initial-letters-leading-punctuation-shift="before"としてイニシャルキャップの before、この FO では先頭行のテキストの位置に合わせています。

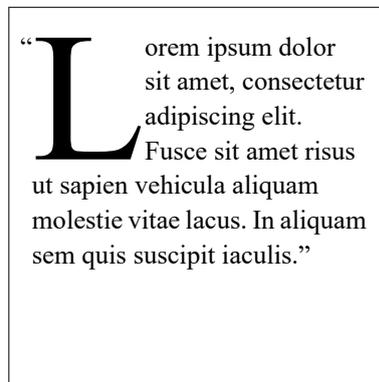


図 2.12-4 先行する約物を先頭行のテキストの位置、サイズで配置する

## 2.12.4 段落最終行の最低文字数を指定する

段落最終行に残る最低語数または文字数を指定します。

widows プロパティは主にページ分割が生じるときに分割された後半に配置される最低行数を指定するプロパティです。

段落の最終行の語、または文字数がひとつだけでは望ましくないときがあるでしょう。

Property Name	axf:avoid-widow-words
Value	false   true   [ <length>    <percentage> ]
Initial	auto
Applies to	fo:block
Inherited	yes
Percentages	refer to the width of containing block

axf:avoid-widow-words プロパティでは段落最終行の単語数または文字数について、最低値を指定します。true のとき、最終行が 1 語 (CJK は 1 字) になりません。

axf:line-break="bpil" のとき、<length> や <percentage> の値を指定するとその幅を最後の行に残そうとします。<length> や <percentage> を両方指定していると、小さい方が採用されます。axf:line-break="bpil" 以外では true と同じです。

## ⚠ 注意事項

他の分割バランスとの兼ね合いから、指定した通りになるとは限りません。行の長さがあまり取れないときや、追い込みができないときには効果は表れません。

CJK などは `axf:line-break="bpil"` の対象外のため、`<length>` や `<percentage>` を指定しても「最終行は 2 字以上」から変更されません。

## CJK の文末で約物を最終行の文字数カウントから除外する

`axf:avoid-widow-words="true"` のとき、最終行が「す。」のようになることもあります。これは CJK の約物も 1 字としてカウントされているためです。

Property Name	<code>axf:avoid-widow-words-cjk-punctuation</code>
Value	<code>auto</code>   <code>true</code>   <code>false</code>   <code>&lt;string&gt;</code>
Initial	<code>auto</code>
Applies to	<code>fo:block</code>
Inherited	<code>yes</code>
Option Property Name	<code>avoid-widow-words-cjk-punctuation</code>
Value	<code>true</code>   <code>false</code>   <code>&lt;string&gt;</code>
Initial	<code>false</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

`axf:avoid-widow-words-cjk-punctuation` プロパティへ `true` を指定すると、「'」（U+2019）、「”」（U+201D）そして句読点、閉じ括弧など UAX #14: Line Breaking Properties での CL（Close Punctuation）または CP（Close Parenthesis）に分類される文字がカウントから除外されます。

オプション設定 `avoid-widow-words-cjk-punctuation` プロパティへ `true` を指定すると、句読点、閉じ括弧など UAX #14: Line Breaking Properties での CL（Close Punctuation）または CP（Close Parenthesis）に分類される文字と RIGHT SINGLE QUOTATION MARK 「'」（U+2019）、RIGHT DOUBLE QUOTATION MARK 「”」（U+201D）がカウントから除外されます。`<string>` を指定すると、これらに加えてその文字列もカウントから外されます。たとえば `avoid-widow-words-cjk-punctuation="「なのです」"` とすると、「なのです。」を最終行に孤立させないようにします。

`axf:avoid-widow-words-cjk-punctuation="auto"` のとき、最終行の文字カウントの除外はオプション設定 `avoid-widow-words-cjk-punctuation` プロパティの指定に従います。

## 2.12.5 ページ、段最初の段落先頭のインデントを指定する

ページ、段の先頭での、段落先頭行のインデントを特別な値にします。

Property Name	axf:text-indent-if-first-on-page
Value	<length>   <percentage>   auto
Initial	auto
Applies to	fo:block, fo:block-container
Inherited	yes
Percentages	refer to the logical width of the containing block

axf:text-indent-if-first-on-page プロパティへ指定した値は、そのブロックの先頭行がページ、段の先頭に配置されるとき、text-indent プロパティの値として適用されます。

```
<!-- 見出し以外すべての段落に同じ
      axf:text-indent-first-on-pageとtext-indent -->
<fo:block ...>Heading</fo:block>
<fo:block ...
  axf:text-indent-if-first-on-page="2em"
  text-indent="4em">
  Lorem ipsum ...
</fo:block>
...
```

[図 2.12-5] (p.165) は見出し以外のブロックで、すべて axf:text-indent-if-first-on-page="2em" と text-indent="4em" を指定しています。図の左の段では「Heading」と記述された見出しブロックが先頭のため、次のブロックは text-indent="4em" で 4 cm インデントされます。真ん中の段の最初は前の段落が続いているため、とくにインデントされていません。右の段の最初は段落の開始となっているため、axf:text-indent-if-first-on-page="2em" で 2 cm インデントされています。

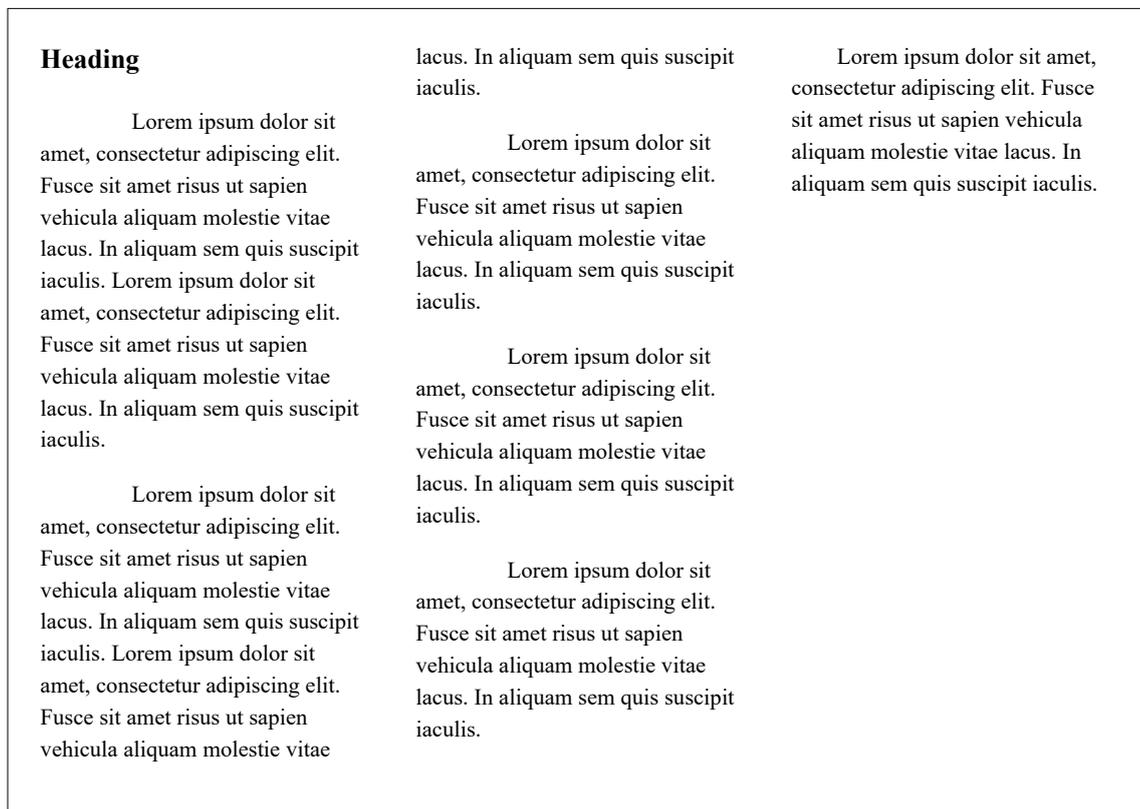


図 2.12-5 axf:text-indent-if-first-on-page の動作

## 2.12.6 先頭行の揃えを制御する

ブロック先頭行の揃えを指定します。

XSL 1.1 仕様には最終行の揃えを制御する text-align-last プロパティがありますが、先頭行の揃えを制御するプロパティはありません。

Property Name	axf:text-align-first
Value	relative   start   center   end   justify   inside   outside   left   right
Initial	relative
Applies to	fo:block
Inherited	yes

axf:text-align-first プロパティへ先頭行の揃えを指定します。

relative を指定したとき、何もしません。他の値は text-align プロパティと同様です。

1 行目のほか、直前行の最後が LINE FEED (U+000A) である行で機能します。

axf:text-align-first プロパティは text-align-last プロパティより優先します。

## 注

たとえばブロックへ `text-align="justify"` と `text-align-last="right"` を指定したとき、記述するテキストが 1 行であれば、その行は最終行として、`text-align-last="right"` に従い右揃えとなります。

`text-align-last="justify"` とすれば均等割り付けになりますが、同じ指定をしたブロックで 2 行以上のとき、最終行が均等割り付けになります。`axf:text-align-first="justify"` を指定すると、「1 行のときは均等割り付け、それ以上の行数のとき通常の行は均等割り付けで最終行は右揃え」のように指定可能です。

長い目次や索引項目の体裁調整にも有用です。

## 2.12.7 最終行の均等割り付けを制御する

最終行で均等割り付けを行うかどうかを、内容によって変更します。

テキストの揃えに `justify` を指定すると、テキストは均等割り付けされます。最終行で文字数が少ないときに均等割り付けを行うと、読みづらくなることがあるでしょう。

AH XSL Formatter では、最終行のアキに応じて均等割り付けにするかどうかを制御可能です。

Property Name	<code>axf:flush-zone</code>
Value	<code>none</code>   <code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>
Initial	<code>none</code>
Applies to	<code>fo:block</code>
Inherited	<code>yes</code>
Percentages	refer to the width of containing block

最終行の行末のアキが `axf:flush-zone` プロパティへ指定した値以下のとき、最終行のテキストの揃えが `justify` として処理されます。

`text-align-last="left"` が指定されているとき、最終行の揃えは `axf:flush-zone` プロパティへ指定した値以上にアキがあれば `left`、以下であれば `justify` として処理されます。このとき、`text-align="justify"` である必要があります。

`none` のとき、とくに作用しません。

```
<!-- axf:flush-zoneが作用する-->
<fo:block
  text-align-last="left" text-align="justify"
  axf:flush-zone="30%">
  ...
</fo:block>
<!-- axf:flush-zoneが作用しない-->
<fo:block
  text-align-last="left" text-align="left"
```

```
    axf:flush-zone="30%">
    ...
</fo:block>
    <!-- axf:flush-zoneが作用しない-->
<fo:block
    text-align-last="justify" text-align="justify"
    axf:flush-zone="30%">
    ...
</fo:block>
```

## 2.13 節 行分割、禁則制御

行分割に用いるアルゴリズム、禁則、ハイフネーションを制御します。

### 行分割規則を制御する

AH XSL Formatter では 2 種類の行分割処理を行います。また、CJK での禁則処理対象とする文字についてもいくつかの制御方法を持ちます。

Property Name	axf:line-break
Value	auto   [[ normal   strict   <del>loose</del>   anywhere]    [ line   bpil]]
Initial	auto
Applies to	all block-level formatting objects
Inherited	yes

axf:line-break プロパティで、行分割アルゴリズムの制御が可能です。値について [CSS3-Text] Breaking Rules for Punctuation: the ‘line-break’ property を基にしていますが、AH XSL Formatter での変更点があります。主に欧文のための処理方法の指定として line、bpil が追加されています。また、loose と anywhere はありません。

normal と strict が CJK、bpil と line が欧文などの行分割規則指定の値です。

axf:line-break	意味
auto	normal と同じ。
line	行分割を 1 行ずつ逐次実行する。
bpil	Knuth-Plass の行分割アルゴリズムを利用し、段落で文字数をバランスする。
normal	CJK の Nonstarter 文字のうち、JIS X 4051 『日本語文書の組版方法』での行頭禁則和字は禁則処理しない。
strict	CJK の Nonstarter 文字を禁則処理する。

ブロックの言語は xml:lang や language プロパティで明示されているかオプション設定 default-lang プロパティで指定されているものとなります。

AH XSL Formatter における行分割処理で bpil、line の処理対象や制御の詳細については「欧文の行分割を制御する」の項をご覧ください。

CJK の禁則処理制御の詳細な指定については「禁則を制御する」の項をご覧ください。

処理対象の文字など、行分割制御全体の詳細はマニュアルをご覧ください。

### 2.13.1 欧文の行分割を制御する

欧文の行分割アルゴリズムとその制御を指定します。

**Breaking Paragraphs into Lines (BPIL)** 段落ごとに、それぞれの行長が均等に近くなるよう行分割を行うアルゴリズムです。BPIL は欧文のような、文を構成する単語がスペースで分割され、そのアキが一定とはならない言語で効果を発揮するアルゴリズムです。TeX や Apache FOP といった組版システムで採用されています。

`axf:line-break="bpil"`であるか、`axf:line-break` プロパティが無指定のとき BPIL の適用対象の言語として設定されている言語のブロックに対して、BPIL による行分割が適用されます。

同じテキストに対し `axf:line-break` プロパティへ `line` と `bpil` を指定した例を示します [図 2.13-1] (p.169)。`line` を指定したブロックでは前半の行に単語が多めに配置され、その後方の行と単語間のアキに差があります。`bpil` を指定したブロックでは、単語間のアキが最終行以外で同程度となっています。

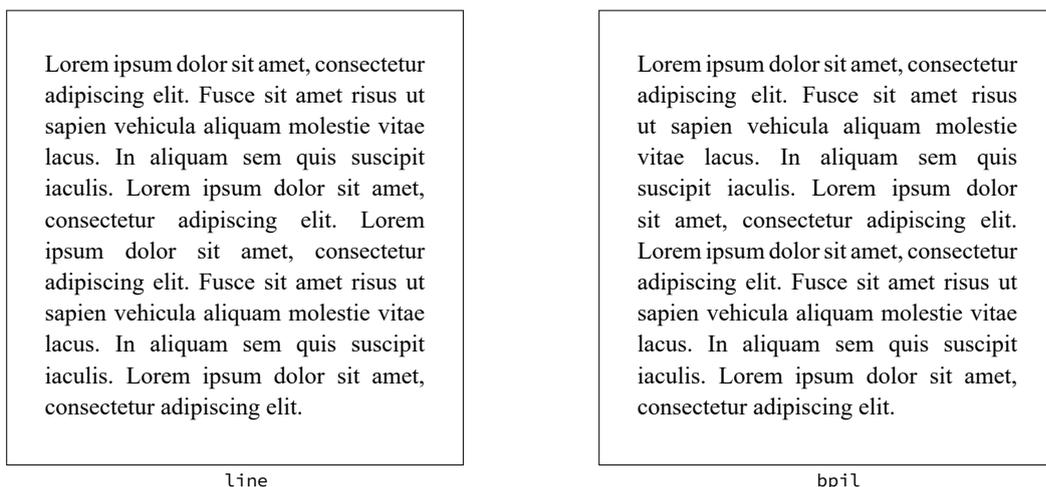


図 2.13-1 `axf:line-break` による欧文処理

## BPIL の適用言語を指定する

Option Property Name	<code>bpil</code>
Value	<code>&lt;string&gt;</code>
Initial	Latn Grek Cyrl Armn Geor
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定 `bpil` プロパティへ言語またはスクリプトを指定すると、BPIL による行分割の対象の言語となります。スクリプトを指定したとき、そのスクリプトが代表するような言語を対象として設定します。

空白区切りで指定します。指定可能な値は、次に挙げるものです。

- 言語コード (例: `en`)
- 言語タグ (例: `en-US`)
- スクリプト (例: `Latn`)

bpil プロパティで指定した言語のブロックで、BPIL を適用しない箇所があるのであれば、`axf:line-break="line"`で明示するとよいでしょう。

## BPIL の動作を制御する

Option Property Name	bpil-limit-chars
Value	<integer>
Initial	50000
Applies to	<formatter-settings>
Option Property Name	bpil-minimum-line-width
Value	<number>
Initial	8
Applies to	<formatter-settings>
Option Property Name	bpil-penalty-hyphenation
Value	<integer>
Initial	100
Applies to	<formatter-settings>

BPIL は段落を通した行分割の最適化を行うため、段落が長く、文字数が多くなるほど処理負荷が増大します。

オプション設定 `bpil-limit-chars` プロパティは BPIL を適用する最大文字数を指定します。およそこの文字数を超えるブロックでは、BPIL を適用しません。

オプション設定 `bpil-minimum-line-width` プロパティは BPIL を適用する最小の行幅を em 単位で指定します。1 未満は 1 とみなされます。

`hyphenation="true"`かつ BPIL の適用ブロックでは、ハイフネーションを考慮して BPIL が適用されます。

オプション設定 `bpil-penalty-hyphenation` プロパティはハイフネーションの起こりやすさを 0 から 1000 の範囲で指定します。0 に近いほどハイフネーションが起こりやすくなります。

## BPIL の適用外について

次の項目の内容が BPIL の適用対象ブロックに含まれるとき、BPIL は適用されず組版され、AH XSL Formatter は警告を表示します。

- リーダ (<fo:leader>など) を含む
- フロートを含む、複雑に食い込んでいる (単純な食い込みでは BPIL が適用されることもある)
- フォームフィールドを含む
- ルビを含む

- BIDI 処理対象
- `axf:indent-here` を含む
- `<axf:tab>`や、`axf:tab-treatment="preserve"`によって除かれない TAB を含む
- `wrap-option="no-wrap"`など、行分割が制限されない行でのオーバーフローをした行を含む
- フォントサイズまたは行の高さが 0 以下
- `bpil-minimum-line-width` プロパティへ指定した行幅以下の領域
- `bpil-limit-chars` プロパティへ指定した文字数を超える
- 版面サイズが異なるページに切り替わっている

また、`<fo:initial-property-set>`要素が含まれるブロックの先頭行では BPIL は適用されません。ほか、3 ページ（段）以上に渡るブロックの 2 ページ（段）目以降では BPIL は適用されないことがあります。

以上のことから、段組、とくに複雑なものでは BPIL による最適化は行いにくくなっています。

## 2.13.2 単語の分割規則を制御する

単語の分割規則を指定可能です。

ここでの単語は Unicode の General Category が L（文字）、M（結合マーク）、N（数字）で文字列が構成されているものを指します。

### 単語内での行分割を制御する

通常、単語はハイフネーション規則などにより分割可能な位置が決定されますが、それよりも強制的に分割の制御が可能です。

Property Name	<code>axf:word-break</code>
Value	<code>normal</code>   <code>keep-all</code>   <code>break-all</code>   <b><code>break=word</code></b>   <code>keep-non-spaces</code>
Initial	<code>normal</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes

`axf:word-break` プロパティは [CSS3-Text] Breaking Rules for Letters: the ‘word-break’ property を基にしたプロパティです。

`normal` のとき、通常の実行分割規則に従います。

`keep-all` を指定したとき、単語は分割されません。単語以外の箇所は通常の実行分割規則に従います。`hyphenate="true"`は無視されるため、分割させず、ハイフネーションも起こしたくないときに有用です。

`break-all` を指定したとき、スクリプトが Latn、Cyril、Grek、Zyyy のいずれかであれば、単語内のすべての箇所が行分割可能とみなされます。そのほかのスクリプトでは通常の実行分割規則に従います。

`keep-non-spaces` を指定したとき、スペース類以外の文字間で分割しません。たとえば、「ハイフンを含む語が連続するが、ハイフン箇所での分割を避けたい」といったときに有用です。

## 単語が長いときに行分割を指定する

`afx:word-break="break-all"` の指定は強制力が強く、単語途中の好ましくない箇所でも分割が発生することがあります。単語が分割できず、行長よりも長いときにのみ分割を行うのであれば、次の拡張仕様が有用です。

Property Name	<code>afx:word-wrap</code>
Value	<code>normal</code>   <code>break-word</code>   <code>break=spaces</code>
Initial	<code>break-word</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes

`afx:word-wrap` プロパティは [CSS3-Text] Overflow Wrapping: the overflow-wrap/word-wrap property を基にしたプロパティです。行の長さよりも長く分割可能でない単語、URL などに有用です。

`normal` では分割されず、長い単語はテキストが領域をはみ出します。

`break-word` ではどこか適当な箇所が無理やり分割をします。`afx:word-break="break-all"` のときよりも望ましい箇所で分割されます [図 2.13-2] (p.172)。

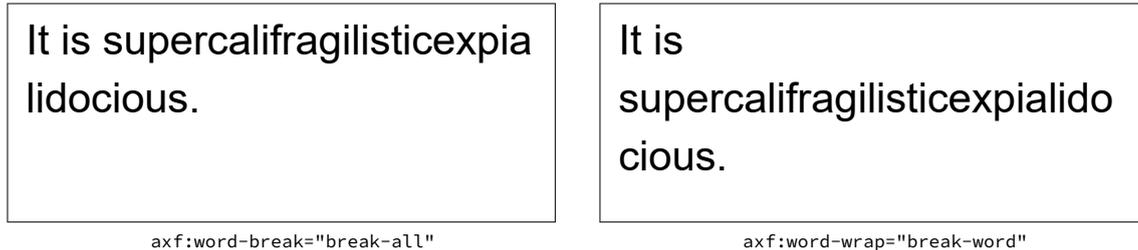


図 2.13-2 `afx:word-break` と `afx:word-wrap` の比較

### 2.13.3 分割禁止語を登録する

分割されたくない語をオプション設定で指定します。

インライン要素やスクリプトに分割を禁止する指定を記述することも可能ですが、FO 全体で分割を禁止したい語の設定が可能です。

Option Element Name	<code>&lt;unbreakable-words&gt;</code>
Parent	<code>&lt;formatter-settings&gt;</code>

オプション設定 `<formatter-settings>` の子要素として、`<unbreakable-words>` を記述します。

`<unbreakable-words>` 要素の内容として分割禁止したい語を改行区切りで記述します。

```

<!-- 分割禁止語の指定 -->
<formatter-settings>
  <unbreakable-words>
PDF1.3
PDF1.4
...
  </unbreakable-words>
</formatter-settings>

```

## 2.13.4 条件によってハイフネーションを抑制する

複数のルールでハイフネーションを抑制します。

### ハイフネーションを抑制する

ハイフネーションされる行が連続するとき、XSL 1.1 仕様の `hyphenation-ladder-count` プロパティへその行数の上限を指定可能です。

AH XSL Formatter では単語の文字数や行幅の残りの長さといった、より細かな条件による制御が可能です。

Property Name	<code>axf:hyphenation-minimum-character-count</code>
Value	<code>&lt;number&gt;</code>
Initial	1
Applies to	<code>fo:block</code> , <code>fo:character</code>
Inherited	yes

ハイフネーション処理を行う最低文字数を `axf:hyphenation-minimum-character-count` プロパティへ指定します。1 以上の整数を指定してください。

Property Name	<code>axf:hyphenation-zone</code>
Value	<code>none</code>   <code>length</code>   <code>percentage</code>
Initial	<code>none</code>
Applies to	<code>fo:block</code>
Inherited	yes
Percentages	refer to the width of containing block

`axf:hyphenation-zone` プロパティへ 0 以上の値を指定したとき、単語末から行末までの幅が指定値以下であれば、その行で単語の分割が行われません。

0 以下の値のとき無効です。

Property Name	axf:hyphenate-hyphenated-word
Value	true   false
Initial	true
Applies to	fo:block
Inherited	yes

axf:hyphenate-hyphenated-word プロパティへ false を指定したとき、元のテキストでハイフンを含む語は、ハイフネーション処理によるハイフン追加を抑制されます。このとき、既に単語内に存在するハイフン箇所での行分割は抑制しません。「axf:hyphenate-hyphenated-word」という語であれば、「axf:hyphenate-」と「hyphenated-」の箇所で行分割される可能性があります。

true のとき、行末での分割によりハイフンを付加することがあります。

ハイフン付けされているとみなされる文字は、HYPHEN-MINUS 「-」 (U+002D)、SOFT HYPHEN 「-」 (U+00AD)、HYPHEN 「-」 (U+2010)、NON-BREAKING-HYPHEN 「-」 (U+2011) です。

## ページ、段末の語のハイフネーション処理を変更する

Option Property Name	hyphenation-keep-mode
Value	word   line
Initial	word
Applies to	<formatter-settings>

hyphenation-keep プロパティへ page や column を指定したとき、ページ、段末の語がハイフネーションされる際の処理はオプション設定 hyphenation-keep-mode プロパティへの指定によって決定します。

word ではハイフネーションされている語を次のページ、段へ追い出します。

line ではその語を含む行が次のページ、段へ追い出されます。

word によって語を追い出した結果 widows の行数制限にひっかかるとき、行が次のページ、段に送られます。この結果 hyphenation-keep の指定に違反していると、さらに行を追い出します。widows、hyphenation-keep プロパティの組み合わせによっては違反を解消できないときもあります。

行を追い出したとき、最終行にシフトした行の最後の語もハイフネーションされる語であるとまた次のページ、段へ送られます。これをある程度で抑えるには hyphenation-ladder-count プロパティを併用し、ハイフネーションで終わる行の連続を制限します。

### 2.13.5 ハイフネーション情報を制御する

ハイフネーション例外辞書を作成し、独自にハイフネーション処理を制御します。

AH XSL Formatter ではハイフネーション用の規則をユーザが用意する必要はありません。しかし、標準のハイフネーション処理では望ましい組版が行えないものを例外として扱うため、ハイフ

ネーション例外辞書を用意することも可能です。GUIでは「組版」メニューの「ハイフネーション例外辞書再読み込み」によって再読み込み、再組版を行ってください。

例外辞書はAH XSL Formatterのインストールフォルダの「hyphenation」フォルダか、環境変数にフォルダのパスを指定します。

辞書ファイル名は言語コード、言語タグ、または言語タグのHYPHEN-MINUS「-」(U+002D)の代わりにLOW LINE「\_」(U+005F)を使用しているもの(例:「en\_us」)に拡張子.xmlを付加したものです。

ファイル名に含む言語コードは、2字であればそれを、それ以外ではTerminologyコードを指定します。国名コードも2文字コードがあればそれを指定します。

ハイフネーション例外辞書の優先順、インストールフォルダや環境変数についての詳細はマニュアルをご覧ください。

## ハイフネーション例外辞書を記述する

Element Name	<hyphenation-info>
Element Name	<hyphen-char>
Parent	hyphenation-info
Element Name	<hyphen-min>
Parent	hyphenation-info
Element Name	<exceptions>
Parent	hyphenation-info
Element Name	<hyphen>
Parent	exceptions
Element Name	<non-eol-words>
Parent	hyphenation-info

ハイフネーション例外辞書は<hyphenation-info>をルート要素として、ハイフネーション文字を表す<hyphen>と<hyphen-char>例外辞書のデータを指定するための<exceptions>、行分割位置前後の最小文字数を指定するための<hyphen-min>、行末禁則される語を指定するための<non-eol-words>によって構成されます。

FOに<axf:hyphenation-info>の子として記述するときには、例外辞書のネームスペースを付加する必要があります。例外辞書のネームスペースは次の通りです。本書ではaxhとして短縮表記しています。

<http://www.antennahouse.com/names/XSL/Hyphenations>

## FO にハイフネーション情報を追加する

Element Name	<axf:hyphenation-info>
Property	src, language
Parent	fo:declarations

ハイフネーション例外辞書の XML ファイルを指定したフォルダの中に置いておくほかに、<fo:declarations>の子要素として<axf:hyphenation-info>へ記述が可能です。src プロパティへファイルを指定するか、子要素として直接記述します。ただし、<axf:hyphenation-info>の子要素としては<axh:exceptions>のみが指定可能です。

## ハイフネーションに TeX の辞書を利用する

AH XSL Formatter はハイフネーション処理を 40 以上の言語に対応した独自処理で行います。ほかに、Apache FOP と同じ形式の XML ファイルとして TeX の辞書によるハイフネーションに対応しています。詳細についてはマニュアル、Apache FOP の Web サイトをご覧ください。

Option Property Name	HyphenationOption
Value	true   false
Initial	true
Applies to	<formatter-settings>

オプション設定 HyphenationOption プロパティへ false を指定すると、ハイフネーションに参照する辞書に TeX の辞書を利用します。このとき TeX の辞書にないハイフネーション処理は行われません。

利用する言語に対応した辞書をすべて用意してください。

## 一部のハイフネーションのみ TeX の辞書を利用する

Option Property Name	hyphenation-TeX
Value	<string>
Initial	empty string
Applies to	<formatter-settings>

オプション設定 hyphenation-TeX プロパティへ、ハイフネーション処理を TeX の辞書で行いたい言語をカンマ区切りで列挙します。

### 2.13.6 ハイフネーション例外情報を追加する

ハイフネーション例外情報を追加します。

記述の例を示します。

```
<!-- ハイフネーション辞書の記述 -->
<hyphenation-info>
  <hyphen-char value="" />
  <exceptions>
    as=so=ciate
    ta<hyphen />ble
  </exceptions>
</hyphenation-info>
```

<exceptions>に単語とハイフン位置を改行区切りで指定します。

ハイフン位置は<hyphen>または<hyphen-char>要素の value プロパティへ指定した文字で記述します。

## 2.13.7 ハイフネーション前後の最小文字数を制御する

行分割でハイフネーション処理される時、ハイフンの前後に残る最小文字数を指定します。

### ハイフネーション前後に残る最低文字数を指定する

Property Name	hyphenation-remain-character-count, hyphenation-push-character-count
Value	<number>   auto
Initial	auto
Applies to	all-elements

行分割でハイフネーション処理が行われるとき、hyphenation-remain-character-count プロパティへ行末に残される文字数、hyphenation-push-character-count プロパティへ次行の行頭に追し出される文字数を最低文字数を指定します。

AH XSL Formatter では auto が追加されています。auto のとき、<hyphen-min>に有効な値が指定されていればその値、無指定のときは 2 とみなします。

### <hyphen-min>での指定を有効にする

Option Property Name	hyphen-min
Value	false   true
Initial	false
Applies to	<formatter-settings>

ハイフネーション例外辞書の<hyphen-min>要素で記述したハイフネーション処理前後の最低文字数の指定は、オプション設定 hyphen-min プロパティへ true を指定したとき、有効になります。

## ハイフネーション前後の最小文字数を指定する

<hyphen-min>要素のプロパティでハイフネーション処理のときの行末、行頭の最低文字数を制御可能です。

before プロパティへ行末に残る最小文字数を、after プロパティへ次行に追い出される最小文字数を指定します。

```
<!-- ハイフネーション前後の最小文字数を指定する -->
<hyphenation-info>
  <hyphen-min before="3" after="2" />
</hyphenation-info>
```

### 2.13.8 大文字の語のハイフネーション処理を制御する

語が大文字で構成されるときにハイフネーションを抑制します。

hyphenate="true"であるとき、行末の語はハイフネーション規則に従い分割されます。

Property Name	axf:hyphenate-caps-word
Value	true   false
Initial	true
Applies to	fo:block
Inherited	yes

語が大文字で構成されるとき、ハイフネーションを行うかどうかを axf:hyphenate-caps-word プロパティで制御可能です。false を指定したとき、ハイフネーションをしません。

「XSL-FO」のように元の語がハイフンを含むときは複数の単語とみなされ、ハイフン位置で行分割の可能性があります。

スモールキャピタルとして処理された文字でも元が小文字であれば小文字とみなされます。

#### 注

スモールキャピタル処理された語で、頭文字語などハイフネーションされたくない語のとき、その語を<fo:inline>要素などで囲い、hyphenate="false"を指定しましょう。

### 2.13.9 CJK の禁則を制御する

CJK において、行頭に表われてはいけない文字や行末に表われてはいけない文字を制御します。

AH XSL Formatter における CJK の行分割アルゴリズムは UAX #14: Line Breaking Properties に変更を加えたものになります。たとえば、全角括弧類に対しては分割を許可するように処理します。

axf:line-break="normal"のとき、UAX #14: Line Breaking Properties において NonStarter に分類される文字は行頭禁則として行分割のときに考慮されます。

axf:line-break="strict"では加えて JIS X 4051 『日本語文書の組版方法』で行頭禁則和字に指定された文字も考慮されます。

## 行頭、行末の禁則を追加、除外指定する

行頭、行末の禁則へ追加、除外を指定可能です。

Property Name	axf:append-non-starter-characters, axf:append-non-end-of-line-characters
Value	<string>
Initial	empty string
Applies to	fo:page-sequence
Property Name	axf:except-non-starter-characters, axf:except-non-end-of-line-characters
Value	<string>
Initial	empty string
Applies to	fo:page-sequence
Option Property Name	append-non-starter-characters, append-non-end-of-line-characters
Value	<string>
Initial	empty string
Applies to	<formatter-settings>
Option Property Name	except-non-starter-characters, except-non-end-of-line-characters
Value	<string>
Initial	empty string
Applies to	<formatter-settings>

NonStarter として扱う文字の追加を axf:append-non-start-character プロパティへ、除外を axf:except-non-start-character プロパティへ指定します。たとえば、「段組で行長が取れないときに小書きのかな文字や長音記号を行頭禁則から除外する」といった利用が考えられます。

行末禁則文字の追加を axf:append-non-end-of-line-character、除外を axf:except-non-end-of-line-character プロパティへ指定します。

追加の行頭、行末の禁則文字の初期値をそれぞれオプション設定 append-non-starter-characters、append-non-end-of-line-characters プロパティへ指定可能です。

同様に、禁則文字からの除外指定指定をオプション設定 `except-non-starter-characters`、`except-non-end-of-line-characters` プロパティへ指定可能です。

## IDEOGRAPHIC SPACE (U+3000) を行頭禁則に含むかを指定する

Option Property Name	<code>non-starter-ideographic-space</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

IDEOGRAPHIC SPACE (U+3000) は Unicode 6.3 より行頭禁則として扱われます。

和文で FULLWIDTH QUESTION MARK 「？」(U+FF1F) や FULLWIDTH EXCLAMATION MARK 「！」(U+FF01) で終了する文の後に IDEOGRAPHIC SPACE を挿入することを考えてみましょう。この禁則が有効であれば「ありがとう！&#x3000;さようなら！」というテキストで「ありがとう！」が行末になるとき、IDEOGRAPHIC SPACE が行頭にならず「！&#x3000;」が次行へ送られ、「！」が行頭禁則であるとき「う！&#x3000;」が次行に送られます。

オプション設定 `non-starter-ideographic-space` プロパティでこの禁則が制御可能です。true のとき有効、false のとき無効です。

### 2.13.10 ポーランド、ハンガリー、チェコ語の禁則を指定する

行末に残される文字数に制限がある禁則を処理します。

ポーランド語、ハンガリー語、チェコ語などでは、1文字の単語を行末に置いてはならない禁則があります。

これらの禁則処理を指定可能です。

Option Property Name	<code>avoid-orphan-single-word</code>
Value	<code>true</code>   <code>false</code>   <code>&lt;string&gt;</code>   <code>empty</code>
Initial	<code>pol,hun,ces</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定 `avoid-orphan-single-word` プロパティへ言語コードをカンマ区切りで指定すると、指定した言語では行末に1文字の単語が置かれないう処理されます。

指定が true のときすべての言語が対象となり、指定が false か空のとき処理されません。

### 2.13.11 SOLIDUS 「/」 (U+002F) を伴う語の分割を抑制する

SOLIDUS 「/」(U+002F) を含む語がその箇所分割されることを抑制します。

URL やパスの表記に登場する SOLIDUS 「/」(U+002F) はその箇所改行されてもあまり意味を損ないませんが、「k/m」のように組み合わせる単位を示す語は改行によって「k/」「m」のように分

割されると望ましくない場合があります。SOLIDUS は既定では「この直後で分割してよい文字」ですが、この動作の変更が可能です。

Property Name	axf:abbreviation-character-count
Value	auto   <number>
Initial	auto
Applies to	fo:block
Inherited	yes
Option Property Name	abbreviation-character-count
Value	<number>
Initial	3
Applies to	<formatter-settings>

axf:abbreviation-character-count プロパティへ、SOLIDUS の後に残す文字数を 0 以上 10 以下の整数で指定します。

「km/minutes」という語に対し axf:bbreviation-character-count="3"が指定されているとき、「km/min」「utes」のように分割が起こる可能性があります。

0 を指定すると、この制御を行いません。10 を超える数字を指定したときは 10 とみなされます。

## 2.13.12 テキストの途中で次行以降のインデントを指定する

ブロックに対してではなく、テキストの途中で次の行以降のインデントを追加で指定します。

Property Name	axf:indent-here
Value	none   <length>
Initial	none
Applies to	fo:inline

axf:indent-here プロパティに、このプロパティを指定した<fo:inline>の先頭位置を基準として、指定した長さを加えた値が、折り返しによる次行以降のインデントとなります。

none のとき何も行いません。

章番号を含む見出しで、章題が改行時に章番号の区切り位置以降にインデントされるよう指定する例を示します [図 2.13-3] (p.182)。

```
<!-- axf:indent-hereの利用例 -->
<fo:block ...>
  第一章:<fo:inline axf:indent-here="0pt"/>そこそこに長いタイトル
</fo:block>
```

## 第一章：そこそこに長い タイトル

図 2.13-3 axf:indent-here の利用例

axf:indent-here="0pt"を指定した<fo:inline>で「第一章」を囲ってしまうと、「第」の開始位置が axf:indent-here プロパティによるインデントの基点となってしまいます。

axf:indent-here プロパティによるインデントは継続行で機能します。たとえば<fo:block>を空要素で挿入するような強制改行を行うとき、機能しません [図 2.13-4] (p.182)。

axf:indent-here プロパティによるインデントを有効にしなが指定したテキスト位置で改行を行いたいとき、<axf:tab>要素によって残りの行幅を埋める方法などが考えられます。

```
<!-- axf:indent-hereが機能しない例 -->
<fo:block ...>
  第一章:<fo:inline axf:indent-here="0pt"/>そこそこに<fo:block/>長いタイトル
</fo:block>
```

## 第一章：そこそこに 長いタイトル

図 2.13-4 axf:indent-here が機能しない例

### 2.13.13 SOFT HYPHEN (U+00AD) の動作を変更する

特殊なフォントの表示などのため、SOFT HYPHEN の動作を変更します。

通常 SOFT HYPHEN (U+00AD) は行分割のときのみ表示されます。しかし絵文字のようなフォントで SOFT HYPHEN に字形を割り当てているときがあります。

Property Name	axf:soft-hyphen-treatment
Value	auto   preserve
Initial	auto
Applies to	all formatting objects
Inherited	yes

axf:soft-hyphen-treatment プロパティに preserve を指定すると SOFT HYPHEN は常に表示されます。auto は通常通りの処理です。

FO に axf:soft-hyphen-treatment の指定を行った例を示します [図 2.13-5] (p.183)。Wingdings は Windows に収録されている、特殊な字体を持つフォントです。

```
<!-- SOFT HYPHENを常に表示する -->
<fo:block>abc&#xAD;xyz</fo:block>
```

```
<fo:block axf:soft-hyphen-treatment="preserve"
font-family="Wingdings">abc&#xAAD;xyz</fo:block>
```



図 2.13-5 通常（左）の処理と SOFT HYPHEN を表示した処理（右）

## 2.14 節 段落の配置、行グリッド

行グリッドや、ページ位置によるブロック表示の制御を指定します。

### 2.14.1 行グリッドを制御する

ベースラインを基準とするグリッドを仮定し、テキストや図表の配置を調整します。

行グリッド 行とブロックの配置を、ページ上に仮定したグリッドへ揃える機能を行グリッドと呼びます。AH XSL Formatter ではフォント情報に基づいたグリッドを行グリッドとして設定可能です。

適切に指定することで、次のような制御が可能です。

- 一定の行ピッチが保たれることで文書を読みやすくする。
- 段組の段同士で行の位置が揃うことで見栄えをよくする。
- ページ表裏の行の位置が揃うようにすることで体裁をよくする。
- 版面の最初の行の前と最後の行の後に意図しないアキができないようにする。
- テキストを配置するエリアの寸法を整数の文字数と行数で収まるように調整する。

Property Name	axf:baseline-grid
Value	normal   none   root   new
Initial	normal
Applies to	block-level formatting objects, fo:flow and fo:static-content / block containers

axf:baseline-grid プロパティへ行グリッドを指定します。

axf:baseline-grid	意味
normal	行グリッドを新たに設定も解除もしない。
none	行グリッドを解除。
root	ルート要素に指定したフォントと行の高さを利用して行グリッドを設定する。
new	新しい行グリッドを設定する。この要素のフォントと行の高さが行グリッドに利用される。

axf:baseline-grid プロパティはブロックレベルのオブジェクトで指定可能であるので、たとえば、<fo:flow>要素へ axf:baseline-grid="root"で本文に行グリッドを設定し、ブロック数式では axf:baseline-grid="none"として行グリッドを行わない、という指定が可能です。

axf:baseline-grid プロパティの指定は継承されません。親で axf:baseline-grid="new"を指定したとき、子要素は無指定ならば axf:baseline-grid="normal"として親要素で設定された行グリッドが適用されます。

また、テキストを含まない図や、ほかの箇所と異なるグリッドを持つブロックと、その外側の行グリッドとの揃えを指定可能です。

Property Name	axf:baseline-block-snap
Value	none   [ auto   before   after   center ]    [ border-box   margin-box ]
Initial	auto border-box
Applies to	block-level elements

ブロックを行グリッドと揃える位置を axf:baseline-block-snap プロパティへ指定します。

axf:baseline-block-snap 意味	
none	ブロックを行グリッドに揃えない。
auto	ページ、段の先頭では before、最後では after、それ以外では center。
before	ブロックの before 側の端を行グリッド上の text-before-edge ラインに揃える。
after	ブロックの after 側の端を行グリッド上の text-after-edge ラインに揃える。
center	ブロックを行グリッド上の text-before-edge ラインと text-after-edge ラインの間で中央寄せにする。

このとき、ブロックの端としてマージンボックス (margin-box) とパディングボックス (padding-box) どちらを用いるかを指定します。

## 2.14.2 ページ、段先頭のブロックの表示を制御する

ブロックがページや段の先頭に来たときの表示を制御します。

いくつかブロックが続くようなとき、そのブロックがページや段の先頭に配置されるかどうかは組版時に決定することもあります。ブロックがページ、段の先頭に配置されたかどうかによって、表示の制御が可能です。

Property Name	axf:suppress-if-first-on-page
Value	true   false   unless
Initial	false
Applies to	fo:block, fo:block-container

axf:suppress-if-first-on-page プロパティに true を指定すると、ブロックがページや段の先頭に来たときに削除します。false のとき、何も影響しません。

unless では、ブロックがページ先頭または段先頭以外に来たとき、それを削除します。

削除されたブロックは、実際には高さのないブロックとして存在し、id の参照などは有効です。

### 注

段組では、span="all"を指定したブロックがあるとき、ブロックの箇所で改段されます。このブロックをテキストのセパレータとして利用し、段の先頭にセパレータが表示されて欲しくないとき axf:suppress-if-first-on-page="true"が有用です。

## 2.15 節 はみ出し（オーバーフロー）処理

テキストが決められた範囲をはみ出すとき（オーバーフロー）の制御を詳細に指定します。

### 2.15.1 はみ出し時の処理を指定する

はみ出し時に指定可能な処理が拡張されています。処理方法の指定を行った上でそれぞれの処理の詳細を拡張プロパティへ指定します。

テキストを配置するとき、配置可能な領域の端に到達すると、次の領域からテキストの配置が再開されます。しかし、表組のセルやチラシの項目など、先に表示できる範囲やページが決まっており、範囲内にテキストを収めたいときもあるでしょう。

無理にテキストの改行を抑制したり、高さの決まっているブロックに配置できる以上のテキストの流し込みを行うと、領域からはみ出し（オーバーフロー）が発生します。

はみ出しにはブロック進行方向 [図 2.15-1] (p.186) と、インライン進行方向が存在します。

はみ出しが生じたときの処理方法を、`overflow` プロパティへ指定可能です。

AH XSL Formatter では XSL 1.1 仕様の `overflow` プロパティを拡張しています。

Property Name	<code>overflow</code>
Value	<code>visible</code>   <code>hidden</code>   <code>scroll</code>   <code>error-if-overflow</code>   <code>repeat</code>   <code>replace</code>   <code>condense</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	all block-level formatting objects
Inherited	yes

ほかの処理方法に加え、はみ出したテキストの置換 (`replace`) と文字の詰め込み (`condense`) が指定可能です。

`scroll` には対応していません。

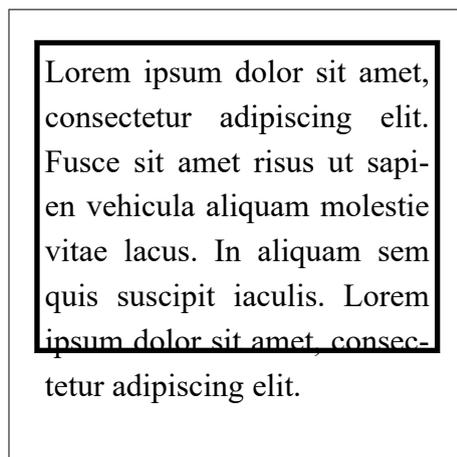


図 2.15-1 ブロック進行方向のはみ出し (`overflow="visible"`)

## 2.15.2 はみ出しのエラー報告を制御する

はみ出し処理でのエラーの条件や処理を制御します。

### overflow="auto"でのエラーを報告する

はみ出しの発生時にエラー報告を行うには `overflow="error-if-overflow"` を明示的に指定すれば可能です。しかし、はみ出しが生じる可能性がある箇所すべてに指定を行うのが難しいときもあるでしょう。

Option Property Name	<code>error-if-overflow</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定 `error-if-overflow` プロパティへ `true` を指定すると、`overflow="auto"` のブロックではみ出しが生じたときエラー報告を行います。

### はみ出し処理の許容値を指定する

はみ出しが生じたとき、`overflow` プロパティへ指定した処理によって、フォントサイズを小さくしたり長体にするなどして、ある程度のテキストを範囲内に収めることが可能です。

しかし、詰め込み処理などを過剰に行うと可読性に影響するほどにレイアウトが崩れてしまうことがあるでしょう。「表示上無理があるときはエラーを表示してほしい」とき、エラー報告の対象とする、はみ出しの制限値を設定可能です。

Property Name	<code>axf:overflow-limit</code>
Value	<code>&lt;length&gt;{1,2}</code>
Initial	システムの既定値に依存
Applies to	<code>block formatting elements</code>
Inherited	<code>yes</code>

Property Name	<code>axf:overflow-limit-block</code>
Value	<code>&lt;length&gt;</code>
Initial	システムの既定値に依存
Applies to	<code>block formatting elements</code>
Inherited	<code>yes</code>

Property Name	axf:overflow-limit-inline
Value	<length>
Initial	システムの既定値に依存
Applies to	block formatting elements
Inherited	yes
Option Property Name	overflow-limit-block
Value	<length>
Initial	0pt
Applies to	<formatter-settings>
Option Property Name	overflow-limit-inline
Value	<length>
Initial	0pt
Applies to	<formatter-settings>

axf:overflow-limit-block プロパティの指定でブロック進行方向の制限値、axf:overflow-limit-inline の指定でインライン方向の制限値を制御可能です。

axf:overflow-limit プロパティはショートハンドです。値がひとつだけであればブロック進行方向とインライン進行方向で共通、ふたつであれば最初の値がインライン進行方向、次の値がブロック進行方向制限値の指定になります。

システム既定値はオプション設定ブロック進行方向、インライン進行方向をそれぞれ overflow-limit-block、overflow-limit-inline プロパティへ指定可能です。

overflow="error-if-overflow"のときの組版エラー出力で、これら\*overflow-limit\*で指定された値を超えたものはエラーレベル2、通常のはみ出し処理を行ったものはエラーレベル1を出力します。

エラー出力についての詳細はマニュアルをご覧ください。

### 2.15.3 はみ出しが生じたブロックをテキストで置換する

ブロック進行方向ではみ出しが生じたとき、ブロックの内容をテキストで置換します。

overflow="replace"を指定したとき、置換されるテキストは拡張プロパティへ記述します。

Property Name	axf:overflow-replace
Value	<string>
Initial	empty string
Applies to	all block-level formatting objects
Inherited	yes

axf:overflow-replace プロパティへ置換されるテキストを記述します。指定したブロックでブロック進行方向のはみ出しが起こるとき、ブロック内が指定したテキストの繰り返しに置換されます。

指定可能であるのは単純なテキストのみです。

## 注

テキストの末尾のみを変更をしたいときは `overflow="hidden"` として `axf:text-overflow` の使用を検討してください。

## 2.15.4 はみ出しを詰め込み表示する

はみ出しが生じるとき、詰め込み処理の方法や行う限度、最終行の揃えを指定します。

`overflow="condense"` のときテキストは詰め込み処理されます。

Property Name	axf:overflow-condense
Value	[ font-size   font-stretch   line-height   letter-spacing ]+   auto   none
Initial	auto
Applies to	all block-level formatting objects, inline-level formatting objects
Inherited	yes

詰め込み処理の方法を `axf:overflow-condense` プロパティで指定可能です [図 2.15-2] (p.190)。

axf:overflow-condense	意味	インライン進行方向での違い
font-size	フォントサイズを調整して詰め込み。	-
font-stretch	フォント幅を調整して詰め込み。	-
line-height	行の高さを調整して詰め込み。	無効
letter-spacing	文字間を調整して詰め込み。文字の重なりや表示位置の逆転を考慮しない。	-
auto	font-size。	font-stretch
none	詰め込み除外を指定する。インライン要素にも指定可。この指定は継承されない。	-

詰め込み処理方法を複数記述したとき、順に試行されます。ある詰め込み処理で、`axf:condense-limit-*` プロパティで指定される限界値を超えても領域に収まらないとき、その状態から続けて次に記述された詰め込み処理方法を試行します。

たとえば `axf:overflow-condense="line-height font-stretch font-size"` として、`axf:condense-limit-line-height="1.1"` とすると、まずブロックの行の高さを縮めて領域内に収めようとしています

が、1.1 よりは縮まらず、`line-height="1.1"`の状態からフォントの幅を調整しての詰め込みを試行します。

`axf:overflow-condense` プロパティの値は継承されます。

一部だけこの指定を無効にするとき、`none` を指定することで詰め込み除外を指定可能です。詰め込み除外はインライン要素への指定が可能で、インライン要素への除外指定は継承されません。

詰め込みを指定したインラインコンテナ、ブロックコンテナ内部で絶対値指定のプロパティがあるとき、詰め込み処理が行われても、そのサイズ指定自体は変更されません。

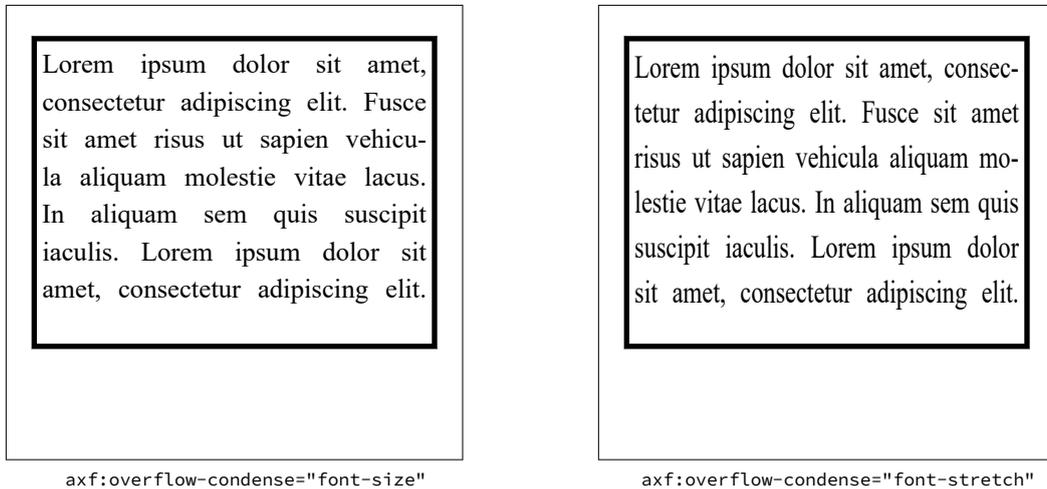


図 2.15-2 `overflow="condense"`

## 2.15.5 詰め込み処理方法ごとに詰め込みの限界値を指定する

はみ出しに対し詰め込み処理を行うとき、詰め込み処理の限界値とフォールバックを指定します。

Property Name	<code>axf:overflow-condense-limit-font-size</code>
Value	<code>none</code>   [ <code>&lt;length&gt;</code> [ <code>visible</code>   <code>hidden</code>   <code>error-if-overflow</code>   <code>repeat</code> ]?]
Initial	<code>none</code>
Applies to	all block-level formatting objects
Inherited	<code>yes</code>

`axf:overflow-condense="font-size"` のとき、その詰め込み処理でのフォントサイズの下限を `axf:overflow-condense-limit-font-size` へ指定可能です。

0 以下の数値は無効です。

`none` は 0 を指定したとみなされます。

Property Name	axf:overflow-condense-limit-font-stretch
Value	none   [[ <length>   <percentage>][ visible   hidden   error-if-overflow   repeat]?)
Initial	none
Applies to	all block-level formatting objects
Inherited	yes
Percentages	refer to the font size

axf:overflow-condense="font-stretch" のとき、その詰め込み処理のフォント幅の下限を axf:overflow-condense-limit-font-stretch プロパティへ指定可能です。

<number>の指定は、その値に 100%を掛けた値とみなされます。

Property Name	axf:overflow-condense-limit-line-height
Value	none   [ <number>[ visible   hidden   error-if-overflow   repeat]?)
Initial	none
Applies to	all block-level formatting objects
Inherited	yes

overflow-condense="line-height" のとき、その詰め込み処理により狭める行の高さの下限を axf:overflow-condense-limit-line-height プロパティへ指定可能です。

<number>はフォントサイズに対する割合です。0 未満を指定したときは 0 とみなされます。

Property Name	axf:overflow-condense-limit-letter-spacing
Value	none   [ <length>[ visible   hidden   error-if-overflow   repeat]?)
Initial	none
Applies to	all block-level formatting objects
Inherited	yes

axf:overflow-condense="letter-spacing" のとき、その詰め込み処理による文字間のアキの下限を axf:overflow-condense-limit-letter-spacing プロパティへ指定可能です。

<length>は詰め量の指定なので、通常負の値を指定します。none のとき、-1em が指定されたときとみなされます。

また、axf:overflow-condense-limit-\*プロパティへ指定したそれぞれの下限を割るときのフォールバックの処理を、下限と同時に指定可能です。overflow プロパティに指定する制御のうち visible、hidden、error-if-overflow、repeat から指定可能です。axf:overflow-condense プロパティに次の処理方法が指定してあるとき、無視されます。

## 2.15.6 詰め込み処理後の最終行の揃えを制御する

詰め込み処理が発生するときの最終行の揃えを指定します。

ブロック最終行の揃えは `text-align-last` プロパティへ指定した値に従います。

詰め込み処理が生じるとき、最終行でテキストが占める割合は大きくなります。このとき、最終行の揃えを均等揃えに変更するよう制御が可能です。

Property Name	<code>axf:condensed-text-align-last</code>
Value	<code>false</code>   <code>true</code>   <code>justify</code>   <code>auto</code>
Initial	<code>false</code>
Applies to	all block-level formatting objects
Inherited	yes
Option Property Name	<code>condensed-text-align-last</code>
Value	<code>false</code>   <code>true</code>   <code>justify</code>
Initial	<code>false</code>
Applies to	<formatter-settings>

詰め込み処理が行われたとき、`text-align-last="justify"`を行うかどうかを、`axf:condensed-text-align-last` プロパティへ指定可能です。

`false` では何もせず、`true` では `text-align-last="justify"` にします。

`justify` を指定すると、`overflow="condense"` を指定したブロックで `text-align="justify"` となっているとき `text-align-last="justify"` とします。

`axf:condensed-text-align-last="auto"` のとき、オプション設定 `condensed-text-align-last` プロパティの値で処理されます。

```
<fo:block overflow="condense" text-align="left" text-align-last="left"
  <fo:block axf:condensed-text-align-last="true">
    ... <!-- 詰め込み処理時、text-align-last="justify"-->
  </fo:block>
  <fo:block axf:condensed-text-align-last="justify">
    ... <!-- 詰め込み処理時、text-align-last="left" -->
  </fo:block>
</fo:block>
<fo:block overflow="condense" text-align="justify" text-align-last="left"
  <fo:block axf:condensed-text-align-last="justify">
    ... <!-- 詰め込み処理時、text-align-last="justify" -->
  </fo:block>
</fo:block>
```

## 注

詰め込み処理により `text-align-last="justify"` となる時、`axf:flush-zone` プロパティなど、関連するほかの仕様も表示結果に影響します。

## 2.15.7 インライン進行方向のはみ出し処理を制御する

インライン進行方向のはみ出しは、折り返さないよう指定されたテキストが、指定された範囲をはみ出すようなときに発生します [図 2.15-3] (p.193)。

Property Name	<code>axf:text-overflow</code>
Value	<code>clip</code>   <code>ellipsis</code>   <code>&lt;string&gt;</code>
Initial	<code>clip</code>
Applies to	block level elements

`axf:text-overflow` プロパティへ、はみ出しが発生するときのテキスト末尾の表示方法を指定します [図 2.15-4] (p.194)。`overflow` プロパティが `hidden` のときのみ有効です。

`axf:text-overflowwrap-option="no-wrap"` を指定したオブジェクトのように、折り返しのできない箇所指定します。

<code>axf:text-overflow</code>	意味
<code>clip</code>	何もしない。 <code>overflow="hidden"</code> の通り、はみ出したテキストが表示から消去される。
<code>ellipsis</code>	HORIZONTAL ELLIPSIS 「...」 (U+2026) 分表示領域を削り、HORIZONTAL ELLIPSIS を末尾に挿入する。最初の文字や画像ではみ出してしまうときは <code>clip</code> と同じ動作。
<code>&lt;string&gt;</code>	指定したテキストが入るだけ表示領域を狭め、末尾にテキストを挿入する。

最初の 1 字または画像ではみ出してしまふときは `clip` として扱われるため、`overflow="hidden"` として処理されます。

## 注

`overflow="condense"` のとき、折り返しが禁止されているのであれば、インライン進行方向のはみ出しに対しても詰め込み処理が適用されます。

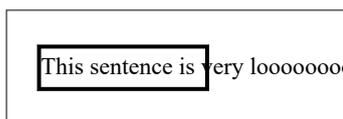


図 2.15-3 インライン進行方向のはみ出し

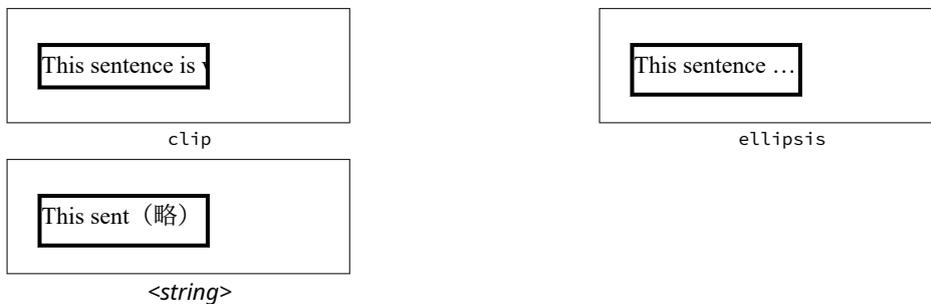


図 2.15-4 axf:text-overflow

## 2.15.8 はみ出したブロックの揃えを指定する

はみ出したブロックの揃えを変更します。

Property Name	axf:overflow-align
Value	normal   start   end   center
Initial	normal
Applies to	block-level formatting objects

はみ出したブロックの位置揃えを `axf:overflow-align` プロパティへ指定します。

`normal` は `text-align` の設定に合わせてます。`overflow="condense"` が指定されているときは無効です。

## 2.15.9 インラインコンテナがはみ出したときの揃えを指定する

インラインコンテナ内のブロックがはみ出すとき、ブロックの位置調整を指定します。

インラインコンテナの含まれる領域が行分割対象となるとき、その子のブロックがはみ出してしまふことがあります。

Property Name	axf:overflow-inline-align
Value	normal   start   end   left   right
Initial	normal
Applies to	
Inherited	yes

インラインコンテナで行分割が生じるとき、その子のブロックの位置を `axf:overflow-inline-align` プロパティへ指定し調整が可能です。

## 2.15.10 keep-together の上限を指定する

分割を `keep-together` により抑制するときの上限を指定します。

ブロックに対し `keep-together` へ条件を指定すると、条件に合致するとき分割が抑制されます。抑制された結果、表示領域外をはみ出すことも起こり得ます。AH XSL Formatter では、この分割抑制について上限値を指定可能です。

## ページ、段分割での `keep-together` の上限を指定する

`keep-together.within-page` や `keep-together.within-column` へ `auto` 以外を指定したとき、フッターやページの下端を超過してしまうことがあるでしょう。

Property Name	<code>axf:keep-together-within-dimension</code>
Value	<code>all</code>   <code>&lt;length&gt;</code>
Initial	<code>all</code>
Applies to	block level formatting object

`axf:keep-together-within-dimension` プロパティに、分割せず配置する上限を指定することで、`keep-together="always"`であっても、上限をはみ出す長さでは `keep-together="auto"`になるように制御可能です。

## 行分割での `keep-together` の上限を指定する

`keep-together.within-line` へ `auto` 以外を指定したとき、行分割を抑制した結果表示領域をはみ出すことがあります。

Property Name	<code>axf:keep-together-within-inline-dimension</code>
Value	<code>all</code>   <code>&lt;length&gt;</code>
Initial	<code>all</code>
Applies to	block level formatting object

`keep-together.within-line` に `auto` 以外を指定したとき、分割を抑制する上限を `axf:keep-together-within-inline-dimension` プロパティへ指定可能です。

`all` のとき、上限はありません。

## 2.16 節 文字間のアキ、揃え制御

日本語や欧文など、スクリプトごとに異なるアキを詳細に制御します。

### 2.16.1 欧文で約物とのアキを制御する

欧文で約物前後のアキを調整します。

欧文組版では、利用する約物が共通でも、言語によってアキを個別に指定したいことがあります。

Element Name	<axf:space-start-punctuation>,<axf:space-end-punctuation>
Property	language, space, code
Parent	fo:declarations

Element Name	<axf:space-between-digit-and-punctuation>
Property	language, space, code
Parent	fo:declarations

Element Name	<axf:space-between-punctuation-and-digit>
Property	language, space, code
Parent	fo:declarations

Option Element Name	<space-start-punctuation>, <space-end-punctuation>
Property	language, space, code
Parent	<formatter-settings>

Option Element Name	<space-between-digit-and-punctuation>
Property	language, space, code
Parent	<formatter-settings>

Option Element Name	<space-between-punctuation-and-digit>
Property	language, space, code
Parent	<formatter-settings>

欧文のテキストで、約物が隣接するときのアキを指定可能です。<axf:space-end-punctuation>へ約物の後ろのアキを、<axf:space-start-punctuation>へ約物の前のアキを、<axf:space-between-digit-and-punctuation>数字の後ろに続く約物とのアキを、<axf:space-between-punctuation-and-digit>へ約物の後ろに続く数字とのアキをそれぞれ指定可能です。

<axf:space-between-\*>への指定が、<axf:space-\*>への指定よりも優先されます。

language へ言語コード、space へアキの量、code へ対象の約物を指定します。

Property Name	space
Value	<number>   <length>   <percentage>   figure   punctuation   thin   hair
Initial	N/A
Applies to	axf:space-end-punctuation, axf:space-start-punctuation, axf:space-between-digit-and-punctuation, axf:space-between-punctuation-and-digit, space-end-punctuation, space-start-punctuation, space-between-digit-and-punctuation, space-between-punctuation-and-digit
Percentages	refer to the font size

アキの量は space プロパティへ指定します。数値や割合による指定のほか、特定の空白文字と同じアキ量を示す文字列が指定可能です。figure は FIGURE SPACE (U+2007)、punctuaion は PUNCTUATION SPACE (U+2008)、thin は THIN SPACE (U+2009)、hair は HAIR SPACE (U+200A) のアキ量となります。

指定した値が負のとき、無効です。

フランス語 (xml:lang="fr") のとき、QUESTION MARK 「?」 (U+003F) のアキを変更する例を次に示します [図 2.16-1] (p.197)。FO に記述した文字やスペースは英語 (xml:lang="en") の箇所と同様ですが、出力したとき前の文字とのアキが異なっています。

```
<!-- フランス語のとき「?」のアキを変更する-->
<fo:declarations>
  <axf:space-start-punctuation code="?" space="1 div 3" language="fr"/>
</fo:declarations>
...
<fo:page-sequence ... xml:lang="en" >
  <fo:flow ...>
    <fo:block>...
      In English: What is a question mark?
      In French:
      <fo:inline font-style="italic" xml:lang="fr">
        Qu'est-ce qu'un point d'interrogation?
      </fo:inline> OK?
    </fo:block>
  </fo:flow>
</fo:page-sequence>
```

... In English: What is a question mark? In French: *Qu'est-ce qu'un point d'interrogation* ? OK?

図 2.16-1 フランス語の疑問符のアキを変更する

## 2.16.2 和欧文間のアキを制御する

テキスト中で和文と欧文が切り換わる箇所にアキを挿入するかどうか、またその幅を指定します。

和文と欧文が混在するテキストで、和文と欧文が隣に並ぶとき、アキを挿入して表示を調整したいことがあります。

**和欧文間** 和文と欧文がテキスト中で、とくにそれぞれの文字が隣り合う配置を取り得るような箇所を指して「和欧文間」と表記します。

Property Name	axf:text-autospace
Value	none   [ ideograph-numeric    ideograph-alpha    ideograph-parenthesis ]   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	text-autospace
Value	true   false
Initial	true
Applies to	<formatter-settings>
Property Name	axf:text-autospace-width
Value	<length>   <percentage>   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Percentages	refer to the font size
Option Property Name	text-autospace-width
Value	<length>   <percentage>
Initial	25%
Applies to	<formatter-settings>
Percentages	refer to the font size

axf:text-autospace	意味
ideograph-numeric	かな漢字と欧文数字間にアキを挿入。
ideograph-alpha	かな漢字と欧文アルファベット間にアキを挿入。
ideograph-parenthesis	かな漢字と欧文括弧間にアキを挿入。かな漢字と欧文閉じ括弧間、欧文開き括弧とかな漢字間にはアキを挿入しない。

axf:text-autospace プロパティで、どの和欧文の文字間にアキを挿入するかを指定し、挿入するアキの幅を axf:text-autospace-width プロパティへ指定します。

「かな漢字」は Unicode の General Category で、Lo (音節や表意文字を含む他の文字) に分類され、符号位置が U+3041 から U+31FF のひらがな、カタカナ、中国語の注音記号、ハングルの字母と、拡張、互換を含む CJK 統合漢字を指します。

「欧文数字」は Unicode の General Category が Nd (10 進数) で、全角幅でないものです。DIGIT ZERO 「0」(U+0030) から DIGIT NINE 「9」(U+0039) および各国語の 10 進数の文字を含みます。

「欧文アルファベット」は Unicode の General Category が Lu (大文字)、Ll (小文字)、Lt (最初が大文字の二重字)、Lm (修飾文字) に分類される文字です。LATIN CAPITAL LETTER A 「A」(U+0041)、LATIN SMALL LETTER A 「a」(U+0061)、LATIN CAPITAL LETTER D WITH SMALL LETTER Z WITH CARON 「Dž」(U+01C5)、LATIN CAPITAL LETTER A WITH GRAVE 「À」(U+00C0) などがあります。

「欧文括弧」は全角幅でない括弧類を指します。LEFT PARENTHESIS 「(」(U+0028)、RIGHT PARENTHESIS 「)」(U+0029)、LEFT CURLY BRACKET 「{」(U+007B)、RIGHT CURLY BRACKET 「}」(U+007D) などがあります。

axf:text-autospace="auto" のとき、オプション設定の text-autospace または GUI の「組版オプション設定ダイアログ」から判断されます。text-autospace="true" のとき、ideograph-numeric ideograph-alpha とみなされます。text-autospace="false" のとき、none とみなされます。

## 2.16.3 全角約物の詰め処理を指定する

和文で、行頭や行末、約物間の詰め処理について指定します。

プロポーショナルでないフォントでは、和文の通常の文字と句読点などの約物の幅は変わりません。AH XSL Formatter では、約物のアキを詰める処理について制御が可能です。

### 行頭、行末、隣接する全角約物の詰め処理を指定する

全角幅の約物のアキ [図 2.16-2] (p.199) を、表示される箇所によって詰める処理を行いたいことがあるでしょう。

行頭で全角開き括弧のアキを詰める(「天ツキ」にする)、行末で全角約物のアキを詰めて行内に収める、全角約物が隣接しアキが広く見える箇所を詰めるといった処理が可能です。



図 2.16-2 全角約物を含むテキスト

Property Name	axf:punctuation-trim
Value	none   [[ start   start-except-first]    [ end   allow-end   end-except-fullstop]    adjacent]   all   <string>   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	punctuation-trim
Value	true   false
Initial	true
Applies to	<formatter-settings>

axf:punctuation-trim プロパティでは和文での全角約物の詰め処理について指定可能です。プロポーショナルフォントでは無効です。

start\*のつく値が行頭の全角約物についての詰め処理 [図 2.16-3] (p.201)、\*end\*のつく値が行末の全角約物についての詰め処理の指定 [図 2.16-4] (p.201) です。

adjacent では、全角約物が隣接するとき、その詰め処理を制御可能です [図 2.16-5] (p.201)。

none では全角約物の詰め処理は行われません。

axf:punctuation-trim	意味
start	行頭で全角約物を半角に扱う。
start-except-first	先頭行、強制改行直後の行を除く、行頭で全角約物を半角に扱う。
end	行揃えが right または justify のとき、行末に位置する全角約物を半角に扱う。
allow-end	行揃えが right または justify のとき、行末に位置する全角約物が行に収まるときは全角、収まらないときは半角に扱う。
end-except-fullstop	行揃えが right または justify のとき、行末に位置する IDEOGRAPHIC FULL STOP 「。」 (U+3002) と FULLWIDTH FULL STOP 「。」 (U+FF0E) を除く全角約物を半角に扱う。
adjacent	和文の全角約物と全角文字間を詰め処理する。
all	全角の括弧類、中点類、句読点類をすべて半角に扱う。
<string>	all で半角に扱われる文字のうち、指定したテキストの文字のみを半角に扱う。
auto	オプション設定 punctuation-trim="false" のとき none、punctuation-trim="true" のとき start end adjacent とみなす。

オプション設定 punctuation-trim プロパティの指定は axf:text-justify-trim="auto" にも影響します。

axf:punctuation-trim="adjacent" のとき、隣接する全角約物のアキを詰める量は axf:punctuation-spacing プロパティへ指定可能です。

|(行頭の括弧詰め)|

図 2.16-3 行頭の全角約物を詰める例

| (行末の括弧詰め)|

図 2.16-4 行末の全角約物を詰める例

|だ。(ところで、話は|

図 2.16-5 隣接する全角約物間のアキを詰める例

## 隣接する全角約物の詰め幅を指定する

`axf:punctuation-trim="adjacent"` のとき、隣接する全角約物のアキを詰める量を指定可能です。

Property Name	<code>axf:punctuation-spacing</code>
Value	<code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Percentages	refer to the font size
Option Property Name	<code>punctuation-spacing</code>
Value	<code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>
Initial	<code>50%</code>
Applies to	<code>&lt;formatter-settings&gt;</code>
Percentages	refer to the font size

隣接する全角約物のアキを詰める量を `axf:punctuation-spacing` プロパティへ指定します。

`axf:punctuation-spacing="auto"` のときのアキ量を、オプション設定 `punctuation-spacing` プロパティへ指定可能です。

## 2.16.4 追い込みを制御する

追い込み処理を制御します。

テキストが想定した範囲に収まらないとき、字間を詰めて範囲内に収めるようにする追い込み処理を行うことがあります。このとき、処理対象の変更が可能です。

Property Name	axf:text-justify-trim
Value	none   [ punctuation    punctuation-except-fullstop    punctuation-except-middledot    [ kana   ideograph]    inter-word]   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	text-justify-mode
Value	4   5
Initial	5
Applies to	<formatter-settings>

axf:text-justify-trim プロパティによって、追い込み処理のとき、追い込み対象を変更します。axf:punctuation-trim プロパティは全角約物に対して出現位置を限定した詰め処理を指定可能ですが、axf:text-justify-trim プロパティはより広範なテキストに影響します。

axf:text-justify-trim="auto" のとき、AH XSL Formatter に処理を任せます。オプション設定 text-justify-mode プロパティが 4 のとき、または axf:avoid-widow-words プロパティが有効であるとき、ideograph inter-word として処理されます。オプション設定 punctuation-trim プロパティの指定によっても処理が変更されます。

axf:text-justify-trim	意味
punctuation	和文で全角の括弧類、中点類、句読点類を詰め処理の対象とする。
punctuation-except-fullstop	和文で IDEOGRAPHIC FULL STOP 「。」(U+3002)、FULLWIDTH FULL STOP 「.」(U+FF0E) を除く全角の括弧類、中点類、読点類を詰め処理の対象とする。
punctuation-except-middledot	和文で KATAKANA MIDDLE DOT 「・」(U+30FB)、FULLWIDTH COLON 「:」(U+FF1A)、FULLWIDTH SEMICOLON 「;」(U+FF1B) を除く全角の括弧類・読点類を詰め処理の対象とする。
kana	ひらがな、カタカナを詰め処理対象とする。
ideograph	漢字、ひらがな、カタカナなどを詰め処理対象とする。
inter-word	欧文単語間のアキを詰め処理対象とする。

## 2.16.5 letter-spacing によるアキの端を揃える

文字間にアキを指定するとき、指定した語の端をどう揃えるかを指定します。

Property Name	axf:letter-spacing-side
Value	both   start   end
Initial	both
Applies to	inline-level formatting objects
Inherited	yes

letter-spacing プロパティを指定したとき、端をどのように配置するかを、axf:letter-spacing-side プロパティで指定可能です。

both では文字の両側に半分ずつ、start では文字の start 側、end では文字の end 側にアキが配置されます。

## 2.16.6 文字数に合わせて CJK 文字の字取りを制御する

文字間のアキを、文字数ごとに制御します。

見出しなど、ある程度決まった範囲内でテキストを置くときに、字間を空けて配置したいとき「2 文字ならこの程度空けて、3 文字ならこの程度、4 文字以上ならとくに字間は空けない」という処理を、版面計算をした上で登場箇所ごとに指定するのは大変な負担になるでしょう。

Property Name	axf:auto-letter-spacing
Value	[ none   <length>   <percentage>]*
Initial	none
Applies to	inline-level formatting objects
Percentages	refer to the font size

axf:auto-letter-spacing プロパティでは、CJK の 2 字以上のテキストに対し、字数ごとに字間のアキを指定可能です。

### 注

auto-letter-spacing プロパティへ値を指定した要素の端のアキは、ブロックでは text-align プロパティや text-align-last プロパティ、インラインでは axf:letter-spacing-side プロパティなどの影響を受けます。

## 2.16.7 均等割り付けの処理を制御する

均等割り付けのときの、割り付け処理を行う単位を制御します。

text-align プロパティなどで justify を指定するとき、その方法を細かく制御可能です。

Property Name	axf:text-justify
Value	auto   inter-word   inter-character   distribute
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes

justify で均等割り付けを行うとき、アキを調整する箇所を axf:text-justify プロパティへ指定します。

axf:text-justify	意味
inter-word	欧文などの一般的な処理。単語間のアキを調整する。
inter-character, distribute	distribute は互換のために残存。和文などの一般的な処理。それぞれの文字間でアキを調整する。
auto	スクリプトによって判断する。漢字、ひらがな、カタカナでは inter-character、アラビア文字では Kashida 処理、そのほかでは inter-word。

axf:text-justify="inter-character"では、letter-spacing プロパティと同様に、カーニングやリガチャ処理が制限されます。

## 2.16.8 Kashida の割合を制御する

アラビア語での Kashida の割合を指定します。

アラビア語ではスペースを空けるとき、通常のアキのほか、Kashida という文字の端から延長される線によって単語間の調整を行います。

Property Name	axf:text-kashida-space
Value	<percentage>   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	text-kashida-space
Value	<percentage>
Initial	100%
Applies to	<formatter-settings>

axf:text-space-kashida プロパティへアキと Kashida の割合を指定可能です。0%から 100%の値で指定します。

0%では Kashida 処理は行われず、100%では可能な限り Kashida を挿入します。範囲を超えた値の入力は無効です。

axf:text-kashida-space="auto"のとき、オプション設定 text-kashida-space プロパティへ指定した値が適用されます。

## 2.16.9 タブ揃えを指定する

タブによる揃えを指定します。

XSL 1.1 仕様では TAB (U+0009) は単なるスペースとして扱われ、ひとつにまとめられるか、xml:space="preserve"の指定でほかのスペースとともに残されるかという処理になります。

AH XSL Formatter では、位置揃えにタブ構造を利用可能です。

### ⚠ 注意事項

<axf:tab>が含まれる行では text-align="start"に強制されます。

BIDI 処理を要するテキストでは、想定したようにタブ揃えが機能しないことがあります。

Element Name	<axf:tab>
Property	axf:tab-align
Parent	fo:block, fo:inline
Property Name	axf:tab-align
Value	auto   start   end   left   right   decimal   <string>
Initial	auto
Applies to	axf:tab
Property Name	axf:tab-stops
Value	[[ <tab-align>?[ <length>   <number>]]*[ <tab-align>? eol]?!
Initial	8
Applies to	all block-level formatting objects
Inherited	yes

<axf:tab>によるタブ揃えの制御が axf:tab-align プロパティと axf:tab-stops プロパティで指定可能です。

それぞれの<axf:tab>ごとにタブ揃えを指定するのであれば、その<axf:tab>に axf:tab-align プロパティで指定します。<string>を指定したとき、指定した文字列の先頭がタブ位置として揃えられます。decimal を指定したとき、axf:tab-align="."として文字列を指定したとみなされます。

axf:tab-align="auto"のとき、ブロックなどに axf:tab-stops プロパティで複数のタブ揃えの記述をまとめて指定可能です。axf:tab-align で指定可能な値に加え、<length>、<number>、行末に揃える eol が指定可能です。

```

<!-- tab-stopsでタブ間隔をまとめて記述する -->
<fo:block axf:tab-stops="eol">
  <axf:tab tab-align="auto"/>
</fo:block>

```

## タブ揃えで文字が重なるときの処理を変更する

Property Name	axf:tab-overlap-treatment
Value	auto   ignore-tab   next-tab
Initial	auto
Applies to	fo:block, fo:character
Inherited	yes
Option Property Name	tab-overlap-treatment
Value	ignore-tab   next-tab
Initial	ignore-tab
Applies to	<formatter-settings>

タブ揃えでタブと文字が重なるときの処理を `axf:tab-overlap-treatment` プロパティで指定します。 `ignore-tab` ではタブ位置に文字を重ねて表示し、 `next-tab` では次のタブ揃えの位置までテキストの配置を送ります。

`tab-overlap-treatment="auto"` のときの処理方法をオプション設定 `tab-overlap-treatment` プロパティへ指定します。

### 2.16.10 TAB (U+0009) の処理を変更する

テキスト中の TAB (U+0009) の扱いを指定します。

`<axf:tab>` とは別に TAB を FO 内で記述しているとき、タブ揃えの対象として扱うか、空白文字として扱うかを指定可能です。

Property Name	axf:tab-treatment
Value	<number>   preserve
Initial	4
Applies to	fo:block, fo:character
Inherited	yes

TAB の扱いを `axf:tab-treatment` プロパティへ指定します。

`preserve` を指定したとき、TAB を `<axf:tab>` として扱います。このとき、TAB は `axf:tab-stops` の処理対象となります。

<number>を指定したとき、その文字数の SPACE (U+0020) として扱います。このとき、TAB は white-space-treatment や white-space-collapse プロパティの処理対象となります。

## 2.16.11 固定幅空白字形の処理を変更する

固定幅字形の扱いを指定します。

多くのフォントでは、EM SPACE (U+2003) や THIN SPACE (U+2009) など、固定幅空白のための字形は用意されていません。これらの空白がテキスト中に挿入されているとき、フォールバックで代替字形のものが表示されると正しいアキが取られないことがあります。

Option Property Name	fixed-width-space-treatment
Value	false   true   always
Initial	true
Applies to	<formatter-settings>

オプション設定 fixed-width-space-treatment プロパティでは、固定幅空白の扱いを指定可能です。制御可能な文字を次に示します。いずれも em 単位です。

Character	em
EN QUAD (U+2000)	$\frac{1}{2}$
EM QUAD (U+2001)	1
EN SPACE (U+2002)	$\frac{1}{2}$
EM SPACE (U+2003)	1
THREE-PER-EM SPACE (U+2004)	$\frac{1}{3}$
FOUR-PER-EM SPACE (U+2005)	$\frac{1}{4}$
SIX-PER-EM SPACE (U+2006)	$\frac{1}{6}$
FIGURE SPACE (U+2007)	DIGIT ZERO 「0」 (U+0030) の幅
PUNCTUATION SPACE (U+2008)	FULLSTOP 「.」 (U+002E) の幅
THIN SPACE (U+2009)	オプション設定 thin-space-width の幅
HAIR SPACE (U+200A)	オプション設定 hair-space-width の幅
MEDIUM MATHEMATICAL SPACE (U+205F)	$\frac{4}{18}$

false のとき、何もしないため代替字形の空白が表示されます。

true のとき、フォントに対象の字形がないときアキを挿入します。フォントに字形が存在すればそちらを優先します。

`always` のとき、フォントに対象の字形があっても常にアキを挿入します。

Option Property Name	<code>thin-space-width</code>
Value	<code>&lt;number&gt;</code>
Initial	<code>0.2</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

Option Property Name	<code>hair-space-width</code>
Value	<code>&lt;number&gt;</code>
Initial	<code>0.1</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

THIN SPACE と HAIR SPACE のアキはそれぞれオプション設定 `thin-space-width`、`hair-space-width` プロパティで指定可能です。

`fixed-width-space-treatment` プロパティへ `true` または `always` に指定したとき、これらのアキが取られます。

## 2.16.12 NO-BREAK SPACE (U+00A0) をアキ調整から除外する

アキ調整対象から NO-BREAK SPACE (U+00A0) を除外します。

NO-BREAK SPACE は挿入された位置での行分割が行われない、スペースを含む単語などで使用される文字です。AH XSL Formatter で `text-align="justify"` など均等割り付けを指定したとき、欧文では単語間のスペースを調整します。このとき調整対象のスペースに NO-BREAK SPACE を含むかを制御します。

Property Name	<code>axf:justify-nbsp</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>all formatting objects</code>
Inherited	<code>yes</code>

`axf:justify-nbsp` プロパティが `true` のとき、均等割り付けの対象とします。`false` では対象外とするため、NO-BREAK SPACE はスペースの幅以上にはアキが取られません。

## 2.17 節 文字、書式

フォント機能によらない、書式の制御や正規化、文字の変換や組方向の変更を制御します。

### 2.17.1 テキストの書式を制御する

`format` プロパティで指定可能なトークンが拡張されています。指定できるトークンについてはマニュアルをご覧ください。

指定できるトークンにない文字列を指定すると、文字列を繰り返します。文字の種類によって、表現の規則が変わります。

#### 数字書式の拡張と変換を制御する

指定する書式によって表現可能な数値が異なります。

Property Name	<code>axf:number-transform</code>
Value	<code>none</code>   <code>kansuji</code>   <code>kansuji-if-vertical</code>   <code>&lt;list-style-type&gt;</code>   <code>&lt;string&gt;</code>
Initial	<code>none</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes

`axf:number-transform` プロパティを指定したブロック、インライン要素では、そのブロック内の数字列が指定した書式へ変換されます。

`none` では変換は行われません。

`kansuji` を指定したとき、数字列は漢数字に変換されます。`axf:kansuji-*` プロパティによって詳細を指定します。`kansuji-if-vertical` では縦組のときのみ `kansuji` を適用します。

`<list-style-type>` は `axf:counter-style` プロパティで定義されたスタイル、AH XSL Formatter での定義済みカウンタスタイル、`symbols()` による指定です。`<string>` は `format` プロパティでの指定と同様です。

#### 漢数字の変換、位取りを指定する

漢数字の表記方法は複数あります。数字列を一対一対応する漢数字に変換する一〇方式の変換、桁数を示す「十」「百」「千」といった文字を使用する十方式の変換と、万など大きな桁数のとき、その桁を示す文字を途中で表記する位取りです。AH XSL Formatter の位取りでは、「万」「億」「兆」「京」「垓」「杼」「穰」「溝」「澗」「正」「載」「極」が順に 5 桁ごとに挿入されます。

`axf:number-transform="kansuji"` のときの漢数字の表記方法の詳細を、拡張仕様により制御可能です。

Property Name	axf:kansuji-style
Value	simple   grouping   readable
Initial	simple
Applies to	all block-level and inline-level formatting objects
Inherited	yes

axf:kansuji-style プロパティへ、漢数字の表記方法を指定します。  
simple では一〇方式の変換を行います。

grouping では一〇方式の変換に加え、位取りが挿入されます。また、ある位取りより下がすべてゼロのとき、表記から捨てられます。

readable では十方式の変換が行われます。位取りの挿入に加え 2 桁が「十」、3 桁が「百」、4 桁が「千」として、可読な漢数字列に変換されます。「極」の次の位取りが必要な巨大な数値では、極より上の桁は一〇方式で変換されます。

axf:kansuji-style	12340 の変換	6700000 の変換
simple	一二三四〇	六七〇〇〇〇〇
grouping	一万二三四〇	六七〇万
readable	一万二千三百四十	六十七万

## 漢数字に使用する文字を指定する

Property Name	axf:kansuji-letter
Value	kansuji   latin   <string>
Initial	kanji
Applies to	all block-level and inline-level formatting objects
Inherited	yes

axf:kansuji-letter プロパティでは axf:number-transform="kansuji" のとき、漢数字として利用する文字を指定可能です。

axf:kansuji-letter="<string>" に指定した文字は順に 0 から 9 の数字として割り当てられます。10 文字に満たないとき、その文字以降は「〇一二三四五六七八九」の該当位置以降の文字が割り当てられます。

kanji を指定すると漢数字を使用します。axf:kansuji-letter="〇一二三四五六七八九" と等価です。

latin を指定すると全角洋数字を使用します。axf:kansuji-letter="0 1 2 3 4 5 6 7 8 9" と等価です。

たとえば大字を使用するのであれば、axf:kansuji-letter="零壹貳參肆伍陸柒捌玖" と指定します。

## 漢数字の位取りに使用する文字を指定する

Property Name	axf:kansuji-grouping-letter
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects
Inherited	yes

axf:number-transform="kansuji"かつ axf:kansuji-style が grouping または readable のときに利用される位取りの文字を、axf:kansuji-grouping-letter プロパティへ指定します。

指定方法は axf:kansuji-letter="<string>"と同様です。

空文字列のとき、axf:kansuji-grouping-letter="十百千万億兆京垓杼穰溝澗正載極"とみなされます。

## 2.17.2 独自カウンタを定義する

CSS3 Counter Styles Level 3 にあるような、カウンタの定義が可能です。

Element Name	<axf:counter-style>
Property	name, system, negative, prefix, suffix, range, pad, fallback, symbols, additive-symbols
Parent	fo:declarations

<axf:counter-style>要素は<fo:declarations>の子要素として記述します。FO 内でカウンタの指定をするときに用いるカウンタ名を name プロパティに、カウント方法を system プロパティに指定します。既存のカウンタを拡張、変更するには system="extends <カウンタスタイル名>"のように記述します。

負値のときの前置、後置文字列を negative プロパティへ指定します。ひとつめの<symbol>が前置、ふたつめを指定すると後置文字列となります。

前置文字列を prefix、後置文字列を suffix プロパティへ指定します。

パディングを行う桁数とそのときに用いるシンボルを pad へ指定します。

range にスタイルが適用可能な範囲を指定し、範囲外やほかの理由でスタイルが適用できないときのフォールバックを fallback へ指定します。

カウンタにシンボルを利用するのであれば symbols、重み付きシンボルカウンタのとき additive-symbols プロパティへ指定します。

name プロパティには default、decimal、none は指定できません。

詳細についてはマニュアルをご覧ください。

**重み付きシンボルカウンタ**      たとえばローマ数字のように表すものです。1 から 3 までは記号 I がその数だけ加算され、4 では記号 IV に、そして 5 では記号 V といったように変更されていきます。

## 注

定義したカウンタを `axf:number-transform` へ書式として指定するとき、`prefix` や `suffix` は無視されます。

### 2.17.3 文字の正規化と除外指定を行う

Unicode の文字正規化と除外指定を制御します。

Unicode 正規化は合成 (Composition)、分解 (Decomposition)、互換分解して合成 (Compatibility decomposition Composition)、互換分解 (Compatibility Decomposition) の 4 種類で規定されています。NFC は合成 (C) ですが、実際は分解をしてから合成を行います。

この問題点の一部は合成除外文字についての説明で補足します。

Property Name	<code>axf:normalize</code>
Value	<code>auto</code>   <code>none</code>   <code>nfc</code>   <code>nfkc</code>   <code>nfd</code>   <code>nfkd</code>
Initial	<code>auto</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes

Property Name	<code>normalize</code>
Value	<code>none</code>   <code>nfc</code>   <code>nfkc</code>   <code>nfd</code>   <code>nfkd</code>
Initial	<code>nfc</code>
Applies to	all block-level and inline-level formatting objects
Inherited	yes

Property Name	<code>axf:normalize-exclude</code>
Value	<code>full-composition-exclusion</code>   <code>none</code>
Initial	<code>full-composition-exclusion</code>
Applies to	all block-level and inline-level formatting objects

`axf:normalize` プロパティへ、Unicode 正規化の制御を指定可能です。

`axf:normalize="auto"` のとき採用する正規化の方法をオプション設定 `normalize` プロパティへ指定します。

`axf:normalize-exclude="full-composition-exclusion"` を指定すると、合成除外文字を正規合成対象から除外します。

## 注

文字列を正準分解し、それを合成するとき、元の文字列に戻らないものを合成除外文字といい、UAX #15 に Composition Exclusion Table として示されています。

**script-specific** 正準分解可能だが、通常そのスクリプトでは合成形を使わない文字。

**post composition version** Unicode 3.0 よりも後に追加された正準分解可能な文字。

**Singleton** 分解結果が被る文字があり、分解後に合成しても戻らない文字。

**Non-Starter Decompositions** COMBINING GREEK DIALYTIKA TONOS (U+0344) のように、分解を開始する基点にならない文字。

---

CJK 互換漢字の補助漢字 (U+2F800~U+2FA1D) のすべてが Singleton に該当します。

## 2.17.4 大文字小文字、全角文字半角文字を変換する

大文字と小文字間や、全角文字と半角文字間で文字コードを変換します。

Property Name	text-transform
Value	none   [[ capitalize   uppercase   lowercase   capitalize-lowercase ]   [ fullwidth   fullwidth-if-vertical ]   fullsize-kana]
Initial	none
Applies to	all formatting objects
Inherited	yes

text-transform	意味
capitalize-lowercase	欧文で頭文字を大文字、続く字を小文字に変換する。大文字で構成された語も「大文字+小文字」に変換。
fullwidth	U+0021 から U+007E の文字 (スペース以外の表示可能な ASCII 文字) を全角に変換する。
fullwidth-if-vertical	writing-mode="tb-rl" など、縦組のときにのみ text-transform="fullwidth" を適用する。
fullsize-kana	HIRAGANA LETTER SMALL TU 「っ」 (U+3064) や HIRAGANA LETTER SMALL YU 「ゅ」 (U+3085) などの小書きのかなを HIRAGANA LETTER TU 「つ」 (U+3065) や HIRAGANA LETTER YU 「ゆ」 (U+3086) などの普通のかなに変換する。全角文字のみ変換。

### 注

text-transform プロパティではコードポイントの異なる文字の変換を行います。同じコードポイントを持つ別の字形を表示するときは font-variant プロパティを利用します。任意の文字について別の文字へ置き換えるときは axf:text-replace プロパティが有効です。

## 2.17.5 テキストを置換する

AH XSL Formatter での処理時にテキストの置換を行います。

Property Name	axf:text-replace
Value	none   [ <string> <string> ]+
Initial	none
Applies to	all block-level and inline-level formatting objects
Inherited	yes

axf:text-replace プロパティへ置換対象のテキストと、置き換えるテキストを組として指定します。組になっていない、置換対象のテキストが空文字列であるときは置換が行われません。

置換対象の判定は単純な比較であるため、改行などを挟むとマッチしません。空白文字は、white-space-treatment プロパティなどで処理された後に置換が行われます。

### 注

指定したテキストの組はマッチした時点で置換されます。また、置換が行われるのは一度です。同じ並びが繰り返されるようなテキストを複数置換対象に指定するときは、より長いテキストを先に指定します。

## 2.17.6 縦組での文字の方向を制御する

縦組などで行われる、正立・縦中横の制御を指定します。

### 文字を正立させる

縦組で和欧混合の文のように複数種類の文字を使うとき、欧文など横組用の字形しかない文字の表示方法はいくつかあります。縦組の全体に対し該当の語の部分を横に倒した形や、1字あるいは数文字ごとに 90 度回転させ縦組に合わせる正立などがあります。

単純な文字の回転に留まらず、前後の文字とのアキを計算して配置が行われます。

Property Name	axf:text-orientation
Value	mixed   upright   sideways-rl   sideways   none   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes

Option Property Name	text-orientation-mode
Value	5   6   7
Initial	6
Applies to	<formatter-settings>

axf:text-orientation プロパティは [CSS3 Writing-Mode] を基にしています。  
 mixed では英数字など欧文を時計回り方向に直角に回転させます。  
 sideways、sideways-rl ではすべての文字を時計回り直角に回転させます。  
 upright では英数字などを和文全角文字と同じ高さとしてそれぞれの文字を正立させます。縦組の文書で文字を正立するとき、どの文字がその対象になるかは UAX#50:Unicode Vertical Text Layout で議論されています。この中の Mixed Vertical Orientation (MVO) に該当の説明があります。AH XSL Formatter はこの MVO に若干の修正を加えたものを実装しています。

none は AH XSL Formatter によって mixed に近い結果が得られます。回転対象の文字は AH XSL Formatter の旧版との互換性を考慮し決定されています。

縦組に対応したフォントには、縦組用の字形を持つものもあります。よく使用される文字としては句読点や括弧類、ひらがな、カタカナの小文字などがあります。upright では、括弧などの約物は回転していません。sideways-\* のとき、これらも回転の対象となります。

axf:text-orientation="auto" のとき、オプション設定 text-orientation-mode によって制御されます。7 のとき mixed、6 以下では none とみなされます。

## 注

正立した文字では axf:word-break="break-all"、hyphenate="false" として処理されます。つまり、どの文字間でも行分割は抑制されず、分割箇所でハイフンは挿入されません。

## 縦中横を制御する

1 字ずつ正立させるのであれば axf:text-orientation による制御で対応可能ですが、「横組数文字を回転させて縦組 1 字分の高さに収めたい」ときは対応できません。

縦組の中で、文字を縦向きのまま横組にすることを「縦中横」と呼びます。XSL 1.1 仕様で縦中横を実現するのであれば、たとえば書字方向を変更したインラインコンテナで実現できます。しかし、文字数による処理の変更などが複雑化するでしょう。

Property Name	axf:text-combine-horizontal
Value	none   all   [ digits <integer>    alpha <integer>    alphanumeric <integer> ]
Initial	none
Applies to	all block-level and inline-level formatting objects
Inherited	yes

axf:text-combine-horizontal プロパティで縦組中での自動縦中横を制御可能です。

axf:text-combine-horizontal	意味
none	自動縦中横を行わない。
all	テキストのみの要素の全体を縦中横にする。
digits <integer>	<integer>で指定された桁数以下の数字の連続を縦中横にする。
alpha <integer>	<integer>で指定された文字数以下の英数字の連続を縦中横にする。
alphanumeric<integer>	<integer>で指定された桁数以下の英数字 (0-9, A-Z, a-z) の連続を縦中横にする。

axf:text-combine-horizontal プロパティは、たとえば文書中で縦中横として出力したいテキストが多く、一括で指定するときなどに有用です。

## 2.17.7 ユースケース：一部の文字を縦組用の文字へ置換する

縦組のとき、括弧類や和文の小文字などは AH XSL Formatter が自動で縦組用の字形に変更します。しかし、縦組用の特殊な文字への置換は対応できません。

テキストの文字の一部を、axf:text-replace プロパティへ指定した縦組用の文字へ置換した例を示します [図 2.17-1] (p.217)。

```
<!-- 横組のテキストの一部の文字を縦組用の文字へ置換する -->
<fo:page-sequence writing-mode="tb-rl">
  <fo:flow ...>
    ...
    <fo:wrapper
      axf:text-replace="離れ離れ 離れ&#x3034;&#x3035;
      トントントン トン&#x3033;&#x3035;&#x3033;&#x3035;
      さまざま さま&#x3034;&#x3035;
      かくかく かく&#x3033;&#x3035;
      しかじか しか&#x3034;&#x3035;">
      <fo:block>離れ離れになる</fo:block>
      <fo:block>トントントン</fo:block>
      <fo:block>さまざまな試み</fo:block>
      <fo:block>かくかくしかじか</fo:block>
    </fo:wrapper>
  </fo:flow>
</fo:page-sequence>
```

前方に記述されたテキストから置換が行われます。「トントントントン」「トントントン」「トントントン」というテキストが登場するとき、置換するテキストの組は、最長の「トントントントン」から順に記述してください。

離れぐゝになる  
トンくくく  
さまぐゝな試み  
かくくしかぐ

図 2.17-1 横組のテキストの一部の文字を縦組用の文字へ置換する

## 2.18 節 フォント

フォント機能を利用した合字処理や字形の変更、言語による処理の変更について扱います。

### 2.18.1 字幅を指定する

字幅指定時の指定が拡張されています。

Property Name	font-stretch
Value	normal   wider   narrower   ultra-condensed   semi-condensed   semi-expanded   expanded   extra-expanded   ultra-expanded   <i>&lt;percentage&gt;</i>   <i>&lt;number&gt;</i>
Initial	normal
Applies to	all formatting objects
Inherited	yes
Percentages	the font size

font-stretch プロパティによる文字の伸縮について、*<percentage>* または *<number>* による指定が可能に拡張されています。

*<number>* での指定は ( $\text{<number>} \times 100\%$ ) として扱われます。たとえば、CJK 全角文字を半分の幅で表示するとき font-stretch="0.5" のように指定します。

OpenType フィーチャによる半角字形の指定を行いたいときは、font-variant プロパティで指定します。

Property Name	font-stretch-mode
Value	5   6
Initial	6
Applies to	font-settings

オプション設定 font-stretch-mode プロパティは、font-stretch プロパティなど、フォントを伸縮する指定を行うとき、コンデンス体など専用のフォントを利用するかを制御します。6 では font-stretch プロパティへ condensed などのキーワードを指定したとき、フォント選択にその情報を利用します。たとえば、extra-condensed が指定されていればコンデンス体フォントを基に圧縮します。

5 ではフォント選択に font-stretch の値は利用されません。

### 2.18.2 疑似的な太字を制御する

フォントファミリに bold を持たないフォントでの太字のとき、太らせる量を指定します。

AH XSL Formatter では、font-weight="bold" などウエイトに太字を指定したとき、フォントファミリに bold のウエイトを持つフォントではそのウエイトが利用されます。bold ウエイトが存在しない

フォントでは、フォントをシステムで決めた量だけ太らせることにより、疑似的な bold 化を行います。

Option Property Name	<code>bold-ratio</code>
Value	<code>&lt;number&gt;</code>
Initial	<code>1.0</code>
Applies to	<code>&lt;font-settings&gt;</code>

疑似的な bold 体としてフォントを太らせる量を、オプション設定 `bold-ratio` プロパティへ指定可能です。指定した `<number>` 倍にフォントを太らせて表示します。1.0 では、疑似的な bold 化処理は行われません。

0.0 以下の値を指定したとき、1.0 とみなされます。

### 2.18.3 既定のフォントサイズ、色を指定する

フォントの指定がされていないときに参照される既定のフォントサイズと色を指定します。

Option Property Name	<code>default-color</code>
Value	<code>&lt;#RRGGBB&gt;</code>
Initial	<code>#000000</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

Option Property Name	<code>default-font-size</code>
Value	<code>&lt;length&gt;</code>
Initial	<code>10pt</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

オプション設定 `default-color` プロパティへ、色の指定が得られないときに採用される色を `<#RRGGBB>` の形式で指定します。色名や `rgb()`、`rgb-icc()` での指定も可能です。`rgb-icc()` による指定では、RGB は省略できません。

オプション設定 `default-font-size` プロパティではフォントサイズが得られないときのフォントサイズを指定します。

### 2.18.4 統合漢字などで採用される既定の CJK 言語を指定する

スクリプトが CJK に対しあいまいなときに採用する言語を指定します。

CJK 統合漢字など、CJK のどれであるかがあいまいなスクリプトがあります。このとき、どの言語を採用するかを指定します。

Option Property Name	default-CJK
Value	<string>
Initial	empty
Applies to	<formatter-settings>

オプション設定 default-CJK プロパティへ CJK いずれかのスクリプトを指定します。無指定のとき、動作環境から決定されます。CJK 以外の動作環境では日本語が仮定されます。

GUI では「組版オプション設定」の「組版」タブの「標準の CJK 言語」で指定可能です。

## 2.18.5 フォントサイズ調整の基準をフォント名で指定する

複数のフォントを混在させてテキストを構成するときなどに、フォントサイズの調整を font-size-adjust プロパティへ指定します。

XSL 1.1 仕様では、変更対象のフォントサイズとの比を <number> として入力可能です。AH XSL Formatter では、フォント名による指定が可能なように拡張されています。

Property Name	font-size-adjust
Value	none   <number>   <string>
Initial	none
Applies to	all formatting objects
Inherited	yes

指定されたフォントの x-height を得て変更対象の x-height を合わせます。指定したフォントが存在しない、フォントに x-height 情報がないときは none とみなされます。

左に Arial、右に Times New Roman で「x」を表示し、font-size-adjust="Times New Roman" を指定した例を示します [図 2.18-1] (p.220)。



図 2.18-1 font-size-adjust

## 2.18.6 カーニング（詰め処理）を指定する

カーニング処理を行うかを制御します。

「V」「A」など、隣り合う文字の形によって、見た目上アキが広く見える文字の組み合わせがあります。ペアカーニングはこうした特定の文字の組み合わせのとき機能する文字詰め処理です。

Property Name	axf:kerning-mode
Value	none   pair   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects
Inherited	yes
Option Property Name	pair-kerning
Value	true   false
Initial	auto
Applies to	<formatter-settings>

axf:kerning-mode	意味
none	カーニング処理をしない。
pair	ペアカーニング処理を行う。
auto	システムの既定値に依存。

axf:kerning-mode="auto"のとき、ペアカーニングを行うかどうかはオプション設定 pair-kerning プロパティで制御可能です。true のときペアカーニングを行います。

全体的な字間の調整は、letter-spacing プロパティで可能です。しかし、letter-spacing プロパティで字間が指定されているとき axf:kerning-mode プロパティへの指定は機能しません。

また、axf:kerning-mode プロパティよりも axf:font-feature-settings プロパティへの kern、vkern の指定が優先されます。

## 2.18.7 字形やフィーチャを指定する

合字処理や字形を指定します。

Property Name	font-variant
Value	normal   none   [ <font-variant-caps>    <font-variant-numeric>    <font-variant-alternates>    <font-variant-ligatures>    <font-variant-alternates>    <font-variant-position>    <font-variant-east-asian> ]
Initial	normal
Applies to	all elements
Inherited	yes

font-variant プロパティについて、AH XSL Formatter では [CSS3-Fonts] Overall shorthand for font rendering: the font-variant property にあるような字体指定が可能に拡張されています。OpenType フォントフィーチャを利用し、オールドスタイルの数字や日本語の旧字体などを指定可能です。

機能を持っていないフォントでは、`small-caps` と `all-small-caps` についてはエミュレートして表示し、ほかの指定のときは無視されます。

`none` を指定したとき、`<font-variant-ligatures>` については `none`、ほかの `font-variant` プロパティで指定可能な値については未指定になります。

## 注

見出しのブロックに `titling-caps` を指定しても、フォントが対応していないときは通常  
の字形で表示されることとなります。

CSS3 仕様では `font-variant` プロパティはショートハンドです。しかし、AH XSL Formatter では個別のプロパティによる指定はありません。`<font-variant-caps>` のように表記上まとめていますが、実際の指定では `font-variant="small-caps"` のように指定します。

`font-variant` プロパティへの指定によって可能な表現は多岐にわたります。各指定の詳細は個別の項目で述べます。

## 2.18.8 欧文のキャピタルを指定する

スモールキャピタルなどを指定します。

スモールキャピタル (`small-caps`) は、多くは小文字の高さを `x-height` に合わせたサイズの字形で、かつ大文字に近い見目を持ちます。たとえば本文で頭文字語に大文字を利用するとき、主要なテキスト (小文字) と高低差を生まないために利用します。

`petite-caps` は通常、`small-caps` よりもさらに低い高さでスモールキャピタル同様に大文字に近い見目の字形です。

通常、大文字の字形は小文字と一緒に使用することを想定しています。欧文の見出しでは大文字のみでテキストを構成することがありますが、通常の大文字であると強すぎる印象を与えてしまうこともあります。`titling-caps` は大文字を並べることを想定した字形です。

Value	Value Composition
<code>&lt;font-variant-caps&gt;</code>	<code>small-caps</code>   <code>all-small-caps</code>   <code>petite-caps</code>   <code>all-petite-caps</code>   <code>titling-caps</code>   <code>unicase</code>

`small-caps` で字形にスモールキャピタルを指定します。OpenType では `smcp` フィーチャとなります。

`small-caps` の指定では元から大文字だったテキストに影響はありませんが、`all-small-caps` の指定では、元のテキストが大文字であってもスモールキャピタルとして表示します。OpenType では `c2sc` フィーチャ (あるいは `smcp` フィーチャ) となります。

`petite-caps` で字形に `petite-caps` を指定します。OpenType では `pcap` フィーチャです。

`all-petite-caps` の指定では `all-small-caps` と同様に字形を `petite-caps` に変更します。`c2pc` フィーチャ (あるいは `pcap` フィーチャ) となります。

`unicase` の指定では、元テキストの大文字をスモールキャピタルに、小文字は通常 of 字形に変更します。OpenType フィーチャは `unic` です。

titling-caps は字形に titling-caps を指定します。OpenType フィーチャとしては titl です。

## small-caps エミュレーションの縮小比率を指定する

通常、small-caps、all-small-caps が指定された箇所ではスモールキャピタルの字形が利用されます。フォントがスモールキャピタル用の字形を持たないとき、大文字の字形を縮小 (small-caps のエミュレーション) して表示します。

Option Property Name	small-caps-emulation-size
Value	<percentage>
Initial	70%
Applies to	<formatter-settings>
Percentages	the font size

オプション設定 small-caps-emulation-size プロパティでこの縮小比率を指定可能です。スモールキャピタルが指定された箇所でのフォントサイズが<percentage>の基準になります。

Option Property Name	small-caps-emulation-x-height
Value	true   false
Initial	true
Applies to	<formatter-settings>

スモールキャピタルをエミュレートするとき、フォントの x-height と caps-height の情報を利用することで、一律の比率指定よりも適したサイズにエミュレートが可能です。

オプション設定 small-caps-emulation-x-height へ true を指定したとき、フォントが x-height と caps-height を持つのであれば、スモールキャピタルのエミュレートを行うときに x-height/cap-height の値を倍率として利用します。持っていなければ small-caps-emulation-size プロパティの値が利用されます。

## 常にエミュレートしたスモールキャピタルを使用する

Option Property Name	small-caps-emulation-always
Value	true   false
Initial	false
Applies to	<formatter-settings>

オプション設定 smallcaps-emulation-always プロパティへ true を指定したとき、スモールキャピタルは、フォント情報にかかわらず常にエミュレートによるものとなります。

## 2.18.9 数字、序数、分数の字形を指定する

数字の字形を指定します。font-variant プロパティへの指定となります。

Value	Value Composition
<code>&lt;font-variant-numeric&gt;</code>	<code>&lt;numeric-figure-values&gt;</code>    <code>&lt;numeric-spacing-values&gt;</code>    <code>&lt;numeric-fraction-values&gt;</code>    <code>ordinal</code>    <code>slashed-zero</code>
<code>&lt;numeric-figure-values&gt;</code>	<code>lining-nums</code>   <code>oldstyle-nums</code>
<code>&lt;numeric-spacing-values&gt;</code>	<code>proportional-nums</code>   <code>tabular-nums</code>
<code>&lt;numeric-fraction-values&gt;</code>	<code>diagonal-fractions</code>   <code>stacked-fractions</code>

`lining-nums` は数字の高さが揃った字形で、`oldstyle-nums` は小文字アルファベットのように数字によって高さが異なります。

`proportional-nums` では数字でも詰め処理が行われ、`tabular-nums` では行われません。

`tabular-nums` を指定すると数字を見比べるような表で、桁数と位置が揃います。

`diagonal-fractions` では、分母と分子間に斜線を引いたような表示となり、`stacked-fractions` では分母と分子間に水平線を引いたような表示となります。

`ordinal` は「1st」など序数の字形、`slashed-zero` は「0」の中心に斜線を引き、LATIN CAPITAL LETTER O「O」(U+004F)などと区別を付けやすくした字形です。

## 2.18.10 リガチャ（合字）を制御する

リガチャ（合字処理）対象の指定が可能です。

### リガチャを行う言語を指定する

Property Name	<code>axf:ligature-mode</code>
Value	<code>none</code>   [ <code>latin</code>    <code>kana</code>    <code>jamo</code> ]   <code>all</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	all block-level and inline-level formatting objects
Inherited	<code>yes</code>
Option Property Name	<code>latin-ligature</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

Option Property Name	jamo-ligature
Value	true   false
Initial	true
Applies to	<formatter-settings>

axf:ligature-mode プロパティへ、リガチャを行う対象の指定が可能です。フォントにリガチャが用意されていない場合は処理は行われません。none のときリガチャは行われません。

Unicode 定義でのリガチャ（正規合成）については axf:normalize プロパティで制御可能です。

axf:ligature-mode	意味
latin	スクリプトが Latn、Grek、Cycl のいずれかのときリガチャを行う。
kana	JIS X 0213:2004 でのかなに加え COMBINING KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK (U+309A) のリガチャを行う。
jamo	ハングル字母 U+1100 から U+11FF のリガチャを行う。
all	"latin kana jamo"とみなす。
auto	kana のリガチャを行う。他はオプション設定による。

axf:ligature-mode="auto"のときの latin、jamo のリガチャを行うかはそれぞれオプション設定で制御します。

オプション設定 latin-ligature プロパティへ true を指定すると latin のリガチャを、オプション設定 jamo-ligature プロパティへ true を指定すると jamo のリガチャを行います。

## リガチャを種類ごとに指定する

リガチャにはいくつか種類があります。axf:ligature-mode プロパティへの指定により、言語によってリガチャを行うかどうかを制御します。種類ごとのリガチャを有効、無効にするには font-variant プロパティへ指定します。

Value	Value Composition
<common-lig-values>	common-ligatures   no-common-ligatures
<historical-lig-values>	historical-ligatures   no-historical-ligatures
<discretionary-lig-values>	discretionary-ligatures   no-discretionary-ligatures
<contextual-values>	contextual   no-contextual

common-ligatures は fi や ffi などの、一般的なリガチャを有効にします。

historical-ligatures はドイツ語での tz などの、古代合字と呼ばれるリガチャを有効にします。

discretionary-ligatures は任意の合字、タイプデザイナーが設計するような想定のリガチャを有効にします。

contextual は筆記体の文字の繋がる部分など、文脈依存で使用する字を変更するようリガチャを有効にします。

no-\*の指定でその種類のリガチャを無効にします。

<common-lig-values>の指定例を示します [図 2.18-2] (p.226)。

```
<!-- common-lig-valuesの指定 -->
<fo:block font-variant="common-ligatures">difficult</fo:block>
<fo:block font-variant="no-common-ligatures">difficult</fo:block>
```

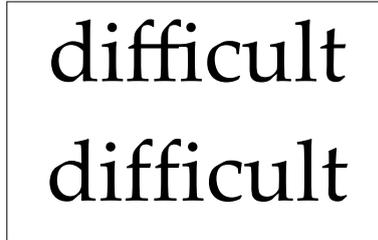


図 2.18-2 <common-lig-values>の指定

## 2.18.11 装飾的な字形を指定する

アクセントになるような装飾的な字形を指定します。font-variant プロパティへ指定します。

font-variant プロパティへ、装飾的な異体字の指定が可能です。

CJK の異体字については<font-variant-east-asian>で扱います。

Value	Value Composition
<font-variant-alternates>	historical-forms   stylistic()   swash()   ornament()   annotation()   styleset()   character-variant()
Function Name	stylistic(), swash(), ornament(), annotation()
Argument Value	<number>
Function Name	styleset(), character-variant()
Argument Value	<number>#

historical-forms は歴史的な字形を指定します。OpenType では hist フィーチャです。

historical-forms 以外では<number>を引数に取ります。これは OpenType の<feature-index>と同じものを指定します。

stylistic 字形を指定するには stylistic() へ<feature-index>を指定します。OpenType では salt フィーチャです。

swash は先端が流線的な字形です。指定するには swash() へ<feature-index>を指定します。OpenType では cswh フィーチャです。

ornament は飾り文字やフルーロンといった記号の字形です。簡条書きや段落の開始記号などに利用されます。ornament() へ<feature-index>を指定します。OpenType では ornm フィーチャです。

annotation は丸数字や鏡文字などの字形です。annotation() に<feature-index>を指定します。OpenType では nalt フィーチャです。

stylistic sets の字形を指定するには `styleset()` へ `<feature-index>` を指定します。複数の指定が可能です。OpenType では `ss` フィーチャです。

`character-variant` はフォント内であまりまとまりのない異体字です。`character-variant()` へ `<feature-index>` を指定します。OpenType では `cv` フィーチャです。

## 2.18.12 上付き、下付き文字用の字形を指定する

上付き、下付き文字用の字形を指定します。`font-variant` プロパティへの指定となります。

Value	Value Composition
<code>&lt;font-variant-position&gt;</code>	<code>super</code>   <code>sub</code>

`super` では上付き文字用の、`sub` では下付き文字用の字形を適用します。

## 2.18.13 CJK の異体字、半角・ルビ字形、繁体・簡体字を指定する

CJK の異体字や半角文字などを指定します。`font-variant` プロパティへの指定です。

Value	Value Composition
<code>&lt;font-variant-east-asian&gt;</code>	<code>&lt;east-asian-variant-values&gt;</code>    <code>&lt;param.east-asian-width-values&gt;</code>    <code>ruby</code>
<code>&lt;east-asian-variant-values &gt;</code>	<code>jis78</code>   <code>jis83</code>   <code>jis90</code>   <code>jis04</code>   <code>hojo-kanji</code>   <code>nlckanji</code>   <code>simplified</code>   <code>traditional</code>
<code>&lt;east-asian-width-values&gt;</code>	<code>full-width</code>   <code>half-width</code>   <code>third-width</code>   <code>quarter-width</code>   <code>proportional-width</code>

いずれも `font-variant` プロパティで指定します。

JIS 1978、JIS 1983、JIS 1990、JIS 2004 で字形の違い、異体字をそれぞれ `jis78`、`jis83`、`jis90`、`jis04` で指定可能です。

JIS 1978 の異体字を一部のテキストに対し指定した例を示します [図 2.18-3] (p.227)。

```
<!-- font-variant="jis78"をテキストの一部に指定する -->
<fo:block font-fmaily="'Noto Serif CJK Jp'"
  >繫という字はJIS78では<fo:inline
    font-variant="jis78">繫</fo:inline>と書く。</fo:block>
```

繫という字は JIS78  
では繫と書く。

図 2.18-3 `font-variant="jis78"` をテキストの一部に指定する

ほかに補助漢字を `hojo-kanji`、NLC 漢字を `nlckanji` で指定可能です。

`ruby` はルビ用の字形です。

中国語フォントでは、`simplified` で簡体字、`traditional` で繁体字の指定となります。

CJK の字幅の指定は、全角 (full-width)、半角 (half-width)、三分幅 (third-width)、四分幅 (quarter-width)、プロポーションナル幅 (proportional-width) を指定可能です。

## 2.18.14 OpenType のフォントフィーチャを直接制御する

GSUB や GPOS など、OpenType フォントのフィーチャを直接指定します。

OpenType フォントは、フォント自体に組版に関連する多様なフィーチャの格納が可能です。これらのフィーチャは組版時に AH XSL Formatter によって内部的に利用されますが、ユーザによる直接指定を可能にするプロパティが用意されています。

Property Name	axf:font-feature-settings
Value	normal   <feature-tag-value>#
Initial	normal
Applies to	all elements
Inherited	yes

OpenType フォントのフィーチャを axf:font-feature-settings プロパティへ指定可能です。normal では何もしません。

<feature-tag-value>には、OpenType の GSUB/GPOS のタグを次の書式で指定します。

Value	Value Composition
<feature-tag-value>	<string>[ <integer>   on   off]?

<string>には OpenType のタグ名を、SPACE (U+0020) から TILDE 「~」 (U+007E) の ASCII 文字列で構成される 4 文字で指定します。4 文字に満たないときは SPACE が補われます。

<integer>はグリフの選択インデクスまたは機能の有効無効を表す 0 以上の整数です。機能の有効無効として指定するとき、1 は on と同様に有効化、0 は off と同様に無効化を意味します。省略時は 1 とみなされます。

'ccmp'、'locl'、'rlig'、'mark'、'mkmk'は設定によらず常に有効です。

### 注

font-variant、axf:ligature-mode、axf:kerning-mode プロパティへの指定よりも axf:font-feature-settings プロパティへの指定が優先されます。

## 2.18.15 WOFF、WOFF2 フォントなどを指定する

組版環境のフォントパスに存在しない Web フォントなどを、選択可能なフォントに追加します。

Element Name	<axf:font-face>
Property	src, font-family, font-style, font-weight
Parent	fo:declarations

<axf:font-face>要素へ追加するフォントを指定します。  
 src プロパティへ追加するフォントのリソースを指定します。  
 追加したフォントを FO 内で指定するときのフォント名を font-family プロパティに記述します。  
 どのフォントスタイルとして扱うかを font-style プロパティへ指定します。normal または italic が指定可能です。  
 どのウエイトとして扱うかを font-weight プロパティへ指定します。normal、bold、1 から 1000 までの数で指定可能です。

Property Name	src
Value	[ <uri-specification>[ format( <font-type>)]?   <font-face-name>]#
Initial	N/A
Applies to	axf:font-face

src プロパティへ追加するフォントのリソースを指定します。  
 <uri-specification>はフォント名または url() 関数、local() 関数で URI を指定します。url() には data スキームを指定可能です。  
 format() 関数でフォントの種類を明示的に指定可能です。省略されたときは拡張子や data スキームのメディアタイプから判断されます。指定可能な値は truetype、opentype、woff、woff2 です。  
 別の<axf:font-face>でフォントファミリ名とし記述した名前も指定可能です。

## 2.18.16 オブリーク体の傾きを指定する

オブリーク体の傾きをシステム既定値から変更します。

Option Property Name	oblique-skew
Value	<number>
Initial	0
Applies to	<formatter-settings>

オプション設定 oblique-skew プロパティへ、フォントのスタイルが oblique または backslant のときの傾き量を<number>で指定可能です。0 以下のとき、システム既定値とみなされます。

### 注

フォントファミリへのスタイル指定が italic でそのフォントにイタリック体がないときは、常にシステムの既定値で傾けられ、このオプション設定は無効です。

どうしても独自の角度で傾きをかけたいとき、たとえばブロックへ `axf:transform="skew()"`を指定し、疑似的にテキストの変形が可能です。

GUI、PDF、PostScript、XPS 出力で有効です。

## 2.18.17 フォントファミリの記法を厳密にチェックする

ゼネリックフォントを除き、フォントファミリの指定は、厳密には APOSTROPHE 「'」 (U+0027) などで囲って記述することが望ましいでしょう。AH XSL Formatter では、この厳密な記法に限定するかどうかをオプション設定により制御可能です。

Option Property Name	family-name-syntax
Value	strict   auto   loose
Initial	auto
Applies to	<formatter-settings>

オプション設定 `family-name-syntax` プロパティへ `loose` または `auto` を指定したとき、フォントファミリの指定は厳密なものでもなくとも許容されます。

`strict` を指定したとき厳密なチェックを行い、適切に記述されていないときはエラーとして処理されます。

## 2.18.18 スクリプトごとのゼネリックフォントを指定する

AH XSL Formatter では OS に合わせて既定のゼネリックフォントを設定します。ゼネリックフォントは GUI の「組版オプション」のほか、オプション設定で変更可能です。

Option Element Name	<script-font>
Property	script, serif, sans-serif, cursive, fantasy, monospace, fallback
Parent	<font-settings>

オプション設定 `<script-font>` 要素はプロパティとしてスクリプトコードを指定する `script` プロパティと、そのスクリプトコードのときのゼネリックフォントを指定する `serif`、`sans-serif`、`cursive`、`fantasy`、`monospace` プロパティを持ちます。また、それらが見つからないときに採用するフォントを `fallback` プロパティへ指定します。

`script` プロパティへ空文字列を指定すると、標準のゼネリックフォントに対する指定となります。

日本語のテキストで `serif` に「源ノ明朝」フォント、`sans-serif` に「源ノ角ゴシック」フォントを指定したいとき、次のように記述します。

```
<!-- 日本語のジェネリックフォントを指定する -->
<formatter-config>
```

```

...
<font-settings>
  <script-font script="Jpan"
    serif="源ノ明朝" sans-serif="源ノ角ゴシック"/>
  ...
</font-settings>
</formatter-config>

```

## 2.18.19 フォントのフォールバックとエラー報告を制御する

指定されているフォントや字形が見つからなかったときの処理を制御します。

指定したフォントや字形が見つからないとき、フォールバックフォントの利用が可能です。

AH XSL Formatter では、フォントのフォールバック処理を制御するオプション設定を用意しています。フォールバック処理を行うかどうかを制御するものと、処理が行われるときのエラー報告レベルを指定するものがあります。

エラー報告レベルを指定するプロパティでは、値が 0 のときエラー報告はされません。1 以降はそのエラーレベルとして報告されます。

Option Property Name	auto-fallback-font
Value	true   false
Initial	true
Applies to	<font-settings>

指定されたフォントファミリー中に字形を持つものが見つからなかったときにフォールバックフォントを探すかどうかを、オプション設定 auto-fallback-font プロパティで制御します。true のときフォールバックフォントを探索します。

Option Property Name	fallback-glyph
Value	0   1   2   3   4
Initial	1
Applies to	<font-settings>

auto-fallback-font="true" のとき、フォントはフォールバックされます。このフォールバックフォントで字形が見つかったときのエラー報告レベルを、オプション設定 fallback-glyph プロパティで制御します。

Option Property Name	missing-font
Value	0   1   2   3   4
Initial	1
Applies to	<font-settings>

指定されたフォントファミリーからフォントが見つからなかったときのエラー報告レベルを、オプション設定 `missing-font` プロパティで制御します。

Option Property Name	<code>missing-glyph</code>
Value	<code>0</code>   <code>1</code>   <code>2</code>   <code>3</code>   <code>4</code>
Initial	<code>1</code>
Applies to	<code>&lt;font-settings&gt;</code>

指定されたフォントファミリーまたはフォールバックしたフォントで指定された文字の字形が見つからなかったときのエラー報告レベルを、オプション設定 `missing-glyph` プロパティで制御します。

Option Property Name	<code>missing-glyph-all</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>&lt;font-settings&gt;</code>

オプション設定 `missing-glyph` プロパティによる報告は、通常同じ文字について一度しかエラー報告を行いません。オプション設定 `missing-glyph-all` プロパティへ `true` を指定すると、既にエラー報告されている文字についても報告を行います。

この指定は、`fallback-glyph` プロパティによるエラー報告についても同様に影響します。

## italic、small-caps のエミュレートを報告する

フォントの修飾に `italic` を指定したとき、AH XSL Formatter は該当する書体がなければ機械的にフォントに傾きをかけて擬似的に斜体を表現します。

`font-variant="small-caps"` によるスモールキャピタルでは、スモールキャピタルの字形がないときに大文字の字形から擬似的にスモールキャピタルを表現します。

Option Property Name	<code>emulated-italic</code>
Value	<code>0</code>   <code>1</code>   <code>2</code>   <code>3</code>   <code>4</code>
Initial	<code>0</code>
Applies to	<code>&lt;font-settings&gt;</code>

Option Property Name	<code>emulated-small-caps</code>
Value	<code>0</code>   <code>1</code>   <code>2</code>   <code>3</code>   <code>4</code>
Initial	<code>0</code>
Applies to	<code>&lt;font-settings&gt;</code>

斜体のエミュレート処理が行われたときのエラー報告レベルをオプション設定 `emulated-italic` プロパティで指定可能です。

同様にスモールキャピタルのエミュレート処理が行われたときのエラー報告レベルをオプション設定 `emulated-small-caps` プロパティで指定可能です。

## 2.18.20 フォントファミリをオプション設定から置き換える

FO に特定のフォントが指定されていて、AH XSL Formatter を動かす環境にそのフォントが存在しないことがあるでしょう。このとき、FO の修正やフォントのフォールバック処理に任せるのではなく、オプション設定へ指定した代替フォントで組版するように設定が可能です。

Element Name	<code>&lt;font-alias&gt;</code>
Property	<code>src, dst</code>
Parent	<code>font-settings</code>

FO 内で記述されているフォントファミリへのエイリアスの設定を、オプション設定 `<font-alias>` へ記述可能です。

`src` プロパティへ FO 中に記述されているフォント名を指定し、`dst` プロパティへ指定したフォント名に置き換えて組版します。

たとえば、FO 内で「`font-family="Futura"`」と指定されている箇所があり、「Futura」フォントを持たない環境で AH XSL Formatter を利用するとき、該当する箇所を「Times New Roman」フォントで組版するための指定を次に示します。

```
<!-- FO -->
<fo:block font-family="Futura">...</fo:block>
```

```
<!-- オプション設定 -->
<font-settings><font-alias src="Futura" dst="Times New Roman">
</font-settings>
```

### ⚠ 注意事項

`dst` プロパティへ、別の `<font-alias>` の `src` プロパティに指定した値を指定しても置き換えはされません。

## 2.19 節 ルビ、圏点

ルビや圏点を細かく制御します。

ルビは、テキストの傍に付けて読みや意味などを示す小さな文字です。また、ルビを付ける対象を親文字と呼びます。ルビについての詳細は次の資料をご覧ください。

- 日本語組版処理の要件（W3C 技術ノート 2012 年 4 月 3 日）
- JIS X 4051 『日本語文書の組版方法』
- CSS Ruby Layout Module Level 1 W3C Working Draft, 5 August 2014
- Ruby Annotation W3C Recommendation 31 May 2001（Markup errors corrected 25 June 2008）

AH XSL Formatter では、日本語組版処理の要件（W3C 技術ノート 2012 年 4 月 3 日）に従いルビを実装しています。

### 2.19.1 ルビを記述する

拡張要素によってルビを記述することで、細かな調整を可能にします。

XSL 1.1 仕様の範囲でルビを再現するのであれば、たとえば、インラインコンテナを利用し、一部ののみフォントサイズを変更した行を配置する方法が考えられます。しかしこういった方法では、行分割や行頭行末での処理、ルビと親文字の長さの調整などを適切に行うことが大変難しくなります。

Element Name	<axf:ruby>
Property	axf:ruby-align, axf:ruby-position, axf:ruby-offset, axf:ruby-overhang, axf:ruby-limit-overhang, axf:ruby-limit-space, axf:ruby-small-kana, axf:ruby-font-family, axf:ruby-minimum-font-size, axf:ruby-font-style, axf:ruby-font-weight, axf:ruby-font-stretch, axf:ruby-condense, axf:ruby-font-color
Parent	block-level and inline-level elements
Element Name	<axf:ruby-base>
Parent	axf:ruby
Element Name	<axf:ruby-text>
Parent	axf:ruby

ルビを使うには、まずルビを振る範囲を<axf:ruby>で囲い、その子要素<axf:ruby-base>要素の内容に親文字を、子要素<axf:ruby-text>の内容にルビを記述します。

<axf:ruby-base>と<axf:ruby-text>の組をルビコンテナと呼ぶこととします。このルビコンテナは同じ<axf:ruby>内に複数記述が可能です。また、ルビコンテナ間に行分割が可能です。

次の例では「親文字 1 つめ」と「ルビ 1 つめ」、「親文字 2 つめ」と「ルビ 2 つめ」がそれぞれルビコンテナの組となります。

```
...
<axf:ruby><axf:ruby-base>親文字1つめ</axf:ruby-base>
  <axf:ruby-text>ルビ1つめ</axf:ruby-text>
  <axf:ruby-base>親文字2つめ</axf:ruby-base>
  <axf:ruby-text>ルビ2つめ</axf:ruby-text>
</axf:ruby>...
```

ルビ制御のプロパティは<axf:ruby>要素へ指定します。

## 注

<axf:ruby-text>要素にはテキストのみ記述が可能です。そのため、ルビを親文字としたさらに小さなルビなど、入れ子が必要な複雑な処理は行えません。

## 2.19.2 ルビのフォントを修飾する

ルビのフォントを親文字とは独立して指定します。

ルビに対し小さくとも可読性の高いフォントを別に指定する、親文字に傾きをかけてもルビはまっすぐに指定するなどが可能です。

Property Name	axf:ruby-color
Value	<color>
Initial	empty
Applies to	axf:ruby
Inherited	yes
Property Name	axf:ruby-font-family
Value	[ <family-name>   <generic-family> ]#
Initial	empty string
Applies to	axf:ruby
Inherited	yes
Property Name	axf:ruby-font-style
Value	normal   italic
Initial	normal
Applies to	axf:ruby
Inherited	yes

Property Name	axf:ruby-font-weight
Value	normal   bold   bolder   lighter   1... 1000
Initial	normal
Applies to	axf:ruby
Inherited	yes

axf:ruby-color、axf:ruby-font-family、axf:ruby-font-style、axf:ruby-font-weight プロパティは無指定のとき、親文字と同じ修飾になります。

Property Name	axf:ruby-font-size
Value	<number>   <absolute-size>   <relative-size>   <length>   <percentage>
Initial	0.5
Applies to	axf:ruby
Inherited	yes
Percentages	refer to the font size

Property Name	axf:ruby-minimum-font-size
Value	none   <absolute-size>   <relative-size>   <length>   <percentage>
Initial	none
Applies to	axf:ruby
Inherited	yes
Percentages	refer to the font size

親文字とルビのフォントサイズを独立して指定したいときやサイズの比を変更したいとき、axf:ruby-font-size プロパティへ指定します。<number>での指定は、親文字のフォントサイズと掛け合わせたものになります。

axf:ruby-minimum-font-size プロパティにはルビのフォントサイズの下限を指定可能です。このとき、ルビのフォントサイズはその指定以下にはなりません。たとえば、axf:ruby-font-size="0.5"をすべての段落で適用するが、本文より小さなフォントサイズを指定した脚注でもルビのサイズは本文と同じにしたい、といったときに有用です。

Property Name	axf:ruby-font-stretch
Value	normal   wider   narrower   ultra-condensed   extra-condensed   condensed   semi-condensed   expanded

extra-expanded | ultra-expanded | *<param.percentage>* |  
*<number>*

Initial	<i>empty</i>
Applies to	axf:ruby
Inherited	yes
Percentages	refer to the ruby font size

axf:ruby-font-stretch プロパティによってルビ文字の伸縮を親文字とは別に指定可能です。  
親文字をはみ出すときにのみルビの伸縮処理を行いたいときは axf:ruby-condense プロパティを  
指定します。

これらのプロパティは<axf:ruby>へ指定します。

### 注

ルビ用の字形のあるフォントでその字形をルビに適用したいときは<axf:ruby-text>要素  
に font-variant="ruby" を指定します。

## 2.19.3 ルビの揃えを指定する

ルビの均等割り付けとベタ組、start と end のどちらに揃えるか、縦組での揃えを指定します。

Property Name	axf:ruby-align
Value	auto   [[ space-between   space-around]    [ center   start   end   start-if-vertical]]
Initial	auto
Applies to	axf:ruby
Inherited	yes
Option Property Name	ruby-align
Value	[ space-between   space-around]    [ center   start   end   start-if-vertical]
Initial	space-around center
Applies to	<formatter-settings>

axf:ruby-align プロパティでルビの揃えを指定可能です。

space-\* を指定するものが均等割り付けで、含まない指定はベタ組になります。space-around はルビの前後にアキを取りますが、space-between では親文字の端にルビの先頭と末尾が揃えられます。

start は start 側、center は中央、end は end 側へのベタ組です。start-if-vertical は、縦組では start、横組では center と同じです。

また、space-around center のように指定すると、ルビが 1 文字のときのみ 2 番目に指定した方向へ揃えます。

たとえば「基本はルビ前後を空けた均等割り付けで、ルビが 1 文字のときは縦組なら肩付きルビに、横組なら中央揃えのルビにしたい」というときの指定は `axf:ruby-align="space-around start-if-vertical"` となります。

## 2.19.4 ルビの配置を指定する

ルビを親文字のどちら側に配置するか、位置をどの程度移動するかを指定します。

Property Name	<code>axf:ruby-position</code>
Value	<code>before</code>   <code>after</code>   <code>[[ over   under   <code>inter=character</code> ]]</code> <code>[ right   left]</code>
Initial	<code>before</code>
Applies to	<code>axf:ruby</code>
Inherited	yes

`axf:ruby-position` プロパティへ、親文字に対するルビの配置を指定します。[CSS3-Ruby] Ruby positioning: the ‘ruby-position’ property に基づくプロパティです。

横組のときは `before` と `over` が親文字の上側、`after` と `under` が親文字の下側です。

縦組のときは `before` と `right` が親文字の右側、`after` と `left` が親文字の左側です。

`inter-character` は台湾のルビなどに使われますが、AH XSL Formatter では対応していません。

Property Name	<code>axf:ruby-offset</code>
Value	<code>&lt;number&gt;</code>   <code>&lt;length&gt;</code>   <code>&lt;percentage&gt;</code>
Initial	<code>0pt</code>
Applies to	<code>axf:ruby</code>
Inherited	yes
Percentages	refer to the font size

ルビと親文字の間隔を `axf:ruby-offset` プロパティへ指定可能です。`axf:ruby-position` プロパティを指定しているのであれば、その配置からの間隔となります。

## 2.19.5 親文字をはみ出すルビを制御する

ルビが親文字をはみ出す長さ、サイズのときの動作を詳細に指定します。

文字数やルビのフォントサイズの都合で、ルビが親文字の幅をはみ出すことがあります。このとき、親文字と隣りあう文字との字間を広げるほか、処理の方針としては「ルビのフォントサイズ、幅、高さを変更する」「隣の文字へかけるルビを調整する」といった方法が考えられます。前者は `axf:ruby-condense`、後者は `axf:ruby-overhang` プロパティで制御が可能です。

Property Name	axf:ruby-condense
Value	none   <i>&lt;percentage&gt;</i>   <i>&lt;number&gt;</i>
Initial	none
Applies to	axf:ruby
Inherited	yes
Percentages	refer to the ruby-font size

axf:ruby-condense プロパティでは、ルビが親文字をはみ出すとき、インライン進行方向にルビを圧縮する割合を指定可能です。この指定によって、たとえば「親文字をはみ出すときは三分ルビにする」といった制御が可能です。

Property Name	axf:ruby-overhang
Value	none   auto   [ start    end    except-kanji    except-katakana    except-hiragana    except-kana    except-start-open-parenthesis    except-end-close-parenthesis ]
Initial	auto
Applies to	axf:ruby
Inherited	yes

Property Name	axf:ruby-limit-overhang
Value	<i>&lt;number&gt;</i>   <i>&lt;length&gt;</i>   <i>&lt;percentage&gt;</i>
Initial	1.0
Applies to	axf:ruby
Inherited	yes
Percentages	refer to the font size

ルビが親文字をはみ出すとき、隣の文字にルビが侵入するかどうかの制御を axf:ruby-overhang プロパティへ指定します。

侵入するルビの量の限界値は axf:ruby-limit-overhang プロパティに指定可能です。

axf:ruby-overhang プロパティの指定は、指定した側の隣の文字へルビを掛けることを許可する start と end、許可しない none と、許可から除外を行う指定である except-\*があります。

auto を指定すると、start end except-kanji と同じになります。

除外指定は和文を想定した指定となります。except-hiragana がひらがな、except-katakana がカタカナ、except-kana はひらがなとカタカナが隣の文字であるときにルビを隣の文字へ掛けません。except-kanji は隣の文字が漢字のとき、except-start-open-parenthesis は start 側の隣の文字が開き括弧類、except-end-close-parenthesis は end 側の隣の文字が閉じ括弧類のとき、ルビを隣の文字へ掛けません。

[図 2.19-1] (p.240) に例を示します。

「end 側の開き括弧」や「start 側の閉じ括弧」を除外する指定はありません。

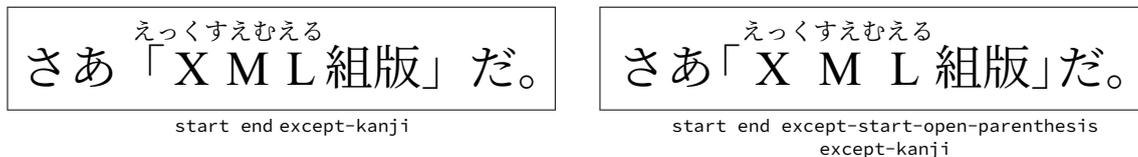


図 2.19-1 axf:ruby-overhang

たとえば「文末のルビが句点を越えないようにしたい」というときは `axf:ruby-overhang` プロパティの指定から `end` を除きます。

`axf:ruby-limit-overhang` プロパティには `<percentage>` を指定可能ですが、このとき、ルビではなく親文字のフォントサイズが基準となります。たとえば「隣の文字の半分まではルビが掛かってもいい」のであれば `axf:ruby-limit-overhang="50%"` と指定します。

隣に掛かる対象がないとき、親文字の字間を空ける処理が行われます [図 2.19-2] (p.240)。



図 2.19-2 隣の文字の有無でのルビ処理の違い

## 2.19.6 小書きかなのルビを普通のかなに変換する

ルビで、和文字形の一部についての変換を制御します。

Property Name	<code>axf:ruby-small-kana</code>
Value	<code>auto</code>   <code>preserve</code>   <code>convert</code>
Initial	<code>auto</code>
Applies to	<code>axf:ruby</code>
Inherited	<code>yes</code>

ルビにおいて HIRAGANA LETTER SMALL TU 「っ」 (U+3063) や HIRAGANA LETTER SMALL A 「ぁ」 (U+3041) など小書きのかな (拗促音) を HIRAGANA LETTER TU 「つ」 (U+3064) や HIRAGANA LETTER A 「あ」 (U+3042) など普通のかなで代替して表記する組版方法があります。

`axf:ruby-small-kana` プロパティでこの変換を制御します [図 2.19-3] (p.241)。

親の要素に `text-transform="fullsize-kana"` が指定されているとき、この指定が `preserve` でも普通のかなに変換されます。

axf:ruby-small-kana	意味
auto, preserve	変換は行わない。
convert	小書きのかなを普通のかなに変換する。



図 2.19-3 axf:ruby-small-kana

## 2.19.7 圏点を指定する

和文などでテキストの傍に配置する圏点を指定します。

主に和文で、テキストを強調する方法として、文字の傍に小さく記号を配置する圏点があります。圏点が付加可能なスクリプトは、CJK と Latn、Cyrl、Grek、Zyyy です。また、圏点はルビの親文字には付加可能ですが、ルビには付加できません。

### 圏点として利用する記号を指定する

Property Name	axf:text-emphasis-style
Value	none   [[ filled   open]]   [ dot   circle   double-circle   triangle   sesame]]   <string>
Initial	none
Applies to	all elements
Inherited	yes

axf:text-emphasis-style プロパティへ、圏点として利用する記号や文字を指定可能です。

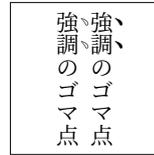
内側まで塗られた状態の filled、輪郭のみ（いわゆる「白抜き」）の状態 open と、sesame や dot など記号名を指定するか [図 2.19-4] (p.242)、テキストを直接指定します。

none のとき圏点は付加されません。

axf:text-emphasis-style	filled	open
dot	BULLET 「●」 (U+2022)	WHITE BULLET 「◦」 (U+25E6)
circle	BLACK CIRCLE 「◎」 (U+25CF)	WHITE CIRCLE 「◯」 (U+25CB)
double-circle	FISHEYE 「⦿」 (U+25C9)	BULLSEYE 「⦿」 (U+25CE)
triangle	BLACK UP-POINTING TRIANGLE 「▲」 (U+25B2)	WHITE UP-POINTING TRIANGLE 「△」 (U+25B3)
sesame	SESAME DOT 「◌」 (U+FE45)	WHITE SESAME DOT 「◌」 (U+FE46)



dot filled (上) open (下)



sesami filled (左) open (右)

図 2.19-4 axf:text-emphasis-style の例

## 圏点のフォントを修飾する

Property Name	axf:text-emphasis-font-family
Value	[ <family-name>   <generic-family> ]#
Initial	empty string
Applies to	emphasis elements
Inherited	yes

axf:text-emphasis-font-family プロパティへ圏点のサイズが揃った専用のフォントを指定することで、サイズ調整の手間を減らせます。

Property Name	axf:text-emphasis-color
Value	<color>
Initial	currentColor
Applies to	emphasis elements
Inherited	yes

Property Name	axf:text-emphasis-font-size
Value	<number>   <absolute-size>   <relative-size>   <length>   <percentage>
Initial	0.5
Applies to	emphasis elements
Inherited	yes
Percentages	refer to the font size

Property Name	axf:text-emphasis-font-style
Value	normal   italic
Initial	empty
Applies to	emphasis elements
Inherited	yes

Property Name	axf:text-emphasis-font-weight
Value	normal   bold   bolder   lighter   1... 1000
Initial	<i>empty</i>
Applies to	emphasis elements
Inherited	yes

Property Name	axf:text-emphasis-font-stretch
Value	normal   wider   narrower   ultra-condensed   extra-condensed   condensed   semi-condensed   expanded   extra-expanded   ultra-expanded   <i>&lt;percentage&gt;</i>   <i>&lt;number&gt;</i>
Initial	<i>empty</i>
Applies to	emphasis elements
Inherited	yes
Percentages	refer to the text-emphasis-font-size

axf:text-emphasis-font-\*プロパティなどの使い方はルビと同様になりますが、ルビとは独立に指定できます。

## 圏点を付加する文字を制御する

圏点を「約物に対して付加するか」「欧文に対して付加するか」といった処理を制御します。

Property Name	axf:text-emphasis-skip
Value	none   [ spaces    punctuation    symbols    narrow]
Initial	spaces
Applies to	all elements
Inherited	yes

axf:text-emphasis-skip プロパティへ指定した種類の文字には圏点が付加されません [図 2.19-5] (p.244)。spaces で空白文字、punctuation で約物、symbols で記号、narrow で半角文字など（幅広文字以外）に圏点が付加されなくなります。

```
<!-- 欧文と約物に圏点を付けない -->
<fo:block>
  <fo:inline axf:text-emphasis-font-size="0.5"
    axf:text-emphasis-font-family="'KentenGeneric'"
    axf:text-emphasis-style="filled dot"
    axf:text-emphasis-font-style="normal"
    axf:text-emphasis-skip="punctuation narrow"
  >この文は<fo:inline font-style="italic"
```

```
>emphasize</fo:inline>されています。</fo:inline>
</fo:block>
```

この文章全体が *emphasize* の対象です。

図 2.19-5 欧文と約物に圏点を付けない

## 圏点の配置を指定する

Property Name	axf:text-emphasis-position
Value	before   after
Initial	before
Applies to	all elements
Inherited	yes

axf:text-emphasis-position プロパティで圏点を配置する方向を指定します。

Property Name	axf:text-emphasis-offset
Value	<number>   <length>   <percentage>
Initial	0pt
Applies to	all elements
Inherited	yes
Percentages	refer to the font size

axf:text-emphasis-offset プロパティへ圏点のオフセットを指定可能です。たとえば圏点とルビを併用するとき、ルビとの位置を調整するときに有用です。

## 2.19.8 ユースケース：さまざまなルビを指定する

ルビ制御の拡張を利用して、さまざまなルビを組みます。

ルビ拡張を用いてルビを組む例を示します [図 2.19-6] (p.246)。肩付きルビや中付きルビはモノルビの概念ですが、ここではグループルビでも同様に扱います。ここで紹介している名称は厳密な分類とは異なるものもあります。

名称	概要
モノルビ	親文字 1 字にルビ 1 字を配置する。親文字が 1 字のグループルビとみなせる。
グループルビ	親文字全体にルビが配置される。
熟語ルビ	複数のルビコンテナで構成され、ルビコンテナ間で行分割ができる。
三分ルビ	ルビの start-end 方向のサイズを親文字の 1/3 に指定したもの。

名称	概要
肩付きルビ	ルビを start 側に配置する。
中付きルビ	ルビを中央に配置する。
逆ルビ	ルビを親文字の after 側に配置する。
両側ルビ	ルビを親文字の before 側と after 側両方に配置する。

熟語ルビではとくにルビが親文字をはみ出すようなとき、配置のパターンが複雑化します。

```
<!-- 肩付きルビ -->
<afx:ruby axf:ruby-align="start">
  <afx:ruby-base>齧</afx:ruby-base>
  <afx:ruby-text>そ</afx:ruby-text>
  <afx:ruby-base>齧</afx:ruby-base>
  <afx:ruby-text>ご</afx:ruby-text>
</afx:ruby>
```

```
<!-- 両側ルビ -->
<!-- ルビをbefore側に指定 -->
<afx:ruby axf:ruby-position="before">
  <afx:ruby-base>
    <!-- ルビをafter側に指定 -->
    <afx:ruby axf:ruby-position="after">
      <afx:ruby-base>薦擧</afx:ruby-base>
      <!-- after側のルビの内容 -->
      <afx:ruby-text>ス、メアゲ</afx:ruby-text>
    </afx:ruby>
  </afx:ruby-base>
  <!-- before側のルビの内容 -->
  <afx:ruby-text>センキヨ</afx:ruby-text>
</afx:ruby>シ
```

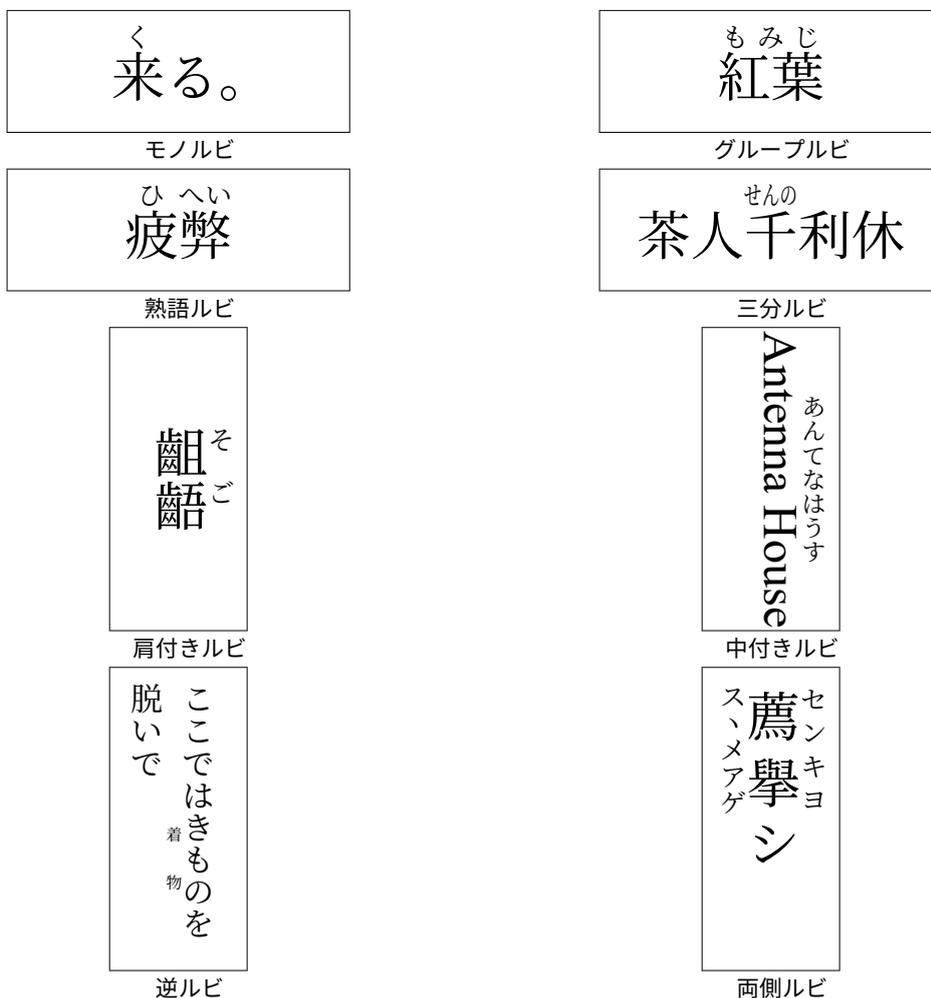


図 2.19-6 ルビの例

### 2.19.9 ユースケース：大きな親文字にルビを記述する

親文字のフォント修飾とルビのフォント修飾を別に指定します。

見出しのように、大きな文字を想定したフォントの修飾をそのままルビ部分に適用すると不都合になることがあります。[図 2.19-7] (p.246) ではルビのサイズ、伸縮、太さを明示的に指定しました。

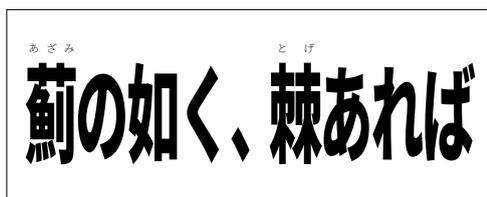


図 2.19-7 大きな親文字にルビを記述する

```

<!-- 大きな親文字にルビを記述する -->
<fo:block ... letter-spacing="-2pt"
font-stretch="ultra-condensed"
font-size="50pt" font-family="sans-serif"
font-weight="1000">
<axf:ruby ruby-font-size="10%"
ruby-font-stretch="normal" ruby-font-family="serif"
ruby-font-weight="1" ruby-offset="3pt">
<axf:ruby-base>薊</axf:ruby-base>
<axf:ruby-text>あざみ</axf:ruby-text>
</axf:ruby>の如く<axf:ruby ...>
<axf:ruby-base>棘</axf:ruby-base>
<axf:ruby-text>とげ</axf:ruby-text>
</axf:ruby>あれば
</fo:block>

```

axf:ruby-font-size は初期値では親文字の 50%となりますが、親文字が大きいサイズするとき、この指定のままだとルビが大きすぎるため、より小さく指定しました。

ルビは親文字とずらした箇所に小さい文字で表示するため、axf:ruby-font-stretch="normal"の指定で親文字とフォントの伸縮の比率を変えています。

親文字が大きいサイズのため、ルビの表示位置を axf:ruby-offset プロパティで調整しています。

axf:ruby-font-size へ lh や em を指定したとき、親文字の値を利用するため注意してください。ルビの修飾を変更しないとき、[図 2.19-8] (p.247) のように表示されます。



図 2.19-8 ルビの修飾を変更しない



# 第 3 章

## PDF 出力

## 3.1 節 PDF 出力

PDF のさまざまな機能が利用可能です。

PDF にはバージョンのほか、利用目的に応じたプロファイル仕様として「PDF/A」や「PDF/X」など、より細かな仕様が規定されています。出力するバージョンやプロファイル仕様によっては、無効化されたり有効でない仕様、機能も存在します。詳細についてはマニュアルをご覧ください。

### 3.1.1 出力する PDF のバージョンを指定する

出力する PDF のバージョンやプロファイル仕様をさまざまに指定可能です。

PDF は、新しいバージョンの公開ごとに新機能の追加や変更、廃止を行っています。たとえば、タグ付き PDF は PDF 1.4 で定義されました。廃止されていない機能は、より新しいバージョンにも含まれます。

バージョン	公開年
1.3	2000
1.4	2001
1.5	2005
1.6	2004
1.7	2008
2.0	2017

#### 注

より新しいバージョンでの機能の変化として、アクセシビリティやセキュリティ、メディア対応などで細かな指定が可能になっている点があります。

バージョンのほか、目的に応じたプロファイル仕様が ISO によって標準化されています。

**PDF/X** ISO 15930 で規定された。印刷用グラフィックス交換を目的とする。「フォントがすべて埋め込まれていなければならない」、「CMYK やスポットカラーのサポート」、「パスワード、印刷制限などの不許可」「リンク、注釈などの不許可」などを要件とする。

**PDF/A** ISO 19005 で規定された。電子文書の長期保存を目的とする。「フォントがすべて埋め込まれていなければならない」「XMP 準拠のメタデータを含んでいなければならない」「外部コンテンツの参照不許可」「JavaScript の不許可」「暗号化の不許可」などを要件とする。

**PDF/UA** ISO 14289-1 で規定された。ISO 32000-1 (PDF 1.7) 仕様にに基づき、PDF のアクセシビリティ向上を目的とする。「コンテンツが論理的な読み順にタグ付けされていること」「意味のある画像や注釈、数式には代替テキストを含めなければならない」「アクセシビリティを禁止し

ない暗号化を行わなければならない」「文書の言語を指定しなければならない」などを要件とする。

Option Property Name	pdf-version
Value	PDF1.3   PDF1.4   PDF1.5   PDF1.6   PDF1.7   PDF2.0   PDF/X-1a:2001   PDF/X-3:2002   PDF/X-1a:2003   PDF/X-2:2003   PDF/X-3:2003   PDF/X-4:2010   PDF/X-4p:2010   PDF/A-1a:2005   PDF/A-1b:2005   PDF1.4/A-2a:2011   PDF1.5/A-2a:2011   PDF1.6/A-2a:2011   PDF1.7/A-2a:2011   PDF1.4/A-2b:2011   PDF1.5/A-2b:2011   PDF1.6/A-2b:2011   PDF1.7/A-2b:2011   PDF1.4/A-2u:2011   PDF1.5/A-2u:2011   PDF1.6/A-2u:2011   PDF1.7/A-2u:2011   PDF1.4/A-3a:2012   PDF1.5/A-3a:2012   PDF1.6/A-3a:2012   PDF1.7/A-3a:2012   PDF1.4/A-3b:2012   PDF1.5/A-3b:2012   PDF1.6/A-3b:2012   PDF1.7/A-3b:2012   PDF1.4/A-3u:2012   PDF1.5/A-3u:2012   PDF1.6/A-3u:2012   PDF1.7/A-3u:2012   PDF1.5/UA-1:2014   PDF1.6/UA-1:2014   PDF1.7/UA-1:2014
Initial	PDF1.5
Applies to	<pdf-settings>

オプション設定 pdf-version プロパティに指定した PDF バージョン、プロファイル仕様の PDF が出力されます。

pdf-version プロパティで指定したよりも新しいバージョンの PDF は埋め込めません。

## 出力する PDF が要件に適合できないとき警告する

AH XSL Formatter では、PDF のバージョン指定に応じて不適合な内容を回避して PDF を出力します。しかし、「フォント埋め込みが必要な PDF/X で埋め込めないフォントが指定されている」など、不適合かつ回避不能なものも存在します。

Option Property Name	error-on-pdfx-fault
Value	true   false
Initial	true
Applies to	<pdf-settings>

PDF/X の要件に不適合なときの動作を、オプション設定 error-on-pdfx-fault プロパティで制御可能です。true のとき、PDF を生成せずエラー報告をします。false のときは PDF を生成しますが、指定した PDF の要件に不適合なときがあります。

## 3.1.2 ネットワークでの表示用に最適化する

Web などネットワーク環境での PDF 閲覧のために構造を最適化した「リニアライズド PDF」を出力します。

最初に開かれるページや、開いたページから遷移する可能性のあるページの表示を高速にするように構造を変更します。

Option Property Name	linearized
Value	true   false
Initial	true
Applies to	<pdf-settings>

オプション設定 `linearized` プロパティへ `true` を指定すると、リニアライズド PDF として出力します。一度通常の PDF を生成した後最適化を行うため、通常の PDF よりも出力に時間を要します。

GUI では「Web 表示用に最適化」で同様に指定可能です。

### 3.1.3 ファイルサイズを圧縮する

PDF 出力時にオブジェクトの圧縮などを行うことで、ファイルサイズを小さくすることが可能です。

Option Property Name	text-and-lineart-compression
Value	true   false
Initial	true
Applies to	<pdf-settings>

Option Property Name	object-compression
Value	true   false
Initial	true
Applies to	<pdf-settings>

オプション設定 `text-and-lineart-compression` プロパティへ `true` を指定すると、出力する PDF のテキストとラインアートを圧縮して PDF のファイルサイズを小さくします。埋め込まれる PDF には影響しません。

また、PDF 1.5 以降で `text-and-lineart-compression="true"` のとき、オプション設定 `object-compression` プロパティに `true` を指定すると、PDF 内のオブジェクトを圧縮します。

### 3.1.4 パスワードを指定する

出力した PDF の編集制御のためのオーナーパスワード、閲覧制御のためのユーザーパスワードを指定します。

Option Property Name	owner-password, user-password
Value	<string>
Initial	empty
Applies to	<pdf-settings>

オプション設定 owner-password プロパティで PDF のオーナーパスワードの、user-password でユーザパスワードのテキストを指定します。未指定のときパスワードは設定されません。

パスワードとして指定可能なテキストには「ASCII 範囲の文字であること」「PDF 1.3 から 1.7 では 32 文字まで」「PDF 2.0 以降では 127 文字まで」という制約があります。それ以外の文字が含まれていたときや文字数が超過しているとき、無効となり、パスワードは設定されません。

### ⚠ 注意事項

オプション設定ファイルや<axf:formatter-config>で記述する、パスワードのテキストは平文です。

## 3.1.5 暗号化を制御する

暗号化した PDF を出力するとき、暗号化方式や暗号化対象にメタデータを含むかなどを指定します。

Option Property Name	encryption-level
Value	40rc4   128rc4   128aes   256aes
Initial	128rc4
Applies to	<pdf-settings>

オプション設定 encryption-level プロパティで PDF を暗号化して出力するときのキー長を制御可能です。40rc4 は PDF 1.3 から 1.7 まで、128rc4 は PDF 1.4 から 1.7 まで、128aes は PDF 1.5 から 1.7 まで、256aes は PDF 1.7 以降のバージョンで有効な指定です。

Option Property Name	encrypt-metadata
Value	true   false
Initial	true
Applies to	<pdf-settings>

オプション設定 encrypt-metadata プロパティへ true を指定すると、出力 PDF の暗号化を行うとき、メタデータも含め暗号化します。対応する PDF のバージョンは 1.5 以降です。

## 注

256 AES は、PDF 1.7 では、アドビシステムズ製品での拡張として対応しています。PDF の仕様としては PDF 2.0 以降の対応となります。

### 3.1.6 画像、テキストのコピーを禁止する

PDF ビューアなどで、出力した PDF からのコンテンツのコピーを制限します。

Option Property Name	no-content-copying
Value	true   false
Initial	false
Applies to	<pdf-settings>

オプション設定 no-content-copying プロパティへ true を指定すると、出力した PDF でテキストとグラフィックファイルのコピーを不許可にします。

この制限を行うには owner-password プロパティへ値を指定する必要があります。

### 3.1.7 PDF のレイヤを指定する

PDF のレイヤ構造を利用し、表示内容を切り換えます。

対応した PDF ビューアでは設定に応じてレイヤを切り換え、表示内容の変更が可能です。設定可能な機能の多くは PDF 1.5 以降で有効です。

Property Name	axf:layer-settings
Value	none   <layer-setting>#
Initial	none
Applies to	root element

Value Name	Value Composition
<layer-setting>	<layer-name>[ on   off]?[ intent[ view   design]    print[ on   off]    export[ on   off]    lang <string> preferred?    zoom <number> <number>?    locked]?

<layer-settings>	意味
<layer-name>[ on   off]	<layer-name>へ、<string>でレイヤの名前を指定する。この名前は識別子として axf:layer で利用する。この項目の on と off は view、print、export が無指定のとき参照される。無指定では on とみなす。
intent[ view   design]	view を指定するとレイヤの表示非表示の切り換えが可能に、design では不可能になる。省略時は view とみなす。
print[ on   off]	on でレイヤを印刷する。
export[ on   off]	on でレイヤを書き出しに含める。
lang <string> preferred?	レイヤの言語を指定する。 <string>で指定した言語と PDF ビューアの言語が一致するときレイヤを表示する。 preferred を指定したとき、部分一致のときもレイヤを表示する。一致の判定は PDF ビューアに依存する。
zoom <number> <number>?	レイヤが表示される倍率の最小値と最大値を指定する。値をひとつだけ指定したとき、最小値として設定される。
locked	レイヤをロックし、編集を無効にする。PDF 1.6 以降で有効。

<fo:root>で axf:layer-settings プロパティにレイヤ名とレイヤの情報を定義します。カンマ区切りで複数のレイヤが定義可能です。

Property Name	axf:layer
Value	none   <layer-name>
Initial	none
Applies to	formatting object that generates area

axf:layer プロパティでは、記述した領域をどのレイヤとするかを指定します。axf:layer-settings で記述した <layer-name>を指定します。

```
<!-- 同じPDFでレイヤごとに表示内容が切り換わる -->
<fo:root ...
  axf:layer-settings="
    estimate on intent view view on print on export on lang 'ja-JP',
    application off intent view view on print on export on lang 'ja-JP'
  ">
...
<fo:page-sequence ...>
  <fo:flow ...>
    <fo:block-container absolute-position="absolute"
      axf:layer="estimate"
      top="0.2cm" left="0.5cm">
      ...
    </fo:block-container>
  </fo:flow>
</fo:page-sequence>
</fo:root>
```

afx:layer-settings で用意したレイヤ名は、出力された PDF でも利用されます。また、異なるレイヤを設定したオブジェクトも、組版時には位置関係が考慮されます。そのため同じ位置に重ねて配置するには、たとえばブロックコンテナの絶対配置を利用します。



図 3.1-1 同じ PDF でレイヤごとに表示内容が切り換わる

### 3.1.8 ページを逆順に出力する

すべてのページを逆順にして出力します。

Option Property Name	reverse-page
Value	true   false
Initial	false
Applies to	<pdf-settings>

オプション設定 reverse-page プロパティへ true を指定したとき、PDF 出力時、すべてのページを逆順に出力します。

それぞれのページを組版後、PDF 上のページ順序を逆順にします。

#### 注

FO 中に指定する `afx:reverse-page="true"` は指定したページシーケンスを逆順にしますが、`reverse-page="true"` は出力する PDF すべてのページを逆順にします。

### 3.1.9 PDF 内部で利用される実数値の範囲を制御する

PDF 内部では、コンテンツの情報保持や JavaScript での計算など、内部でさまざまな数値を利用します。この内部的な数値のサイズが PDF ビューアの動作に影響する可能性があります。

Option Property Name	real-value-limit
Value	0   1   2   3
Initial	2
Applies to	<pdf-settings>

Option Property Name	real-value-limit-modify
Value	true   false
Initial	false
Applies to	<pdf-settings>

オプション設定 `real-value-limit` プロパティでは、出力される実数値をチェックし、限界値を超えないように制御可能です。PDF 1.5 以降で有効です。

real-value-limit	意味
0	チェックをしない。
1	±32767 以内かチェックする。
2	約±2.14748×10 <sup>9</sup> 以内かチェックする。
3	約 ±9.22337×10 <sup>18</sup> 以内かチェックする。

出力する PDF のバージョンが 1.4 以下のとき、`real-value-limit` プロパティの値が 2 または 3 でも、1 を指定したとみなされます。

また、限界値を超えたときに限り値の補正をしたいというときは、オプション設定 `real-value-limit-modify` プロパティへ `true` を指定します。

PDF ビューアの動作を保証するものではありません。

## 3.2 節 文書情報の制御

PDF にタイトルや著者情報をメタデータとして埋め込んだり、PDF ビューアでの開き方などをある程度指定可能です。

たとえば Adobe Acrobat Reader では、「プロパティ」から文書情報が閲覧可能です [図 3.2-1] (p.258)。



図 3.2-1 Adobe Acrobat Reader で「プロパティ」を開き閲覧できる文書情報

Element Name	<axf:document-info>
Property	name, value
Parent	fo:declarations, fo:page-sequence

<fo:declarations>の子要素として<axf:document-info>を記述します。この要素の name プロパティに文書情報の項目、value プロパティにその項目の内容を記述します。value プロパティには単なるテキストを記述します。

分冊出力を行うときは<fo:page-sequence>の最初の子要素として置くことも可能です。<fo:page-sequence>の子要素の文書情報はそれぞれの分冊の文書情報として扱われます。このとき<fo:declarations>の子要素として記述した文書情報はすべての分冊に埋め込まれます。

```
<!-- 分冊での文書情報 -->
<fo:declarations>
  <!-- すべての文書に適用する文書情報 -->
  <axf:document-info name="author" value="アンテナハウス" />
  <!-- 分冊制御、ここではfo:declarationsで記述した
  文書情報をすべてのファイルに適用させる -->
  <axf:output-volume-info document-info-include="all" ... />
</fo:declarations>
...
<fo:page-sequence ... axf:output-volume-break="true" >
  <!-- この分冊のみに適用 -->
  <axf:document-info name="document-title" value="分冊1">
  ...
</fo:page-sequence>
<fo:page-sequence ... axf:output-volume-break="true">
  <!-- この分冊のみに適用 -->
```

```
<axf:document-info name="document-title" value="分冊2">
...
</fo:page-sequence>
```

### 3.2.1 基本的な文書情報を指定する

PDF ビューアの「プロパティ」で確認できるような、基本的な文書情報を記述します。

name	value
document-title	<文書のタイトル>
subject	<文書のサブタイトル>
keywords	<キーワード (カンマ区切り) >
author	<著者名>

```
<!-- 基本的な文章情報の埋め込み -->
<fo:declarations>
  <axf:document-info name="document-title" value="AHF-FO Guidebook" />
  <axf:document-info name="subject" value="文書情報の埋め込み" />
  <axf:document-info name="keywords" value="&lt;axf:document-info&gt;" />
  <axf:document-info name="author" value="Antenna House, Inc." />
  ...
</fo:declarations>
```

### 3.2.2 文書の著者、書き手についての情報を指定する

より詳細な文書情報を記述します。

name	value
author-title	<文書の著者の所属、役職>
description-writer	<文書の説明の書き手についての説明>

PDF ビューアによっては、詳細な文書情報の表示には対応していないものも存在します。

### 3.2.3 作成日、修正日を指定する

PDF の作成日や修正日を明示的に指定します。

name	value
createdate	<作成日時>
modifydate	<修正日時>

無指定のとき、PDF 出力時を現時刻とします。書式は ISO 8601 (JIS X 0301) 形式で、次の書式が有効です。

西暦表記      YYYYY-MM-DD(Thh(:mm(:ss)?)?)(Z|±hh(:mm)?)?

和暦表記      NYY.MM.DD(Thh(:mm(:ss)?)?)?

読み方を次に示します。

文字	意味
Y	年。0 から 9 の数字。
N	和暦の元号。昭、平、令、あるいは s、H、R のいずれか 1 字。
M	月。0 から 9 の数字。
D	日。0 から 9 の数字。
T	日付と時刻の区切り文字。
h	時刻の時。0 から 9 の数字。
m	時刻の分。0 から 9 の数字。
m	時刻の秒。0 から 9 の数字。
-	日付の数字間の区切り文字。省略可。
.	日付の数字間の区切り文字。省略可。
:	時刻の数字間の区切り文字。省略可。
Z	タイムゾーン。UTC であることを示す。
±hh(:mm)	タイムゾーン。標準時からの時差。±は+または-。

作成、修正日時をともに「日本時間で西暦 2021 年 1 月 2 日 12 時 34 分」とするのであれば、たとえば次のように記述できます。

```
<!-- PDFの作成、修正日時を指定する -->
<axf:document-info name="createdate" value="20210102T1234+09" />
<axf:document-info name="modifydate" value="20210102T1234+09" />
```

この FO の value には、同値の表現として 2021-01-02T12:34:00+09:00 令 03.01.02T12:34 という指定もできます。

西暦による表記でタイムゾーンを省略すると、AH XSL Formatter をインストールしているシステムのタイムゾーン（ローカルタイム）とみなされます。和暦による表記では常に日本時間（+09:00）とみなされます。存在しない日付、たとえば平成 40 年などを記述しても適当に補正されます。

modifydate が createdate よりも古いとき、createdate に揃えられます。不正な日付、不正な書式であると、出力はビルドを行った時刻に修正されます。

1970 年以前は西暦でも和暦でも無効な年となるため、たとえば N として明治を示す「明」や、YYYY に「1000」を指定しても無視され、出力された PDF には現在の日時が記録されます。

### 3.2.4 著作権情報を指定する

著作権についての情報を PDF に記述します。

著作権表示についての文書情報は PDF 1.4 以降で対応しています。

name	value
copyright-status	著作権のステータス。「不明」を示す Unknown、著作権保持を示す Copyrighted、著作権が発生しないまたは消滅を示す PublicDomain のいずれかを指定。
copyright-notice	著作権情報のテキスト。
copyright-info-url	著作権情報の URL

copyright-info-url には URL を指定しますが、処理時にアクセスは行われません。

### 3.2.5 文書情報を XMP ファイルで指定する

XMP 形式で PDF の文書情報を記述し、外部から読み込むことが可能です。

PDF 1.4 以降では、PDF の文書情報には Extensible Metadata Platform (XMP) という XML 仕様が含まれます。

XMP 形式で記述を行うには、<axf:document-info>要素で、name プロパティに xmp、value プロパティに XMP ファイルの URL を指定します。

```
<!-- PDFに適用するメタデータを含むXMPファイルを指定 -->
<fo:declarations>
  <axf:document-info name="xmp" value="common.xmp"/>
</fo:declarations>
```

PDF 出力時、AH XSL Formatter により、ほかの<axf:document-info>で指定した内容を優先したり、<pdf:Producer>や<xmp:CreatorTool>へ AH XSL Formatter を指定するなどの調整が行われます。

たとえば共通する文書情報を XMP ファイルとして分離することで、より効率的に文書が作成できるでしょう。

### XMP のパディングサイズを指定する

Option Property Name	xmp-padding
Value	<integer>
Initial	2000
Applies to	<pdf-settings>

オプション設定 xmp-padding プロパティへ、インプレースでの XMP 編集補助のためのパディングサイズを指定可能です。0 以下のとき、パディングされません。この指定があっても、xpacket end="r" (読み取りのみ) であるとき、パディングは付加されません。

外部ファイルとして XMP ファイルを指定したときだけでなく、PDF 出力時に自動生成される XMP メタデータに対しても有効です。

**XMP のパディング** XMP の構造中にパディングの領域を設けることで、埋め込まれた状態でも記述の編集や拡張が可能になります。

## 3.2.6 PDF を開いたときの PDF ビューアの表示を指定する

<axf:document-info>で指定可能な PDF の文書情報には、PDF ビューアでその PDF を開いたときの動作制御が含まれます。PDF を開いたときの動作であり、その後のユーザの操作には影響しません。

また、実際にその動作が行われるかについては PDF ビューアに依存します。

name	value
pagemode	ページモードの制御。アウトラインやサムネイル表示など。
pagelayout	ページレイアウトの制御。見開き表示や2列表示など。
hidetoolbar	ツールバー表示の制御。true のときツールバーを隠す。false では表示。
hidemenubar	メニューバー表示の制御。true のときメニューバーを隠す。false では表示。
hidewindowui	スクロールバーやナビゲーション用コントロールなどのユーザインターフェイス要素の表示制御。true のとき隠す。false では表示。
fitwindow	文書ウィンドウのサイズをページに適合するように変更するかを制御。true のとき変更する。false では変更しない。
centerwindow	文書ウィンドウを画面中央に配置するかを制御。true のとき中央に配置。false ではとくに制御しない。
displaydoctitle	文書タイトルの表示制御。true のとき表示。false では非表示。
openaction	表示される宛先、実行されるアクションの制御。internal-destination の<number-with-fragment>と同様の値を記述可能。指定可能な値に名前付きアクションや JavaScript を含む。

name="openaction"	意味
value="#Named=Print"	PDF を開いたときに印刷ダイアログを表示。
value=<number-with-fragment>	#123 や#page=123 によって PDF を開いたときに該当のページを表示するなどが可能。「リンク」の internal-destination に詳細。
value="#Named=LastPage"	PDF を開いたときに最終ページを表示。
value="#JavaScript=<任意の JavaScript プログラム>"	PDF を開いたときに指定した JavaScript を実行。PDF/A では不可。オプション設定 allow-javascript が false のとき、ここで指定した JavaScript は無視される。

## 3.2.7 カスタムプロパティを指定する

追加の PDF プロパティを指定します。

Element Name	<axf:custom-property>
Property	name, value
Parent	fo:declarations

<axf:document-info>要素の name プロパティにない項目を、<axf:custom-property>で埋め込み可能です。name と value プロパティへ<axf:document-info>と同様に指定します。

name プロパティの大文字小文字は区別され、Title、Author、Subject、Keywords、Creator、Producer、CreationDate、ModDate、Trapped は指定できません。

## 3.3 節 PDF 注釈

AH XSL Formatter では PDF 注釈を付加可能です。

PDF 注釈は PDF 仕様の一部で、通常のコンテンツとは別に編集可能なテキストなどをページ上に付加する機能です。種類や機能はさまざまで、`axf:annotation-*`で指定できるもののほかに、マルチメディアやリンクも PDF 注釈です。また、PDF 注釈はいくつかを組み合わせ利用可能です。

PDF のバージョンによって対応している機能が異なります。PDF 注釈の詳細については ISO 32000-1:2008・ISO 32000-2:2017 をご覧ください。

### 3.3.1 PDF 注釈の種類を指定する

PDF 注釈の種類を指定します。種類によって記述、指定できる内容が異なるため、種類の指定は必須です。

Property Name	<code>axf:annotation-type</code>
Value	<code>&lt;string&gt;</code>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

`axf:annotation-type` プロパティへ注釈の種類を指定します。指定できる注釈とその指定を次に示します。

<code>axf:annotation-type</code>	意味
Text	テキスト注釈。Acrobat では「ノート注釈」と表示。
FreeText	フリーテキスト注釈。Acrobat では「テキスト注釈」「テキストボックス注釈」「引き出し線付きテキストボックス注釈」と表示。
Stamp	ラバースタンプ注釈。
FileAttachment	ファイル添付注釈。

#### 注

**ポップアップ注釈** PDF 注釈は、大きくはマークアップ注釈とそのほかの種類に分けられます。AH XSL Formatter では、いくつかの種類のマークアップ注釈を `axf:annotation-*` プロパティによって明示的に指定可能です。ポップアップ注釈は、PDF ではマークアップ注釈の子として、マークアップ注釈の内容のテキストを表示したりする際に利用されます。マークアップ注釈の種類によっては、注釈の指定時に AH XSL Formatter によってポップアップ注釈も記述されます。

**フリーテキスト注釈** テキスト注釈は Acrobat 上では PDF に付箋を付加したように見えます。対し、フリーテキスト注釈は、PDF 上に直接書き込まれたように見える注釈です [図 3.3-1] (p.265)。

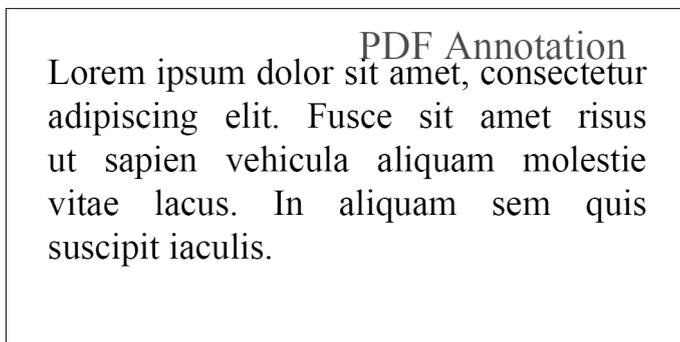


図 3.3-1 フリーテキスト注釈の例 (右上の「PDF Annotation」)

### 3.3.2 PDF 注釈のタイトルと作成者を記述する

PDF 注釈のタイトルと作成者情報を記述します。

Property Name	axf:annotation-title
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-author
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

PDF 注釈のタイトルを axf:annotation-title プロパティへ、作成者を axf:annotation-author プロパティへ記述可能です。

axf:annotation-title プロパティへの指定は PDF 1.5 以降で有効です。

### 3.3.3 PDF 注釈の作成日と修正日を記述する

PDF 注釈の作成日と修正日を指定します。

PDF 注釈では、その注釈の作成日と修正日の情報が、それぞれ注釈辞書の /CreationDate、/M の値として格納されます。

AH XSL Formatter では、これらの日付を指定可能です。

Property Name	axf:annotation-createdate, axf:annotation-modifydate
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

PDF 注釈を記述するとき、`axf:annotation-createdate` プロパティへ注釈の作成日を、`axf:annotation-modifydate` プロパティへ注釈の修正日を指定可能です。日付の書式は `<axf:document-info>` の作成日、修正日の指定と同様です。

`axf:annotation-modifydate` プロパティへ指定した日付が `axf:annotation-createdate` プロパティへ指定した日付よりも古いとき、`axf:annotation-createdate` プロパティの値へ修正されます。

これらのプロパティを省略したとき、`<axf:document-info>` の `name="modifydate"` へ指定した値が採用されます。

### 3.3.4 ファイル添付注釈を指定する

PDF へファイル添付注釈を指定します。

PDF では、注釈としてページ上の領域から添付したファイルへの参照を表現可能です。ファイル添付注釈を付加するオブジェクトへ `axf:annotation-type="FileAttachment"` を指定します。

Property Name	<code>axf:annotation-file-attachment</code>
Value	<code>&lt;uri-specification&gt;</code>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

`axf:annotation-file-attachment` プロパティへ添付するファイルの URI を指定します。

ファイル添付注釈は、ファイルへの参照として PDF 上にアイコンで表示されます。

### 3.3.5 PDF 注釈の背景色を指定する

背景を持つような PDF 注釈で背景色を指定します。

Property Name	<code>axf:annotation-color</code>
Value	<code>&lt;color&gt;</code>   none
Initial	none
Applies to	all block-level and inline-level formatting objects

PDF 注釈の背景などに利用される色を、`axf:annotation-color` プロパティへ指定します。

アイコンを持つような PDF 注釈では、PDF ビューアによってはアイコンの色にも反映されます。

### 3.3.6 フリーテキスト注釈のボーダーを修飾する

フリーテキスト注釈のボーダーはある程度修飾可能です。

Property Name	axf:annotation-border-color
Value	<color>
Initial	axf:annotation-text-color
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-border-style
Value	solid   dotted   dashed   dot-dash   dot-dot-dash
Initial	solid
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-border-width
Value	<border-width>
Initial	0pt
Applies to	all block-level and inline-level formatting objects

axf:annotation-border-color、axf:annotation-border-width、axf:annotation-border-style プロパティにはショートハンドはありません。axf:annotation-border-style プロパティで PDF 注釈の枠線修飾を指定可能ですが、指定できる枠線の種類に制限があります。border-style プロパティのように double や waveなどを指定しても無効となり、solid にフォールバックされます。

### 3.3.7 PDF 注釈のアイコンを指定する

PDF 注釈特有の修飾として、PDF 注釈のアイコンがあります。

Property Name	axf:annotation-icon-name
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

アイコンを axf:annotation-icon-name プロパティへ指定します。指定するアイコンの名前には任意の指定が可能ですが、対応するアイコンを表示できるかは PDF ビューアに依存します。「Acrobat Reader で表示できていたアイコンが Firefox では表示されない」ということも起こり得ます。注釈の種類によっても対応するアイコンの名前が異なります。

axf:annotation-type	axf:annotation-icon-name	空のとき
Text	Comment   Help   Insert   Key   NewParagraph   Note   Paragraph	Note
Stamp	Approved   AsIs   Confidential   Departmental   Draft   Experimental   Expired   Draft Final   ForComment   ForPublicRelease   NotApproved   NotForPublicRelease   Sold   TopSecret	
FileAttachment	Graph   PaperClip   PushPin   Tag	PushPin

たとえばテキスト注釈では、アイコンをクリックすると axf:annotation-\* プロパティで指定した内容やタイトルが表示されます。

### 3.3.8 フリーテキスト注釈を修飾する

フリーテキスト注釈の修飾を指定します。

Property Name	axf:annotation-text-color
Value	<color>
Initial	depends on user agent
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-font-family
Value	<string>
Initial	depends on user agent
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-font-size
Value	<absolute-size>   <relative-size>   <length>   <percentage>
Initial	depends on user agent
Applies to	all block-level and inline-level formatting objects
Percentages	refer to the font size
Property Name	axf:annotation-font-weight
Value	normal   bold
Initial	normal
Applies to	all block-level and inline-level formatting objects

Property Name	axf:annotation-font-style
Value	normal   italic
Initial	normal
Applies to	all block-level and inline-level formatting objects
Property Name	axf:annotation-text-align
Value	left   center   right
Initial	left
Applies to	all block-level and inline-level formatting objects

とくに指定しなければ PDF ビューアが設定したスタイルで表示されます。

### 3.3.9 ポップアップ、ラバースタンプ注釈のサイズを指定する

ポップアップ、ラバースタンプ注釈のサイズを指定します。

Property Name	axf:annotation-width, axf:annotation-height
Value	<length>   auto
Initial	auto
Applies to	all block-level and inline-level formatting objects

ポップアップ、ラバースタンプの幅と高さはそれぞれ `axf:annotation-width`、`axf:annotation-height` プロパティへ指定可能です。PDF 注釈は横組となるため、常に `axf:annotation-width` が水平方向、`axf:annotation-height` が垂直方向となります。そのほかの注釈ではこの指定は影響しません。

#### 注

横組の注釈を回転して配置することは可能です。

### 3.3.10 PDF 注釈の配置を指定する

PDF 注釈を記述したオブジェクトと注釈の表示位置を変更します。

ポップアップやアイコン、ラバースタンプ注釈の表示位置は、既定ではその領域の開始位置です。ブロックであればブロック開始位置となります。

Property Name	axf:annotation-position-horizontal, axf:annotation-position-vertical
Value	<length>
Initial	0pt
Applies to	all block-level and inline-level formatting objects

axf:annotation-position-\*プロパティによって、PDF 注釈の表示位置を領域開始位置から変更可能です。水平方向は axf:annotation-position-horizontal、垂直方向は axf:annotation-position-vertical プロパティへ指定します。

たとえば「段落のブロックに対して右端にポップアップを表示したい」というときには axf:annotation-position-horizontal に段落の行長分の長さを指定するとよいでしょう。

### 注

領域の長さが分からないとき、たとえば「ある段落の下端に表示したい」という制御は axf:annotation-position-vertical プロパティでは難しくなります。

axf:annotation-position-vertical="100vh"を指定したブロックが複数ページにわたるとき、PDF 注釈はページ内でそのブロックの開始位置から 100 vh に配置されます。

「表示位置さえ変えられればよい」ということであれば、注釈を付けたい領域の終了後に、keep-with-previous を指定した空の領域を作成し、その領域に注釈を付けることで実現が可能です。

## 3.3.11 出力 PDF の PDF 注釈に対し可能な操作を制御する

出力された PDF の PDF 注釈へ、権限を指定します。

Property Name	axf:annotation-flags
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

閲覧ユーザは PDF ビューアや編集ソフトで、PDF 注釈の内容を書き換えるなどの操作が可能です。これらの操作を許可するかどうかの制御を、axf:annotation-flags プロパティへ指定可能です。

ほか、注釈の印刷を可能にするかどうかなどの制御も可能です。

axf:annotation-flags	意味
Invisible	PDF 注釈を基本的には非表示にするが、標準の注釈に含まれるものかどうかで動作が変わる。
Hidden	ディスプレイ表示にも、印刷にも含まない。
NoView	ディスプレイ表示には含まないが、ほかの指定によっては印刷時の出力には含まれる。
ToggleNoView	イベント発火時（マウスクリックなど）のときに NoView を反転（非表示なら表示に、表示なら非表示に）させる。
Print	印刷時、出力に含まれる。
NoZoom	拡大縮小操作を制限する。
NoRotate	回転操作を制限する。
ReadOnly	ディスプレイ表示や印刷には影響しない。マウスクリックなどの操作に反応しなくなる。
Locked	PDF 注釈自体は操作できなくなる。フォームフィールドなど、内容の書き換えが必要な操作は許可する。
LockedContents	内容操作を制限する。

### 3.3.12 閲覧するときの PDF 注釈の開閉状態を指定する

PDF ビューアで PDF を開いたとき、その注釈が開いた状態で表示されるかを指定します。ポップアップ注釈など、開閉の状態がある PDF 注釈の初期状態を制御可能です。

Property Name	axf:annotation-open
Value	true   false
Initial	false
Applies to	all block-level and inline-level formatting objects

axf:annotation-open プロパティへ true を指定すると、出力した PDF を閲覧するとき、その PDF 注釈が表示された状態になります。

### 3.3.13 PDF 注釈のテキストを記述する

テキストを含むような PDF 注釈のテキストを記述します。

Property Name	axf:annotation-contents
Value	<string>
Initial	empty string
Applies to	all block-level and inline-level formatting objects

テキスト、フリーテキスト、ラバースタンプ、ファイル添付注釈の表示内容を axf:annotation-contents プロパティへ指定します。<fo:basic-link>によるリンクの、代替テキスト指定としても利用可能です。

axf:annotation-title、axf:annotation-author、axf:annotation-contents プロパティへ指定した内容は、PDF 出力時、ポップアップ注釈と組み合わせた形で表示されます。

### 3.3.14 PDF を埋め込むときに保持する PDF 注釈を指定する

Option Property Name	import-annotation-types
Value	<string>
Initial	empty string
Applies to	<pdf-settings>

#### import-annotation-types に指定可能な注釈の種類

All, Text, Link, FreeText, Line, Square, Circle, Polygon, PolyLine, Highlight, Underline, Squiggly, StrikeOut, Stamp, Caret, Ink, Popup, FileAttachment, Sound, Movie, Screen, 3D, RichMedia, Other

PDF を埋め込むとき、保持する注釈の種類を、オプション設定 import-annotation-types プロパティへ指定可能です。空白区切りで複数種類の指定が可能です。All はすべての注釈、Other はここに記載がない種類の注釈を埋め込むときに指定します。

#### 注

埋め込みを保持できる PDF 注釈の種類は、AH XSL Formatter で PDF を出力するときに指定可能なものとは異なります。

### 3.3.15 ユースケース：表組に元データを添付する

表組へファイル添付注釈を付加し、注釈のサイズと位置、メタデータを指定します。

PDF 注釈のサイズを指定しないときそのサイズは注釈を指定したオブジェクトの領域に従います。たとえば<fo:table>へ指定すると、その表全体にファイル添付注釈を示すアイコンが表示されるでしょう。

表組に元データをファイル添付注釈として付加する例を示します。

#### ファイル添付注釈を付加する

表組へ付加する注釈の種類と、ファイル添付注釈、アイコンを指定します。

```
<!-- 表組にファイル添付注釈を付加する -->
<fo:table-and-caption ...
  axf:annotation-type="FileAttachment"
  axf:annotation-file-attachment="url(./table.xlsx)"
  axf:annotation-icon-name="PushPin" ...>
  <fo:table-caption>
    <fo:block... >表</fo:block>
```

```
</fo:table-caption>
<fo:table ...>
  ...
</fo:table>
</fo:table-and-caption ...>
```

## PDF 注釈の位置とサイズを指定する

表組のキャプション開始行の右端に、縦横 1cm でファイル添付注釈のアイコンを付加します。

```
<!-- PDF注釈の位置とサイズを指定する -->
<fo:table-and-caption ...
  axf:annotation-width="1cm" axf:annotation-height="1cm"
  axf:annotation-position-horizontal="100vw - 1cm">
  ...
</fo:table-and-caption>
```

先ほどファイル添付注釈の付加を指定したオブジェクトへ、PDF 注釈の幅、高さ、位置を指定します。オブジェクトの開始位置が始点となるため、右端はたとえば 100vw のように指定が可能です。上の FO では注釈の幅の分の長さ 1cm を引いて、表組の領域に収まるように指定しました。

## 3.4 節 リンク制御

PDF でのリンクはリンク注釈という注釈の形で存在します。リンクがどのように開かれるかなどを指定します。

`<fo:basic-link>`要素で囲われた範囲がリンクの領域とみなされます。

Web ページへのリンクなど、出力する文書外へのリンクは `external-destination` プロパティへ URL を記述し、文書内部へのリンクは `internal-destination` プロパティへ記述します。

AH XSL Formatter では、PDF 出力のとき、同一の PDF 内にリンク先があることを想定していません。

### 3.4.1 外部リンクの動作を指定する

外部リンクがどのように開かれるかを指定します。

Property Name	<code>axf:action-type</code>
Value	<code>auto</code>   <code>gotor</code>   <code>launch</code>   <code>uri</code>
Initial	<code>auto</code>
Applies to	<code>fo:basic-link</code>

`<fo:basic-link>`で外部リンクを指定するとき、その動作を `axf:action-type` プロパティで制御可能です。

`<fo:basic-link>`に記述するとき、`axf:action-type` にはリンクに関連した値のみ指定可能です。

<code>axf:action-type</code>	動作
<code>gotor</code>	PDF として開く。リンク先の URI は PDF とみなされる。
<code>launch</code>	ファイルとして開く。
<code>uri</code>	URI として開く。

Option Property Name	<code>use-launch-for-local-file</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>&lt;pdf-settings&gt;</code>

`axf:action-type="auto"`のとき、リンクの動作はオプション設定 `use-launch-for-local-file` プロパティへの指定によって決定されます。`use-launch-for-local-file="true"`のとき、PDF のリンクプロパティの動作を「ファイルを開く」と変換し、`false` のとき「World Wide Web リンク」として変換します。

### 3.4.2 リンク先になるときの開かれ方（宛先形式）を指定する

出力した PDF のページがリンク先として開かれるとき、どのように表示されるか（宛先形式）を指定します。倍率や位置が制御対象となります。

宛先形式を、位置指定と倍率で指定するとき、PDF では次のように表記します。page は宛先形式を適用するページを示し、画面上の位置を左端から left で、上端から top で、また表示倍率を zoom で、それぞれパラメータを指定します。zoom は%値を指定します。無指定（null）のとき、既定の動作に従います。

[page /XYZ left top zoom]

宛先形式を、PDF ビューアのどこを基準として表示サイズを調整するかで指定するとき、PDF では次のように表記します。/Fit のほか、/FitH や/FitV など、パラメータを取って位置を調整する指定も存在します。

[page /Fit]

Property Name	axf:destination-type
Value	<string>
Initial	empty string
Applies to	block level formatting object

宛先形式を axf:destination-type プロパティへ指定します。宛先形式は、指定のないときや無効な文字列のとき axf:destination-type="xyz-top"とみなされます。大文字と小文字は区別されません。

表示位置の左端からの距離と上端からの距離は AH XSL Formatter が計算します。xyz\*の指定で、これらの null と非 null を明示可能です。またこのとき、倍率 zoom も指定しなければ null とみなされます。

実際に開かれるときに有効であるかは PDF ビューアに依存します。

axf:destination-type	意味	PDF の宛先形式
xyz <number>?	倍率を<number>で指定。left と top を null とする。	[page /XYZ null null zoom]
xyz-top   <number>	倍率を<number>で指定。left を null とする。	[page /XYZ null top zoom]
<number>	倍率を<number>で指定。left を null とする。	[page /XYZ null top zoom]
xyz-left <number>?	倍率を<number>で指定。top を null とする。	[page /XYZ left null zoom]
xyz-left-top <number>?	倍率を<number>で指定。	[page /XYZ left top zoom]

axf:destination-type	意味	PDF の宛先形式
fit	ページの垂直、水平方向ともにページのコンテンツ全体がフィットする倍率で表示。幅と高さが異なるときは小さい倍率になるように合わせ、ウインドウ中央に表示。	[page /Fit]
fitv	縦方向の端がウインドウにフィットする倍率で表示。水平方向位置は AH XSL Formatter が計算する。	[page /FitV left]
fitH	内部リンクでの指定のみ有効。縦の端がウインドウにフィットする倍率で表示。垂直方向位置は AH XSL Formatter が計算する。	[page /FitH top]
fitr	内部リンクでの指定のみ有効。ウインドウ内の領域に表示。	[page /FitR left bottom right top]
fitb	ページの境界ボックスの幅全体がウインドウにフィットするように表示。	[page /FitB]
fitbh	内部リンクでの指定のみ有効。ページの境界ボックスの縦の端がウインドウにフィットするよう表示。	[page /FitBH top]
fitbv	ページの境界ボックスの横の端がウインドウにフィットするよう表示。	[page /FitBV left]

### 3.4.3 ユースケース：フォームフィールドをリンクにする

フォームフィールドによるボタンなどを、リンクのトリガとして扱うことが可能です。

```
<!-- プッシュボタンによるリンク -->
<axf:form-field field-type="button" ...
  action-type="uri"
  external-destination="https://www.antenna.co.jp" />
```

## 3.5 節 多分冊出力制御

文書の内容を分割し複数の PDF ファイルを出力可能です。分冊箇所やそのときのしおり情報、文書情報などを制御します。

ひとつの文書の内容を分割して出力するとき、考慮しなければならない点があります。

分割された内容に共通する文書情報、個別の文書情報、ページ数、分割されたそれぞれのファイル名、分割された先への相互参照などです。

また、印刷される目次とは別に、PDF にはしおりを設定できます。このしおりは通常、`<fo:bookmark>`によって作成が可能です。

AH XSL Formatter には複数ファイルに PDF を分割して出力する多分冊出力機能があります。`<fo:bookmark>`ではなく、AH XSL Formatter の拡張プロパティを利用します。多分冊出力を行うとき、`<fo:bookmark>`によるしおりは無効です。

分冊の指定は、PDF ファイル出力時に機能します。印刷やストリーム出力では機能しません。

Element Name	<code>&lt;afx:output-volume-info&gt;</code>
Property	<code>afx:initial-volume-number</code> , <code>afx:document-info-include</code> , <code>format</code>
Parent	<code>fo:declarations</code>

`<afx:output-volume-info>`はページシーケンスを単位として分冊出力を行うとき、その制御を記述するための要素です。

### 3.5.1 分冊箇所を指定する

内容を分割し分冊出力にする箇所をページシーケンス単位で指定します。

Property Name	<code>afx:output-volume-break</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>fo:page-sequence</code>

`afx:output-volume-break` プロパティへ `true` を指定した`<fo:page-sequence>`を単位として分冊されます。分冊を行うとき、最初の`<fo:page-sequence>`は常に `true` とみなされます。

### 3.5.2 分冊出力される PDF のファイル名を指定する

出力される PDF のファイル名を個別に指定します。

分冊は先頭の`<fo:page-sequence>`または `afx:output-volume-break="true"`が指定された`<fo:page-sequence>`で行われます。

Property Name	axf:output-volume-filename
Value	<string>
Initial	empty string
Applies to	fo:page-sequence

分冊される PDF のファイル名を axf:output-volume-filename プロパティへ指定可能です。

### 3.5.3 分冊での文書情報を一括で指定する・個別に指定する

作者名や PDF ビューアでの開き方指定などの文書情報の扱いを指定します。

Property Name	axf:document-info-include
Value	first   all
Initial	first
Applies to	axf:output-volume-info

<axf:document-info>へ指定する PDF の文書情報を、分冊時にどう扱うかを axf:document-info-include プロパティへ指定します。

first では第 1 分冊のみに文書情報を付加しますが、all を指定することで全分冊に文書情報を付加します。

### 3.5.4 分冊の開始番号を指定する

分冊出力時に参照される分冊番号の開始番号を指定します。

Property Name	axf:initial-volume-number
Value	<number>
Initial	1
Applies to	axf:output-volume-info

分冊の開始番号を axf:initial-volume-number プロパティへ指定可能です。

axf:output-volume-filename プロパティが無指定のとき、この番号は分冊出力された PDF のファイル名に接尾辞として付加されます。接尾辞の書式は<axf:output-volume-info>の format プロパティへ指定します。format プロパティへ指定可能な書式は XSL 1.1 仕様と同様です。

### 3.5.5 分冊でのしおり情報を記述する

FO の内容を分割し複数の PDF ファイルを出力するとき、しおり情報や文書情報を制御します。

Property Name	axf:outline-title
Value	<string>
Initial	empty string
Applies to	block-level formatting objects
Property Name	axf:outline-level
Value	<number>
Initial	0
Applies to	block-level formatting objects
Property Name	axf:outline-group
Value	<string>
Initial	empty string
Applies to	block-level formatting objects

しおりのタイトルになるテキストを axf:outline-title プロパティへ記述します。省略すると、このプロパティが付加されているオブジェクトのテキストがしおりのタイトルとなります。テキストの修飾は axf:outline-color プロパティなどで制御します。

<fo:bookmark>では<fo:bookmark-tree>の子要素として入れ子で記述することにより階層構造を記述します。一方分冊で利用する axf:outline-\*では、プロパティで階層を指定します。axf:outline-level は 0 以上の整数でしおり項目の階層を指定します。初期値の 0 ではしおりは作成されません。最上位が 1 で、階層が深くなる度に数字を増やします。

axf:outline-group プロパティは、指定したテキストが同じものをグループとしてまとめて出力します。axf:bookmark-include="separate-group"のとき、すべての分冊にグループ化されたしおりも付加されます。axf:bookmark-include="separate"のとき、最後の分冊にグループ化されたしおりが付加されます。

## 分冊へのしおり付加を制御する

Property Name	axf:bookmark-include
Value	first   all   separate   separate-group
Initial	separate
Applies to	axf:output-volume-info

axf:bookmark-include	意味
first	第1分冊にすべての分冊のしおり情報を付加する。
all	すべての分冊にすべての分冊のしおり情報を付ける。
separate	それぞれの分冊にそれぞれのしおりを付ける。
separate-group	separateに加えてグループ化されたしおりもそれぞれの分冊に付加する。

axf:bookmark-include="separate"のとき、しおりは axf:outline-level="1"が現れた分冊に属します。複数の分冊にまたがるとき、先に登場する分冊に属します。

axf:outline-group プロパティによってグループ化されたしおりは最後の分冊にまとめて付加されます。

次の FO では、「見出し bunsatsu」のしおりは、bunsatsu2.pdf として出力される PDF に付加されます。

```

<!-- 「見出しbunsatsu」のしおりをbunsatsu2.pdfに付加する -->
<fo:page-sequence axf:output-volume-break="true"
  axf:output-volume-filename="bunsatsu2.pdf">
  <fo:flow ... >
    ...
    <fo:block axf:outline-level="1"
      axf:outline-title="見出しbunsastu">
      見出しbunsastu
    </fo:block>...
  </fo:flow ... >
</fo:page-sequence>

```

### 3.5.6 多分冊でしおりに外部リンク、内部リンクを指定する

多分冊出力で、しおりにリンクを指定します。

Property Name	axf:outline-external-destination
Value	<uri-specification>
Initial	empty string
Applies to	block-level formatting objects
Property Name	axf:outline-internal-destination
Value	<ref-id>   <number-with-fragment>
Initial	empty string
Applies to	block-level formatting objects

外部リンクは axf:outline-external-destination プロパティへ <uri-specification> を指定します。

内部リンクは `axf:outline-internal-destination` プロパティへ `<ref-id>` またはページ番号の数字と「#」による指定子を組み合わせた `<number-with-fragment>` を指定します。

### 3.5.7 分冊されたしおりのテキストを修飾する

分冊で、しおりのテキストの色や太さを修飾します。

ひとつの PDF ファイルでの出力であれば `<fo:bookmark>` の子要素として `<fo:bookmark-title>` を記述するとき、`color` プロパティなどでしおりのテキストを修飾可能です。分冊出力するしおりのテキストは、拡張プロパティで修飾します。

Property Name	<code>axf:outline-color</code>
Value	<code>&lt;color&gt;</code>
Initial	<code>transparent</code>
Applies to	block-level formatting objects

Property Name	<code>axf:outline-font-style</code>
Value	<code>normal</code>   <code>italic</code>
Initial	<code>normal</code>
Applies to	block-level formatting objects

Property Name	<code>axf:outline-font-weight</code>
Value	<code>normal</code>   <code>bold</code>
Initial	<code>normal</code>
Applies to	block-level formatting objects

`axf:outline-font-style` プロパティは斜体かどうか、`axf:outline-font-weight` プロパティは太字かどうかのみ指定可能です。PDF 1.4 以降に対して有効です。

### 3.5.8 分冊でのしおりの展開を制御する

分冊出力の PDF を PDF ビューアで開いたとき、階層構造を持つしおりの下位項目を展開した状態にするかどうかを指定します。

Property Name	<code>axf:outline-expand</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	block-level formatting objects

`axf:outline-expand` プロパティへ `true` を指定したしおりの下位項目は、対応した PDF ビューアで開いたとき、展開して表示されます。

## 3.5.9 ユースケース：多分冊用の目次、索引を作成する

多分冊の PDF を作成するときの目次の例を作成します。  
上巻、下巻、索引として分冊しました [図 3.5-1] (p.284)。

### 分冊を指定する

```
<!-- 分冊を指定する -->
<fo:declarations>
  <axf:output-volume-info
    initial-volume-number="1" format="上下索"
    bookmark-include="separate" />
</fo:declarations>
```

format はトークン、シンボル以外の文字列のときまですべての字を一度ずつ使用するので、「上」「下」「索」が出力された PDF のファイル名末尾に付加されます。テキストは 1 字ずつ使用されるため、「上下索引」としても「索引」をファイル名末尾に付けることはできません。

axf:bookmark-include="separate"により、上巻は上巻のしおり、下巻は下巻のしおりを付加します。

3 分冊程度であれば、axf:output-volume-filename で個別にファイル名を指定するのもよいでしょう。

### しおりを指定する

```
<!-- 分冊でのしおりを指定する -->
<fo:block-container ... axf:outline-level="1"
  axf:outline-title="OSDC V7.1 マニュアル 上巻">
  <fo:block ...
    >Office Server Document Converter V7.1 マニュアル</fo:block>
  <fo:block ...>- 上巻 -</fo:block>
  <fo:block ...>インターフェイス編</fo:block>
</fo:block-container>
...

<fo:block ... axf:outline-level="2"
  letter-spacing="0.4em">上巻目次</fo:block>
...
```

分冊出力のとき、しおりは axf:outline-\* プロパティで指定します。

下の「上巻目次」は axf:outline-title プロパティを未指定なので、しおりのタイトルには「上巻目次」が使用されます。

表示の修飾として見出しの字間を空けるときの直接スペースなどを挿入してしまうと、しおりの表示にもスペースが挿入されてしまいます。axf:outline-title プロパティを別に指定することでも防げますが、「表示上の字間を空ける」目的であれば letter-spacing プロパティなどが有効です。

## 分冊の接頭辞を指定する

```
<!-- 上巻 -->
<fo:page-sequence master-reference="PageMaster">
  <fo:folio-prefix>上</fo:folio-prefix>
  <fo:static-content flow-name="xsl-region-after">
    <fo:block ... text-align="center">
      <fo:inline>- </fo:inline>
      <fo:page-number axf:suppress-folio-prefix="true" />
      <fo:inline> -</fo:inline>
    </fo:block>
  </fo:static-content>
  ...
</fo:page-sequence>
...
<!-- 下巻 -->
<fo:page-sequence master-reference="PageMaster">
  ...
  <fo:folio-prefix>下</fo:folio-prefix>
  ...
</fo:page-sequence>
```

`<fo:folio-prefix>`でページ番号の接頭辞を指定しています。

分冊するとき、分冊ごとにページ番号へ接頭辞や接尾辞を付けておくとページ番号参照のときに有効です。しかし、上巻を読んでいるときにはそのページが上巻であることは明らかですから、`region-after`に表示するページ番号では `axf:suppress-folio-prefix` を指定し、接頭辞の表示を抑制しています。

## 目次のレイアウトを指定する

```
<!-- 目次のレイアウト -->
<fo:block ...>上巻目次</fo:block>
<fo:block ...><axf:tab tab-align="')'"
  />1) コマンドラインインターフェイス
  <fo:leader leader-pattern="dots" leader-length.optimum="100%"/>
  <fo:page-number-citation ref-id="commandIF"
    axf:suppress-folio-prefix="true" />
</fo:block>
```

見出しとして「上巻目次」とあるので、ページ番号表示にも「上」と表示するのは冗長として、ここでも `axf:suppress-folio-prefix="true"` でページ番号参照の接頭辞表示を抑制しています。

<afx:tab tab-align="" />や<fo:leader>の指定は数字表示で桁数が増えたときの揃えを行うための指定です。

## 索引のレイアウトを指定する

```
<!-- 索引のレイアウト -->
<fo:block ... >索引</fo:block>
<fo:block-container column-count="2" column-gap="10mm">
  <fo:block text-align-last="justify" space-before="0.5em"
  >.NETインターフェイス<fo:leader leader-pattern="dots" />
    <fo:page-number-citation ref-id="dotnetIF" />
  </fo:block>
  ...
</fo:block-container>
```

索引でのページ参照は上巻、下巻どちらのページかわかるように、接頭辞を表示しました。  
<fo:block-container>に column-count="2"を指定し、索引項目を段組にしています。

この例では数ページ程度なので項目あたり1行ですが、多分冊を行う文書では連続したページ番号参照、桁数の多いページ番号参照など、考慮すべき事項が多くあります。

1) コマンドラインインターフェイス
<small>コマンドラインインターフェイスは単独の実行形式ファイルとして提供されます。Windows のバッチファイルや Linux のシェルスクリプトなどから呼び出すことができます。 Office Server Document Converter V7.1 のコマンドラインインターフェイスは、ファイルを指定して動作させることもできますが、標準入力からファイルを読み、標準出力へ PDF / SVG を書くフィルタとして動作させることができます。</small>
2) .NET インターフェイス
<small>.NET インターフェイスを使えば、Visual Basic.NET、Visual C#.NET などで作成された .NET アプリケーションから Office Server Document Converter V7.1 の機能を利用することができます。</small>
3) COM インターフェイス
<small>COM インターフェイスを使えば、Visual Basic、Delphi、VBScript などの COM をサポートしている言語で作成したアプリケーションから Office Server Document Converter V7.1 の機能を利用することができます。 COM インターフェイスを利用するためには、Windows への登録が必要です。正しくインストールされている場合は、COM の登録は行われています。</small>
- 3 -

上巻のページ

下巻目次
6) PDF 出力..... 3
7) SVG 出力..... 3
8) XPS 出力..... 3
9) INX 出力..... 4
10) Flash 出力..... 4
11) イメージ出力..... 4

下巻の目次ページ

索引
.NET インターフェイス..... 上 3
C/C++ インターフェイス..... 上 4
COM インターフェイス..... 上 3
Flash 出力..... 下 4
INX 出力..... 下 4
Java インターフェイス..... 上 4
PDF 出力..... 下 3
SVG 出力..... 下 3
XPS 出力..... 下 3
イメージ出力..... 下 4
コマンドラインインターフェイス..... 上 3

索引ページ

図 3.5-1 多分冊用の目次、索引を作成する

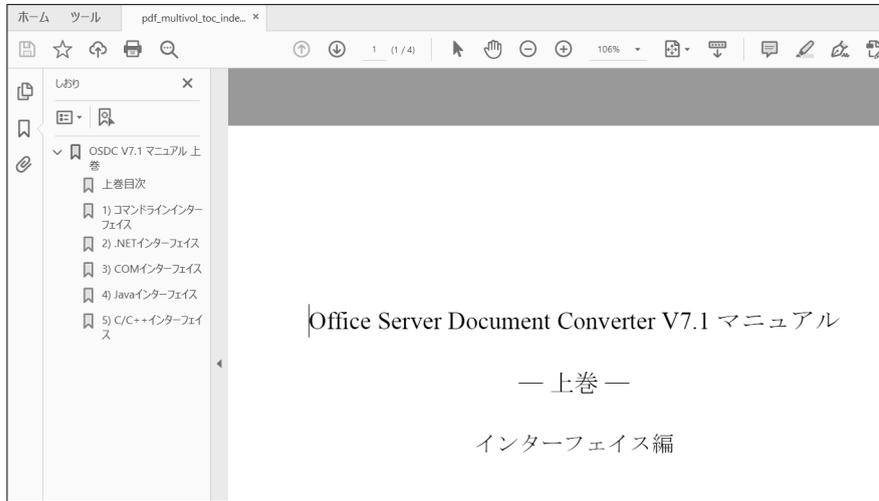


図 3.5-2 分冊ごとのしおりを付加する

## 3.6 節 マルチメディアの埋め込み

PDF には動画や音声などのメディアを埋め込むことが可能です。他にも、3DCG などを埋め込むことが可能です。

動画などのメディアの埋め込みには 2 種類あります。マルチメディア注釈としての埋め込みと、リッチメディア注釈としての埋め込みです。

マルチメディア注釈の埋め込みは PDF/X、PDF/A を除いた PDF 1.5 以降で指定可能です。ただし、PDF 2.0 以降ではマルチメディア注釈の埋め込みは非推奨です。background-image プロパティには指定できません。

リッチメディア注釈の埋め込みは PDF 1.7 以降で指定可能です。

### 3.6.1 埋め込むメディア形式を明示する・埋め込みを制御する

マルチメディアは画像と同様の指定で埋め込めますが、埋め込まずにリンクとしたり、AH XSL Formatter が自動判別できないときにメディア形式を明示的に指定します。

#### マルチメディアの埋め込みを制御する

Property Name	axf:multimedia-treatment
Value	auto   embed   absolute-link   relative-link   richmedia   richmedia-windowed
Initial	auto
Applies to	fo:external-graphic / multimedia objects
Inherited	yes
Option Property Name	multimedia-treatment
Value	embed   absolute-link   relative-link   richmedia   richmedia-windowed
Initial	embed
Applies to	<pdf-settings>
Option Element Name	<multimedia>
Property	audio, video, flash
Parent	<formatter-settings>

axf:media-treatment	意味
auto	オプション設定 multimedia-treatment で制御する。
embed	マルチメディアを埋め込む。
absolute-link	埋め込まず、絶対パス参照のリンクをする。
relative-link	埋め込まず、相対パス参照のリンクをする。
richmedia	リッチメディア注釈として埋め込む。
richmedia-windowed	リッチメディア注釈として埋め込む。再生は別ウィンドウを表示させる。

## ✔ ヒント

axf:media-treatment="richmedia-windowed"は、「動画を埋め込むが PDF のページ上に十分な大きさの動画サイズが確保できない」ときなどに有用です。

## メディアの形式を明示的に指定する

マルチメディアの埋め込みでは content-type の指定は必須です。AH XSL Formatter では content-type の記述が audio/\* のときオーディオ、video/\* のときビデオ、application/x-shockwave-flash のとき Flash とみなします。これら以外の content-type を指定するとき、AH XSL Formatter は指定されたコンテンツがマルチメディアかどうか自動的に判別できないことがあります。AH XSL Formatter が判別できない content-type のマルチメディアを指定するとき、オプション設定の <multimedia> に、content-type が audio、video、flash のどれに属するかを記述します。

```
<!-- オプション設定でcontent-typeを追加する -->
<formatter-settings>
  ...
  <multimedia audio="something/audiolike" />
</formatter-settings>
```

実際に再生できるかどうかは PDF ビューアに依存します。埋め込める代表的なマルチメディアの形式を次に示します。

拡張子	content-type
*.3g2	video/x-msvideo
*.3gp	video/x-msvideo
*.f4v	video/mp4
*.flv	video/x-msvideo
*.m4v	video/mp4

拡張子	content-type
*.mov	video/quicktime
*.mp3	audio/x-mp3
*.mp4	video/mp4
*.swf	application/x-shockwave-flash

### 3.6.2 代替のポスター画像を指定する

指定したメディアが再生できない環境などで代替表示されるポスター画像を指定します。

Property Name	axf:poster-image
Value	<uri-specification>   none   auto
Initial	auto
Applies to	fo:external-graphic / multimedia objects

マルチメディアの埋め込みを行うとき、動画メディアであれば非再生時のプレースホルダ、音声メディアでも参照領域を表すためにポスター画像が指定可能です。axf:poster-image プロパティへ、ポスター画像を通常の画像と同様に指定します。auto では AH XSL Formatter で用意したフォールバック画像を挿入します。none ではポスター画像を使用しません。

Property Name	axf:poster-content-type
Value	<string>   auto
Initial	auto
Applies to	fo:external-graphic / multimedia objects

ポスター画像の content-type を明示するときは axf:poster-content-type に content-type の文字列を指定します。auto ではポスター画像から AH XSL Formatter が判断し content-type を設定します。

### 3.6.3 メディア再生時の音量を制御する

メディア再生時の音量について、割合や数値で指定します。

Property Name	axf:media-volume
Value	<percentage>   <number>   auto
Initial	auto
Applies to	fo:external-graphic / multimedia objects

メディア再生時の音量を、axf:media-volume プロパティへ指定可能です。マルチメディア注釈、リッチメディア注釈ともに有効です。実際に指定した音量が適用されるかどうかは PDF ビューア、プレイヤーに依存します。

### 3.6.4 マルチメディア注釈の再生を制御する

PDF ビューアでのマルチメディア注釈の再生制御について指定します。

指定が有効になるかどうかは、メディアデータ、PDF ビューア、プレイヤなどに依存します。

Property Name	axf:show-controls
Value	true   false
Initial	false
Applies to	fo:external-graphic / multimedia objects

マルチメディア注釈の再生コントロールバーの表示を、axf:show-controls プロパティへの指定で制御可能です。true のとき、再生コントロールバーを表示します。コントロールバーは再生中のマルチメディア注釈の下側に表示されるため、他の要素と重ならないようにするにはマルチメディア要素の下側に十分スペースを確保してください。

Property Name	axf:media-play-mode
Value	once   continuously   <number>
Initial	once
Applies to	fo:external-graphic / multimedia objects

Property Name	axf:media-duration
Value	intrinsic   infinity   <number>
Initial	intrinsic
Applies to	fo:external-graphic / multimedia objects

axf:media-play-mode プロパティへの指定でマルチメディア注釈の再生回数を制御可能です。once で一度のみ、continuously で繰り返し再生、<number>で指定回数を再生します。

axf:media-duration プロパティへマルチメディア注釈の再生時間を指定します。

Property Name	axf:media-extraction-policy
Value	tempnever   tempextract   tempaccess   tempalways
Initial	tempaccess
Applies to	fo:external-graphic / multimedia objects

マルチメディア注釈の再生時、一時ファイルの作成を許可するかの制御を axf:media-extraction-policy プロパティへ指定可能です。

axf:media-extraction-policy 一時ファイルの作成	
tempnever	常に不許可。
tempextract	ユーザーアクセス権に「内容のコピー」が与えられているとき許可。
tempaccess	ユーザーアクセス権に「アクセシビリティのための内容抽出」が与えられているとき許可。
tempalways	常に許可。

### 3.6.5 リッチメディア注釈で別ウィンドウのサイズを指定する

リッチメディア注釈は別ウィンドウで表示するように指定が可能です。このウィンドウのサイズを指定します。

`axf:media-treatment="richmedia-windowed"`を指定したとき、リッチメディア注釈を開く別ウィンドウのサイズは自動で決定されます。このサイズは明示的な指定が可能です。

Property Name	<code>axf:media-window-width</code> , <code>axf:media-window-height</code>
Value	<code>auto</code>   <code>&lt;length&gt;</code>
Initial	<code>auto</code>
Applies to	<code>fo:external-graphic</code> / <code>multimedia objects</code>

`axf:media-window-width` プロパティにウィンドウの幅、`axf:media-window-height` プロパティにウィンドウの高さを指定可能です。

### 3.6.6 リッチメディア注釈のスキンを制御する

リッチメディア注釈は PDF ビューア側ではスキンを介して再生を制御します。このスキンの制御を指定します。

Property Name	<code>axf:media-skin-control</code>
Value	<code>none</code>   <code>all</code>   [ <code>play</code>    <code>stop</code>    <code>forward</code>    <code>rewind</code>    <code>seek</code>    <code>mute</code>    <code>volume</code> ]
Initial	<code>all</code>
Applies to	<code>fo:external-graphic</code> / <code>multimedia objects</code>

Property Name	<code>axf:media-skin-auto-hide</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>fo:external-graphic</code> / <code>multimedia objects</code>

リッチメディア注釈のスキンに表示するコントロール項目を `axf:media-skin-control` プロパティで制御可能です。

リッチメディア注釈が有効化されていないとき、スキンを自動的に隠すかどうかを `axf:media-skin-auto-hide` で制御可能です。true で自動的に隠されるようになります。

Property Name	<code>axf:media-skin-color</code>
Value	<code>auto</code>   <code>&lt;color&gt;</code>
Initial	<code>auto</code>
Applies to	<code>fo:external-graphic</code> / multimedia objects

スキンの色は `axf:media-skin-color` で制御可能です。無効な指定をしたときは既定の値が採用されます。

### 3.6.7 リッチメディア注釈の有効化タイミングを制御する

注釈のページを表示したときに動画の再生を開始するなど、リッチメディア注釈をどのタイミングで有効化、無効化するかを制御可能です。

Property Name	<code>axf:media-activation</code>
Value	[ <code>ExplicitlyActivated</code>   <code>PageOpen</code>   <code>PageVisible</code> ]? [ <code>ExplicitlyDeactivated</code>   <code>PageInvisible</code>   <code>PageClose</code> ]?
Initial	<code>empty</code>
Applies to	<code>fo:external-graphic</code> / multimedia objects

リッチメディア注釈の有効化と無効化のタイミングを、`axf:media-activation` プロパティへ指定します。それぞれ空白区切りで指定します。それぞれの値は、次の表にあるように2字での略記も可能です。

<code>axf:media-activation</code>	タイミング	有効、無効
<code>ExplicitlyActivated, XA</code>	コンテンツをクリックしたとき	有効
<code>PageOpen, PO</code>	コンテンツを含むページを開いたとき	有効
<code>PageVisible, PV</code>	コンテンツを含むページが表示されているとき	有効
<code>ExplicitlyDeactivated, XD</code>	コンテキストメニューから[コンテンツを非アクティブにする]を選択したとき	無効
<code>PageClose, PC</code>	コンテンツを含むページを閉じたとき	無効
<code>PageInvisible, PI</code>	コンテンツを含むページを表示していないとき	無効

大文字と小文字は区別されず、無指定のとき (`axf:media-activation=""`) はそれぞれ `ExplicitlyActivated` `ExplicitlyDeactivated` とみなされます。

## 3.6.8 Flash メディアを制御する

リッチメディア注釈として埋め込む Flash メディアについて、メニューやプレーヤーに渡す変数、表示を制御します。

AH XSL Formatter ではリッチメディア注釈として PDF への Flash の埋め込みをサポートしています。Flash メディアの埋め込みは通常のリッチメディア注釈同様、`<fo:external-graphic>`などへ指定することで埋め込みが可能です。

### ⚠ 注意事項

アドビシステムズ社は 2020 年末に Flash メディアのサポートを終了しました。Flash を埋め込むときは閲覧環境があるか事前に確認してください。

Property Name	<code>axf:media-flash-context-menu</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>fo:external-graphic</code> / multimedia objects

Flash メディアを埋め込むとき、Flash メディアのコンテキストメニューの表示を `axf:media-flash-context-menu` プロパティで制御可能です。true のとき表示します。

Property Name	<code>axf:media-flash-vars</code>
Value	<code>&lt;string&gt;</code>
Initial	empty string
Applies to	<code>fo:external-graphic</code> / multimedia objects

Flash メディアを埋め込むときの変数を `axf:media-flash-vars` プロパティへ記述します。AH XSL Formatter はテキストの検証を行いません。埋め込んだ変数はそのまま PDF へ出力されます。

Property Name	<code>axf:media-transparent-background</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>true</code>
Applies to	<code>fo:external-graphic</code> / multimedia objects

Flash メディアを埋め込むとき、背景を透明にするかを `axf:media-transparent-background` プロパティへ指定します。true のとき、Flash メディアの背景が透明になります。

### 3.6.9 ユースケース：メディア代替のポスター画像を印刷する

マルチメディアは注釈として PDF に埋め込まれます。注釈のフラグ制御で印刷を許可すると、ポスター画像が印刷されるようになります。

```
<!-- 動画の指定とポスター画像、注釈の印刷指定 -->
<fo:block space-after="1em">
  <fo:external-graphic ... src="Wildlife.mpg"
    content-type="video/mpeg"
    axf:poster-image="Wildlife_0001.jpg"
    axf:poster-content-type="image/jpeg"
    axf:annotation-flags="Print" />
</fo:block>
```

## 3.7 節 PDF 対話フォーム (AcroForm)

PDF 対話フォームのある PDF を作成可能です。

PDF 対話フォームは、PC 上で入力可能な帳票 PDF の作成に有用です。また、電子署名の付加や、入力した値を JavaScript で加工するなど、さまざまに利用可能です。

Element Name	<axf:form>
Property	external-destination, axf:field-submit-method, axf:field-submit-coordinates
Parent	page area objects
Element Name	<axf:form-field>
Property	axf:field-type
Parent	axf:form, page area objects

PDF 対話フォームは、テキストやチェックボックスなど値の入力を求めるフォームフィールドと、フォームフィールドに入力された値をどうするか決定するフォームアクションで構成されます。

<axf:form-field>によって作成されたフォームフィールドへ入力された値を、axf:action-type に submit や reset を指定して操作するのであれば、<axf:form-field>を<axf:form>の子孫の要素として記述します。

<axf:form-field>はインライン要素です。width と height でサイズを指定します。

### 注

フォームは GUI のプレビューでは表示されません。

### 3.7.1 フォームフィールドを作成する

閲覧ユーザによる入力可能なフォームフィールドを作成します。

#### フォームフィールドの種類を指定する

Property Name	axf:field-type
Value	text   button   checkbox   radio   listbox   combobox   signature
Initial	なし (必須)
Applies to	axf:form-field

フォームフィールドの種類を指定する axf:field-type プロパティには値が必須です。

## フォームフィールドの名前を指定する

フォームフィールドの名前は、入力した値を何らかの形で取り出す際に利用されます。

Property Name	axf:field-name
Value	<string>
Initial	なし
Applies to	axf:form-field

Property Name	axf:field-name-suffix-page-number
Value	<string>
Initial	empty string
Applies to	axf:form-field

フォームフィールドの名前を axf:field-name プロパティへ指定可能です。無指定のとき、AH XSL Formatter によって設定されます。

axf:field-name-suffix-page-number プロパティへ書式を指定すると、axf:field-name プロパティの指定が空文字列でないとき、この値に書式を適用したページ番号を付加します。<fo:static-content>内で各ページ独立したフォームを用意したいときに有用です。

## 選択フィールドの項目に、対応する値を指定する

Property Name	axf:field-value
Value	<string>
Initial	empty string
Applies to	axf:form-field, axf:form-field-option

チェックボックスやラジオボタンでそのフィールドを選択しているときなどに適用される値を、axf:field-value プロパティへ指定可能です。

## フィールドのフォントサイズを指定する

Property Name	axf:field-font-size
Value	font-size   auto   <length>
Initial	font-size
Applies to	axf:form-field

フォームフィールドで表示されるテキストのフォントサイズを axf:field-font-size プロパティへ指定可能です。axf:field-font-size="font-size" のとき font-size プロパティで指定されている値となり、axf:field-font-size="auto" ではフォームフィールドに合わせて調整されます。

```

<!-- font-size="10pt"のブロックコンテナ -->
<fo:block-container font-size="10pt" width="100vw" height="3rlh" >
  <!-- formAのフォントサイズは10pt -->
  <axf:form-field field-name="formA" field-font-size="font-size" .../>
  <!-- formBのフォントサイズはAH XSL Formatterが設定 -->
  <axf:form-field field-name="formB" field-font-size="auto" .../>
</fo:block>

```

## 注

axf:field-font-size プロパティは多くのフォームフィールドで共通する制御プロパティです。テキストフィールドに入力されるテキストについては、より細かな制御が可能です。

## フィールドの説明を記述する

Property Name	axf:field-description
Value	<string>
Initial	empty string
Applies to	axf:form-field

axf:field-description プロパティへ、フォームフィールドの説明を記述可能です。Acrobat ではツールチップとして表示されます [図 3.7-1] (p.296)。

```

<!-- 必須フィールドにツールチップによる説明を付加する -->
<axf:form-field
...
field-required="true"
field-description="このフィールドは必須です。" ... />

```

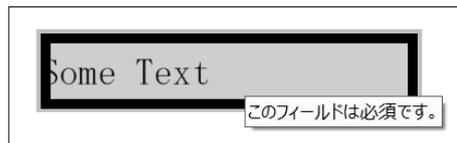


図 3.7-1 必須フィールドにツールチップによる説明を付加する

## 3.7.2 各フィールドの操作権限、状態を指定する

フィールドを読み込み専用や入力必須にします。

Property Name	axf:field-flags
Value	<string>
Initial	empty string
Applies to	axf:form-field

axf:field-flags プロパティへフィールドで可能な操作や状態を指定可能です。

Hidden 以外の項目は個別の拡張でも指定可能ですが、いずれも axf:field-flags での指定が優先されます。文字列は空白区切りで複数指定できます。大文字小文字は区別されません。

axf:field-flags	意味
ReadOnly	読取り専用フィールド。axf:field-readonly="true"と同じ。
Required	入力必須フィールド。axf:field-required="true"と同じ。
Multiline	テキストフィールドを複数行にする。axf:field-multiline="true"と同じ。
Scroll	テキストフィールドをスクロール可能にする。axf:field-scroll="true"と同じ。
Password	テキストフィールドをパスワード化する。axf:field-password="true"と同じ。
Checked	チェックボックス、ラジオボタンをチェック状態にする。axf:field-checked="true"と同じ。
Multiple	リストボックスを複数選択可能にする。axf:field-multiple="true"と同じ。
Editable	コンボボックスで値を編集可能にする。axf:field-editable="true"と同じ。
Hidden	非表示フィールド。フィールドの大きさは確保され、背景やボーダーは表示される。

## フィールドを読み取り専用にする

Property Name	axf:field-readonly
Value	true   false
Initial	false
Applies to	axf:form-field

axf:field-readonly="true"を指定したフォームフィールドは、閲覧ユーザからの入力操作ができなくなります。

### 注

ほかのフォームフィールドで入力した値から JavaScript で計算した結果などを格納するテキストフィールドなどに有用です。

## フィールドを入力必須にする

Property Name	<code>axf:field-required</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field</code>

`axf:field-required="true"`を指定したフィールドは入力必須属性が付加されます。見た目上の変化は PDF ビューアに依存します。

### 3.7.3 テキストフィールドを指定する

閲覧ユーザがテキストを記入可能なテキストフィールドの書式、修飾を指定します。

`<axf:form-field>`に `axf:field-type="text"`を指定するとテキストフィールドになります。テキストフィールドに対してさまざまなレイアウトの指定が可能です。

#### テキストフィールドの揃えを指定する

Property Name	<code>axf:field-text-align</code>
Value	<code>left</code>   <code>center</code>   <code>right</code>
Initial	<code>left</code>
Applies to	<code>axf:form-field</code>

入力されたテキストの揃えを `axf:field-text-align` プロパティへ指定します。

#### 既定で入力されているテキストを指定する

Property Name	<code>axf:field-default-text</code>
Value	<code>&lt;string&gt;</code>
Initial	<code>empty string</code>
Applies to	<code>axf:form-field</code>

`axf:field-default-text` プロパティへ、テキストフィールドの既定値を記述可能です。

#### フィールドをパスワードフィールドにする

Property Name	<code>axf:field-password</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field</code>

axf:field-password プロパティへ true を指定すると、テキストフィールドをパスワードフィールドにします。パスワードフィールドはフォームフィールドへの入力時に入力がマスクされ、PDF ビューアに入力を保存しないように求めます。

Property Name	axf:field-maxlen
Value	<number>
Initial	0
Applies to	axf:form-field

入力可能な最大文字数を axf:field-maxlen プロパティへ指定します。フィールドとして入力可能な文字数を制御するものなので、テキストフィールドの見た目上の端と合わせるには axf:field-font-size などと調整を行う必要があります。

指定が 0 以下のときは文字数を制限しません。axf:field-default-text プロパティに最大文字数以上のテキストを記述したとき、この初期値は設定されません。

axf:field-password="true" を指定したテキストフィールドの例を示します [図 3.7-2] (p.299)。最大文字数をテキストフィールドの領域内に収まるように指定しました。

```
<!-- パスワードフィールドの例 -->
<axf:form-field axf:field-type="text" axf:field-name="TextPassword"
  axf:field-multiline="false"
  axf:field-default-text=""
  axf:field-password="true" axf:field-required="true"
  axf:field-maxlen="8"

  width="10em" height="1.2em"
  background-color="#ff8" border="1pt red inset" />
```



図 3.7-2 パスワードフィールドの例

## テキストフィールドの書式を指定する

フォームへ数値や日付を入力させたいときもテキストフィールドを利用します。

Property Name	axf:field-format-category
Value	none   number   percentage   date   time
Initial	none
Applies to	axf:form-field

Property Name	axf:field-format
Value	auto   [[ <string>   <number>][ <string>   <number>   true   false]*]
Initial	auto
Applies to	axf:form-field

テキストフィールドに入力可能な書式のカテゴリを axf:field-format-category プロパティへ指定し、詳細な書式を axf:field-format プロパティへ指定します。

axf:field-format-category="number" のとき、axf:field-format は空白区切りで最大五つの値を指定します。

axf:field-format	意味	初期値
左から 1 番目	小数点以下の桁数。入力された値の小数点以下の桁数が多いとき、四捨五入。	2
左から 2 番目	桁区切りの方法。	0
左から 3 番目	負数の表示方法。	0
左から 4 番目	通貨記号の任意文字列。	-
左から 5 番目	4 番目で指定した文字列の表示位置。true なら前置、false なら後置。	true

桁区切りの方法は、次に示すいずれかの数値で指定します。

指定	1234.56 の表示
0	1,234.56
1	1234.56
2	1.234,56
3	1234,56
4	1'234.56

負数の表示方法は次の表のいずれかの数値を指定します。

指定	負数の表示方法
0	加工しない。
1	赤色で表示。
2	括弧付き表示。
3	赤色で括弧付き表示。

axf:field-format-category="percentage" を指定したとき、axf:field-format の最大指定は 1 個です。number と同様に、左から 1 番目が小数点以下の桁数、2 番目が桁区切りの方法となります。

通貨記号の文字列は数値の直前直後に配置されるので、空白を含めた文字列を指定すると見やすくなるでしょう。

axf:field-format-category プロパティの指定が date または time のとき、axf:format にはフォーマット文字列での指定と、等価に割り当てられた数字での指定があります。

axf:field-format-category="date"での axf:field-format の数値指定と文字列によるフォーマットの対応は次の表になります。初期値は 0 です。

指定	書式	2019年4月12日13時7分
0	m/d	4/12
1	m/d/yy	4/12/19
2	mm/dd/yy	04/12/19
3	mm/yy	04/19
4	d-mm	12-04
5	d-mmm-yy	12-Apr-19
6	dd-mmm-yy	12-Apr-19
7	yy-mm-dd	19-04-12
8	mmm-yy	Apr-19
9	mmm-yy	April-19
10	mmm d, yyyy	Apr 12, 2019
11	mmm d, yyyy	April 12, 2019
12	m/d/yy h:MM tt	4/12/19 1:07 PM
13	m/d/yy HH:MM	4/12/19 13:07

axf:field-format-category="time"での axf:field-format の数値指定と文字列による書式の対応は次の表になります。初期値は 0 です。

数値	文字列	13時7分02秒の表示
0	HH:MM	13:07
1	h:MM tt	1:07 PM
2	HH:MM:ss	13:07:02
3	h:MM:ss tt	1:06:02 PM

任意の書式が指定可能ですが、誤った書式指定のとき動作は保証されません。

出力した PDF で書式の指定が無効になるときは、Acrobat のバージョンを確認してください。

## テキストフィールドの表示を複数行にする

Property Name	axf:field-multiline
Value	false   true
Initial	false
Applies to	axf:form-field

axf:field-type="text"かつ axf:field-multiline="true"であるとき、複数行のテキストフィールドとなります。見た目上何行入るかについては axf:field-font-size プロパティの値や高さの指定などによります。

Property Name	axf:field-scroll
Value	false   true
Initial	false
Applies to	axf:form-field

「ページの見た目上余裕はないが複数行入るようにしたい」といったとき、axf:field-scroll="true"を指定するとテキストフィールドがスクロール可能になります。

複数行のテキストフィールドの例を示します [図 3.7-3] (p.302)。axf:field-maxlen="100"により最大文字数は 100 となっています。今、フィールドの文字サイズはとくに指定しておらず、テキストフィールドの領域を超えた入力がある可能性があります。axf:field-scroll="true"により、文字が領域を超えるときはスクロールで表示されるようになります。

```
<!-- スクロールバーを表示する -->
<axf:form-field axf:field-type="text"
  axf:field-name="TextMultiline"
  axf:field-default-text="Input text here! sample text"
  axf:field-multiline="true" axf:field-scroll="true"
  axf:field-maxlen="100"

  width="10em" height="2.4em"
  background-color="#ff8" border="1pt solid silver" />
```

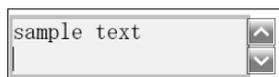


図 3.7-3 スクロールバーを表示する

### 3.7.4 プッシュボタンを指定する

axf:field-type="button"を指定したフィールドはプッシュボタンとなります。アクションの開始やリンクの明示として有用です。

プッシュボタンの外観についての拡張です。プッシュボタンを押したときに何らかのアクションを起こさせたいときは、フォームアクションを記述します。

Property Name	axf:field-button-layout
Value	auto   icon   caption-below-icon   caption-above-icon   icon-caption   caption-icon   caption-over-icon
Initial	caption
Applies to	axf:form-field

Property Name	axf:field-button-face
Value	<string>
Initial	empty string
Applies to	axf:form-field

Property Name	axf:field-button-icon
Value	<uri-specification>
Initial	empty string
Applies to	axf:form-field

axf:field-button-layout プロパティへプッシュボタンのキャプションとアイコンの表示と配置を指定します。

caption はプッシュボタン上に表示されるテキスト（キャプション）の指定です。そのときの内容は axf:field-button-face プロパティへ記述します。

icon はプッシュボタン上に表示されるアイコン画像の指定です。アイコンは PDF 注釈と異なり、<uri-specification>で指定します。

Property Name	axf:field-button-face-down, axf:field-button-face-rollover
Value	<string>
Initial	empty string
Applies to	axf:form-field

Property Name	axf:field-button-icon-down, axf:field-button-icon-rollover
Value	<uri-specification>
Initial	empty string
Applies to	axf:form-field

プッシュボタンを押した（マウスクリックをした）とき、ボタンの表示内容変更を指定可能です。axf:field-button-face-down プロパティへキャプションの変更内容を、axf:field-button-icon-down プロパティへアイコンの変更内容を指定します。押ししている状態が終了すると元の表示内容に戻ります。

RollOver はフィールドの上にマウスポインタがある状態です。プッシュボタン上にマウスポインタがあるときの表示内容の変更が可能です。axf:field-button-face-rollover にキャプションの変更内容、axf:field-button-icon-rollover にアイコンの変更内容を記述します。

プッシュボタンの例を示します [図 3.7-4] (p.304)。axf:field-button-layout="caption-below-icon"によって、アイコン画像がテキストの上に配置されています。アイコンは axf:field-button-icon-rollover でマウスポインタがプッシュボタンの上にあるとき、axf:field-button-icon-down でプッシュボタンが押されたとき、画像が切り換わります。border-style="outset"を指定することで、簡単に周囲よりも浮き出して見えるボタンを表現できます。

```
<!-- アイコンの下にキャプションのあるプッシュボタン-->
<axf:form-field
  axf:field-type="button"
  axf:field-button-layout="caption-below-icon"
  axf:field-button-face="Antenna House, Inc."
  axf:field-button-icon="url('../img/icon.png')"
  axf:field-button-icon-rollover="url('../img/icon-rollover.png')"
  axf:field-button-icon-down="url('../img/icon-down.png')"

  width="15em" height="3em"
  background-color="#eee"
  border="2px silver outset" />
```

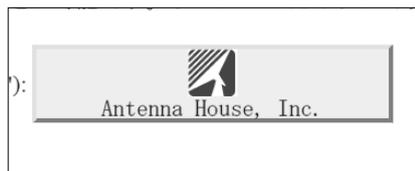


図 3.7-4 プッシュボタンの例

## プッシュボタンを押したときに色を変更する

JavaScript を利用することで、プッシュボタンのアイコンやテキスト以外の要素も変更可能です。

```
<!-- JavaScriptを動作させるプッシュボタン -->
<axf:form-field axf:field-type="button"
  axf:field-name="lock">
  <axf:form-field-event action-type="javascript" name="MouseUp">
    button = this.getField("lock");
    button.fillColor = color.red;
    button.textColor = color.white;
```

```
</axf:form-field-event>
</axf:form-field>
```

マウスが放された（クリックされた）タイミングで `axf:action-type="javascript"` を指定された `axf:form-field-event` の内容が実行されます。PushButtonのフィールド名を取得し、フィールドの色とテキストの色を変更しています。

### 注

「奇数回クリックした」「偶数回クリックした」のような切り換え表示を行うのであれば、非表示のラジオボタンをPushButtonで指定したフォームアクションで切り換えるような方法があります。

## 3.7.5 選択フィールドを指定する

ラジオボタン、チェックボックス、リストボックス、コンボボックスなどを指定します。

### ラジオボタン、チェックボックスを指定する

`axf:field-type` プロパティの指定が `checkbox` でチェックボックス、`radio` でラジオボタンになります。複数項目を選択可能にするのであればチェックボックス、高々ひとつを選択させるのであればラジオボタンのように使い分けられます。いずれも要素の内容は空です。選択時にフィールドとして入力される値は `axf:field-value` プロパティに指定してください。

複数のラジオボタン項目から択一させたいとき、`axf:field-name` プロパティへ指定する値を同一にすることで、複数のラジオボタンを組として扱えます。

Property Name	<code>axf:field-checked</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field</code>

Property Name	<code>axf:field-checked-style</code>
Value	<code>checkmark</code>   <code>circle</code>   <code>square</code>   <code>cross</code>   <code>star</code>
Initial	<code>checkmark</code>
Applies to	<code>axf:form-field</code>

`axf:field-checked="true"` を指定したフィールドは、チェックされた状態をデフォルトの状態とします。

`axf:field-checked-style` はチェックしたときのフィールド枠内の表示スタイルを指定します。枠の形状は変化しません。

## 注

ラジオボタンでは `axf:field-checked="true"` を同じ組の複数のフィールドに指定しても、どれかひとつにしかチェックは付きません。

```
<!-- チェックボックスの指定例 -->
<fo:block ...>
  <axf:form-field axf:field-type="checkbox"
    axf:field-name="CheckMe" axf:field-checked="true"
    axf:field-checked-style="checkmark"

    width="1em" border="1px solid silver" /> Check Me!
</fo:block>
<fo:block ...>
  <axf:form-field axf:field-type="checkbox"
    axf:field-name="CheckCross" axf:field-checked="true"
    axf:field-checked-style="cross"... /> ...
</fo:block>
<fo:block ...>
  <axf:form-field axf:field-type="checkbox"
    axf:field-name="CheckStar"
    axf:field-checked="true" axf:field-checked-style="star".../>...
</fo:block>
```

チェックボックスの例を示します [図 3.7-5] (p.306)。`axf:field-name` に異なる値を指定しているため、それぞれのチェックボックスは独立しています。`axf:field-checked-style` に `checkbox`、`cross`、`star` をそれぞれ指定し、`axf:field-checked="true"` によりチェック状態をデフォルトにしています。



図 3.7-5 チェックボックスの指定例

ラジオボタンの例を示します [図 3.7-6] (p.307)。`axf:field-name="Choice"` をどちらのフィールドにも指定し、同じ組としています。`axf:field-value` の値である `Cat` または `Dog` が、`axf:field-name="Choice"` のフィールドに格納される値となります。フィールドの枠を円形にするため `axf:border-radius="50%"` を指定しています。

```
<!-- ラジオボタンの指定例 -->
<axf:form-field axf:field-type="radio"
  axf:field-name="Choice" axf:field-value="Cat"
  axf:field-checked="false"
  axf:field-checked-style="circle"

  width="1em" border-radius="50%"
  border="1px solid silver"></axf:form-field> Cat
```

```
<axf:form-field axf:field-type="radio" axf:field-name="Choice"
  axf:field-value="Dog" axf:field-checked="false" axf:field-checked-style="circle"
  ... ></axf:form-field> Dog
```



図 3.7-6 ラジオボタンの指定例

## リストボックス、コンボボックスを指定する

`axf:field-type` プロパティへ `listbox` でリストボックスの指定、`combobox` でコンボボックス（ドロップダウンリスト）の指定になります。リストボックスはリスト項目が領域に表示されていますが、項目に対し領域が足りないときはスクロールが表示されます。

コンボボックスは最初に表示されている項目、もしくは空欄を押すと選択可能なリスト項目が表示されます。

Element Name	<code>&lt;axf:form-field-option&gt;</code>
Property	<code>field-value</code>
Parent	<code>axf:form-field</code>
Property Name	<code>axf:field-multiple</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field</code>

`<axf:form-field-option>` 要素へリストボックス、コンボボックスの構成要素を指定します。`<axf:form-field>`の子要素として、出現順序通りリストに並びます。`axf:field-value` プロパティとテキストは値が必須です。

Property Name	<code>axf:field-top-index</code>
Value	<code>&lt;number&gt;</code>
Initial	<code>1</code>
Applies to	<code>axf:form-field</code>
Property Name	<code>axf:field-selected</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field-option</code>

`axf:field-top-index` プロパティは、リストボックス、コンボボックスでの最初の選択項目（デフォルトで選択されている項目）を指定します。`<axf:form-field-option>`の出現順序と指定した数値

が対応します。選択する要素の先頭は 1 となり、`axf:field-top-index` の指定は、1 未満や選択する要素数を超える値のとき 1 とみなされます。

`axf:field-multiple` プロパティへ、リストボックスでの複数項目を選択可能にするかを指定します。`true` のとき複数選択が可能になります。

Property Name	<code>axf:field-editable</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form-field</code>

`axf:field-editable` プロパティへ、コンボボックスで表示されているテキストを編集可能とすることを指定します。`true` のとき編集可能となります。編集可能となるのは `axf:field-value` プロパティの値ではなく、`<axf:form-field-option>` 要素のテキストです。

リストボックスとコンボボックスの例を示します [図 3.7-7] (p.309)。ボーダーに `inset` を指定することで、フィールドを周囲から凹んだように表現可能です。

```
<!-- リストボックスの例 -->
<axf:form-field field-type="listbox"
  field-name="ListChoice"
  field-top-index="1"
  field-multiple="true"

  width="10em" height="3em"
  background-color="#eee"
  border="1pt silver inset">
  <axf:form-field-option
    field-value="ja">Japan</axf:form-field-option>
  <axf:form-field-option
    field-value="ko">Korea</axf:form-field-option>
  <axf:form-field-option
    field-value="cn">China</axf:form-field-option>
  <axf:form-field-option
    field-value="th">Thailand</axf:form-field-option>
</axf:form-field>
```

```
<!-- コンボボックスの例 -->
<axf:form-field field-type="combobox"
  field-name="ComboEditable" field-editable="true"
  field-top-index="1"

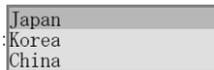
  width="10em" height="1.2em"
  background-color="#ff8" border="1pt silver inset">
  <axf:form-field-option
    axf:field-value="ja">Japan</axf:form-field-option>
  <axf:form-field-option
```

```

    axf:field-value="ko">Korea</axf:form-field-option>
<axf:form-field-option
    axf:field-value="cn">China</axf:form-field-option>
<axf:form-field-option
    axf:field-value="th">Thailand</axf:form-field-option>
</axf:form-field>

```

コンボボックスでは、`<axf:form-field-option>`の内容はリストボックスの例と同じですが、`axf:field-editable="true"`を指定しているため、選択した内容を PDF ビューア上で編集可能です。



リストボックスの例



コンボボックスの例

図 3.7-7 リストボックスとコンボボックスの例

### 3.7.6 電子署名フィールドを制御する

`axf:field-type="signature"`で電子署名フィールドを指定します。このとき、対話フォームへ入力した値のロックを行う設定が可能です。

Property Name	<code>axf:field-lock-document</code>
Value	<code>auto</code>   <code>false</code>   <code>true</code>
Initial	<code>auto</code>
Applies to	<code>axf:form-field</code>

`axf:field-lock-document` プロパティでは、文書に電子署名を行ったときのロックを制御可能です。`auto` は PDF ビューアがダイアログなどのユーザインターフェースを表示します。`false` では文書はロックされず、`true` でロックします。ロックされた PDF は編集ができなくなります。

電子署名での文書ロックは PDF 1.7 以降のとき有効です。

```

<!-- 電子署名フィールドの例 -->
<axf:form ...>...<axf:form>
<!-- 文書のロックを行う電子署名フィールド -->
<axf:form-field axf:field-type="signature"
  axf:field-name="sig"
  axf:field-lock-document="true"
  background-color="lightblue"
  border="1pt silver solid" width="10em" height="3em"/>

```

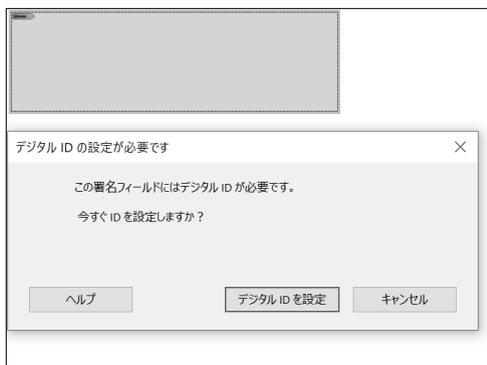


図 3.7-8 電子署名フィールドの例

### 3.7.7 フォームアクションを指定する

PDF フォームの入力と JavaScript でさまざまな動作が行えます。



PDF の種類によっては JavaScript の実行が不許可のものがああります。

Property Name	<code>axf:action-type</code>
Value	<code>auto</code>   <code>goto</code>   <code>gotor</code>   <code>launch</code>   <code>uri</code>   <code>reset</code>   <code>submit</code>   <code>javascript</code>
Initial	<code>auto</code>
Applies to	<code>axf:form-field</code>
Property Name	<code>axf:field-submit-method</code>
Value	<code>get</code>   <code>post</code>
Initial	<code>get</code>
Applies to	<code>axf:form</code>
Property Name	<code>axf:field-submit-coordinates</code>
Value	<code>false</code>   <code>true</code>
Initial	<code>false</code>
Applies to	<code>axf:form</code>

`axf:action-type` プロパティによって指定可能な値はリンクの制御のための `auto`、`goto`、`gotor`、`launch`、`uri` と、フォームに入力した値を操作するための `reset`、`submit`、`javascript` があります。ここでは値の操作に関連する指定について説明します。

入力には人間が慣れた紙の帳票のような PDF 画面を使い、入力された情報は単純な文字列などのデータとして別に保持が可能です。ほか、JavaScript を実行し、「数値の合計を入力から自動で表示したい」といったときに有用です。

axf:action-type	意味
reset	フォームに入力した内容を初期化する。
submit	フォームに入力した内容を HTTP で<axf:form>要素の external-destination プロパティへ指定したリンク先に送信する。
javascript	JavaScript によるアクションを行う。JavaScript プログラムは<axf:form-field-event>要素に記述する。

axf:action-type="submit" のとき、axf:field-submit-coordinates="true" であればマウス座標を含めて送ります。

<axf:form-field-event>を記述しないとき、MouseUp を指定したものとされます。

```
<!-- マウス座標と子のフィールド内容をexternal-destinationのURL先へ送信する -->
<axf:form
  axf:external-destination="..."
  axf:field-submit-method="post" axf:field-submit-coordinates="true">
  <fo:block ...>
    <!-- 入力するテキストフィールド -->
    <axf:form-field axf:field-type="text"
      axf:field-name="InputText" ... width="10em" height="2.4em" ... />
  </fo:block>
  ...
  <fo:block ...>
    <!-- 入力を送信するボタン -->
    <axf:form-field axf:field-type="button"
      axf:field-name="Submit" axf:action-type="submit" />
  </fo:block>
</axf:form>
```

## ユーザの操作イベントを制御する

Element Name	<axf:form-field-event>
Property	name, action-type

<axf:form-field-event>要素は<axf:form-field>の子として記述します。プロパティはいずれも必須です。

name プロパティへイベントの種類、action-type プロパティへ<axf:form-field>と同様に値を指定します。

フォームフィールドをマウスクリック以外で動作させたいときなどに指定します。<axf:form-field>の action-type へ指定したアクションよりも優先されます。

name	意味
MouseUp	マウスボタンがフォームフィールド上で放されたときアクション発生。
MouseDown	マウスボタンがフォームフィールド上で押されたときアクション発生。
MouseEnter	マウスカーソルがフォームフィールドに入ったときアクション発生。
MouseExit	マウスカーソルがフォームフィールドから出たときにアクション発生。
OnFocus	フォームフィールドがフォーカスを得たときアクション発生。
OnBlur	フォームフィールドがフォーカスを失ったときアクション発生。

プッシュボタンでいえば、ボタンを押して放す動作が MouseUp、押しているときに MouseDown です。 MouseEnter はボタン上にマウスを置いたとき、 MouseExit はボタン上から外したときです。

### 注

プッシュボタン上にマウスがある状態は「Roll Over」です。

OnFocus や OnBlur は、たとえば文字を入力するとき、あるテキストフィールドに入力される状態であれば、そのテキストフィールドは「フォーカスを得ている状態」ということです。

## PDF での JavaScript 実行を不許可にする

Option Property Name	allow-javascript
Value	true   false
Initial	true
Applies to	<pdf-settings>

オプション設定 allow-javascript プロパティへ false を指定すると、<axf:document-info name="openaction">や axf:action-type="javascript" のフォームフィールドで記述した JavaScript が無視されます。

### 3.7.8 PDF 全体のフォームへの入力と署名を不許可にする

PDF のフォーム全体に対して、ユーザによる入力と署名を不許可にします。

Option Property Name	no-fill-form
Value	true   false
Initial	false
Applies to	no

出力した PDF ファイルのフォーム全体に対して、ユーザによる入力と署名を許可するかを、オプション設定 no-fill-form プロパティへ指定します。

true にすると変更を許可しません。この制限を行うには `owner-password` が設定されている必要があります。

帳票の例示など、PDF フォームを含む PDF を埋め込みたいが、フォームへの入力禁止したいときに有用です。

## 3.8 節 タグ付き PDF を出力する

オプション設定や GUI、コマンドラインパラメータによりタグ付き PDF を出力します。

文書構造を埋め込まれた PDF を「タグ付き PDF」と呼びます。

AH XSL Formatter の出力設定でタグ付き PDF を選択すると、タグ付きの PDF が出力されます。タグの対応を記述しなくても、適切に FO を記述することで、文書構造に沿ったタグ付き PDF が作成可能です。

Option Property Name	tagged-pdf
Value	true   false
Initial	false
Applies to	<pdf-settings>

オプション設定 `tagged-pdf` プロパティへ `true` を指定すると、出力する PDF をタグ付き PDF にします。出力する PDF のバージョン、種別によってはこの指定は無視されます。

GUI では「組版」「PDF オプション設定 (P)」の「タグ付き PDF」の項目にチェックを入れてください。

タグ付き PDF についての詳細は、マニュアルや、ISO 32000-1:2008・ISO 32000-2:2017 をご覧ください。

タグ付き PDF は PDF 1.4 以降で有効です。

### 3.8.1 PDF タグを記述、マッピングする

オブジェクトに PDF タグを明示します。また、独自のタグのマッピングを行います。

AH XSL Formatter による自動的なタグ付けとは別に、オブジェクトにタグを指定可能です。

Property Name	axf:pdftag
Value	<string>
Initial	empty string
Applies to	all formatting objects

`axf:pdftag` プロパティへタグの名称を文字列で指定します。

PDF の標準タグを `axf:pdftag="P"` のように指定するか、それ以外の非標準タグとして指定します。

Option Property Name	new-tagging-mode
Value	true   false
Initial	false
Applies to	<pdf-settings>

Option Element Name	<tag-role-map>
Parent	<pdf-settings>
Option Element Name	<tag-element>
Property	tag-base-name, tag-derived-name
Parent	<tag-role-map>
Option Property Name	tag-base-name,tag-derived-name
Value	<string>
Initial	なし (必須)
Applies to	<tag-element>

オプション設定 new-tagging-mode プロパティへ true を指定したとき、axf:pdftag="" のように明示的な空文字列を指定すると、その要素はタグを生成せず、親のエリアが属するタグに属します。

このとき、非標準タグであっても、オプション設定 <tag-role-map> で非標準タグ名から標準タグ名へのマッピングが可能です。<tag-role-map> 要素の子要素 <tag-element> では、tag-derived-name プロパティに記述した値を tag-base-name プロパティのタグの要素名へマッピングします。この <tag-role-map> 内でマッピングした値にさらにマッピングを行うことも可能です。

### ⚠ 注意事項

PDF/UA では標準タグ名をほかのタグ名へマップすることは禁止されています。ほかのタグから標準タグへのマップは問題ありません。

## 3.8.2 PDF 2.0 以降に対応したタグ付き PDF かチェックする

PDF 2.0 にはタグの親子関係の制約が追加されました。AH XSL Formatter では、この制約に違反したタグを付与しないよう警告表示をさせられます。

Option Property Name	check-tag-relationship
Value	true   false
Initial	false
Applies to	<pdf-settings>

オプション設定 check-tag-relationship プロパティへ true を指定すると、タグ付きの PDF 2.0 の出力時、タグの親子関係の妥当性を検証します。結果が不正な構造のとき警告を表示します。

### 3.8.3 画像、リンクの代替テキストを記述する

画像が表示できないときやリンクが表示できないとき、スクリーンリーダーによる読み上げのときに利用される代替テキストを記述します。

`<fo:external-graphic>`要素など内容テキストが存在しない画像要素では、そのままの出力ではアクセシビリティ要件である代替テキストが空となります。そのとき、AH XSL Formatter ではアクセシビリティチェックでエラーが生じないように SPACE (U+0020) を代替テキストに挿入します。この代替テキストについて、明示的な指定が可能です。

Property Name	<code>axf:alttext</code>
Value	<code>&lt;string&gt;</code>
Initial	empty string
Applies to	<code>fo:external-graphic</code> , <code>fo:instream-foreign-object</code> , <code>fo:basic-link</code>

`axf:alttext` プロパティへ代替テキストの記述が可能です。

リンクの代替テキストは `axf:annotation-contents` プロパティで指定可能ですが、この指定がない、または空のときにも `axf:alttext` プロパティへ指定したテキストを代替テキストとして処理します。

`axf:alttext` プロパティに指定されたテキストは、設定によっては、画像が存在しないときにテキスト表示として利用されます。

Property Name	<code>axf:display-alttext</code>
Value	<code>true</code>   <code>false</code>   <code>auto</code>
Initial	<code>auto</code>
Applies to	<code>fo:external-graphic</code> , <code>fo:instream-foreign-object</code> , <code>fo:basic-link</code>

Option Property Name	<code>display-alttext</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

`axf:display-alttext` プロパティへ `true` を指定すると、FO で指定されている画像が見つからないとき、`axf:alttext` プロパティへ指定したテキストが表示されます。

`false` では、`src` プロパティで指定された画像が見つからないとき、代替画像を表示します。

`axf:display-alttext="auto"` では、オプション設定 `display-alttext` プロパティでの指定が適用されます。

### 3.8.4 構造要素にスクリプトを指定する

タグ付き PDF では構造要素の言語設定が存在します。AH XSL Formatter はこの機能に対応しています。

オブジェクトに言語指定プロパティ language、country または xml:lang によって言語が指定されていれば、その言語が構造要素の言語とみなされます。

とくに複数のスクリプトが混在する文書などでは言語指定を細かくすることで、組版上の表現以外にもアクセシビリティ対応として効果が得られるでしょう。

```
<!-- 複数の言語が混在するFO -->
<fo:block xml:lang="ja">
「おはよう」はエスペラントで「<fo:inline ...
  xml:lang="eo">Bonan Matenon.</fo:inline>」です。
</fo:block>
```

### 3.8.5 表組でヘッダとセルの対応を明示する

ヘッダが複数あったり、セルが段抜きである表で対応を明示的に指定します。

AH XSL Formatter では「タグ付き PDF」を指定すればオブジェクトから推測されたタグが書き込まれます。表組では、推測できる範囲を超えた、見出しとセルの対応を付加したいことがあります。

Property Name	axf:scope
Value	auto   row   col   rowgroup   colgroup   both
Initial	auto
Applies to	fo:table-cell
Property Name	axf:headers
Value	<idref>
Initial	empty
Applies to	fo:table-cell

axf:scope プロパティは見出しセルへの指定、axf:headers プロパティは内容のセルへの指定です。HTML5 における headers、scope に対応します。

横組の表で axf:scope の指定は、col はそのセルがある縦 1 列、row はそのセルがある横 1 行の見出しであることを示します。見出しセルに colspan や rowspan が含まれる表であれば、colgroup、rowgroup で該当する複数の行または列まとめて見出しであることを示します。

both は縦横方向の見出しであることを示します。縦方向と横方向に見出しを持つような表の「見出しの見出し」になる左上のセルなどに指定するとよいでしょう。

axf:headers プロパティへ、見出しのセルに振った id を指定することでそのセルが対応する見出しを明確に指定します。

次に示すような、縦方向の見出しセル id="head-v1" と横方向の見出しセル id="head-h" に属するセルであれば、axf:headers="head-v1 head-h" のように空白区切りで複数指定が可能です。

```
<!-- 縦横に見出しを持つ表のヘッダとセルの対応を記述する -->
<fo:table>
  <fo:table-header>
    <fo:table-row>
      <fo:table-cell axf:scope="col" id="head-heading" >
        ...
      </fo:table-cell>
      <fo:table-cell axf:scope="col" id="head-v1" >
        ...
      </fo:table-cell>
      <fo:table-cell axf:scope="col" id="head-v2" >...</fo:table-cell>
    </fo:table-row>
  </fo:table-header>
  <fo:table-body>
    <fo:table-row>
      <fo:table-cell axf:scope="row" id="head-h" >...</fo:table-cell>
      <fo:table-cell axf:headers="head-v1 head-h" >...</fo:table-cell>
      <fo:table-cell axf:headers="head-v2 head-h" >...</fo:table-cell>
    </fo:table-row>
    ...
  </table-body>
</fo:table>
```

### 3.8.6 表組の要約を記述する

HTML の table 要素における summary プロパティ相当を記述します。

Property Name	axf:table-summary
Value	<string>
Initial	empty
Applies to	fo:table

表組の要約を axf:table-summary プロパティへ単純なテキストとして記述可能です。タグ付き PDF のとき出力されます。

### 3.8.7 正確なテキスト情報を含む PDF を出力する

出力にタグ付き PDF を指定することで、表示用に加工される前のテキスト情報を含めることが可能です。

PDF 内のテキストを検索したいときには、ハイフネーションや `text-transform` プロパティといった、表示用の加工処理前のテキストが PDF 内にあると有用です。たとえばアラビア語やクメール語など複雑なスクリプトで、Unicode の複雑な変換出力処理をしていたり、タイ語などでのアクセント記号をフォントの GPOS 機能で処理しているときには単純なテキスト検索は難しいでしょう。

タグ付き PDF では `ActualText` としてこれらの加工前テキストを PDF に含めることが可能です。

### 3.8.8 スクリーンリーダーのテキストアクセスを制限する

出力した PDF のテキストがスクリーンリーダーで読み上げられたり、テキストを選択してコピーされることを禁止します。

Option Property Name	<code>no-accessibility</code>
Value	<code>true</code>   <code>false</code>
Initial	<code>false</code>
Applies to	<code>&lt;pdf-settings&gt;</code>

オプション設定 `no-accessibility` プロパティへ `true` を指定すると、出力した PDF ファイルはスクリーンリーダーデバイスからのテキストアクセスを制限します。この制限を行うとき、オプション設定 `owner-password` プロパティへ値が設定されている必要があります。

PDF 1.4 以降への指定で有効です。



# 第4章

## その他の拡張

## 4.1 節 印刷

印刷工程に関連する制御について扱います。

### 4.1.1 用紙の端から仕上り寸法までの距離を指定する

用紙の端から仕上り寸法までの距離を指定します。また、仕上り寸法の位置をはみ出したオブジェクトの描画を制御します。

仕上り寸法は `page-width`、`page-height` または `size` プロパティに指定した寸法です。印刷方法によっては、この仕上り寸法の外側からページ端までに距離が必要になるでしょう。

Property Name	<code>axf:crop-offset</code> , <code>axf:crop-offset-top</code> , <code>axf:crop-offset-bottom</code> , <code>axf:crop-offset-left</code> , <code>axf:crop-offset-right</code>
Value	<code>&lt;length&gt;</code>
Initial	<code>0pt</code>
Applies to	<code>fo:simple-page-master</code>

仕上り寸法の外側から物理的な端までの距離を `axf:crop-offset*` プロパティへ指定します。上下左右に同じ距離を取るのであれば `axf:crop-offset`、個別に指定するのであれば `axf:crop-offset-*` プロパティのそれぞれを指定します。個別の指定と同時に `axf:crop-offset` プロパティを指定したときは個別の値が優先されます。

`page-width="182mm"` で `axf:crop-offset-left="14mm"`、`axf:crop-offset-right="14mm"` とすると、物理的な用紙のサイズは幅 210 mm となります。

#### ⚠ 注意事項

`border` プロパティなどとは異なり、`3mm` `5mm` のようには指定できません。`axf:bleed` プロパティも同様です。

### ページをはみ出した領域を表示する

Property Name	<code>axf:crop-area-visibility</code>
Value	<code>hidden</code>   <code>visible</code>
Initial	<code>hidden</code>
Applies to	<code>fo:simple-page-master</code>

`axf:crop-area-visibility` プロパティへ `visible` を指定すると、仕上り寸法または塗り足しの位置をはみ出す領域を表示します。`hidden` を指定したとき、仕上り寸法または塗り足し領域の位置で領域はクリップされます。

## 4.1.2 塗り足しの幅を指定する

裁ち落としのための塗り足しの領域の幅を指定します。

紙面の端まで画像や色を置くとき、紙面への印刷では紙の伸縮や裁断処理によるズレが生じます。このときにズレによって印刷に隙間ができないように、仕上り予定よりも幅を取って画像や色を配置する領域を塗り足しと呼びます。

Property Name	axf:bleed, axf:bleed-top, axf:bleed-bottom, axf:bleed-left, axf:bleed-right
Value	<length>
Initial	0pt
Applies to	fo:simple-page-master

axf:bleed\*プロパティへ、仕上り領域の外側の塗り足し領域の幅を指定可能です。axf:printer-marks\*プロパティを指定しなければ領域のボーダーは表示されません。上下左右の幅をそれぞれaxf:bleed-top、axf:bleed-bottom、axf:bleed-left、axf:bleed-rightプロパティで指定します。

axf:bleedプロパティへの指定は上下左右に同じ値を指定します。個別の指定と同時に指定したときは個別の値が優先されます。

## 4.1.3 印刷マーク（トンボ）を表示する

西洋トンボや和トンボを表示します。

仕上り寸法や塗り足しの位置を紙面上に表示する、トンボについての拡張です。

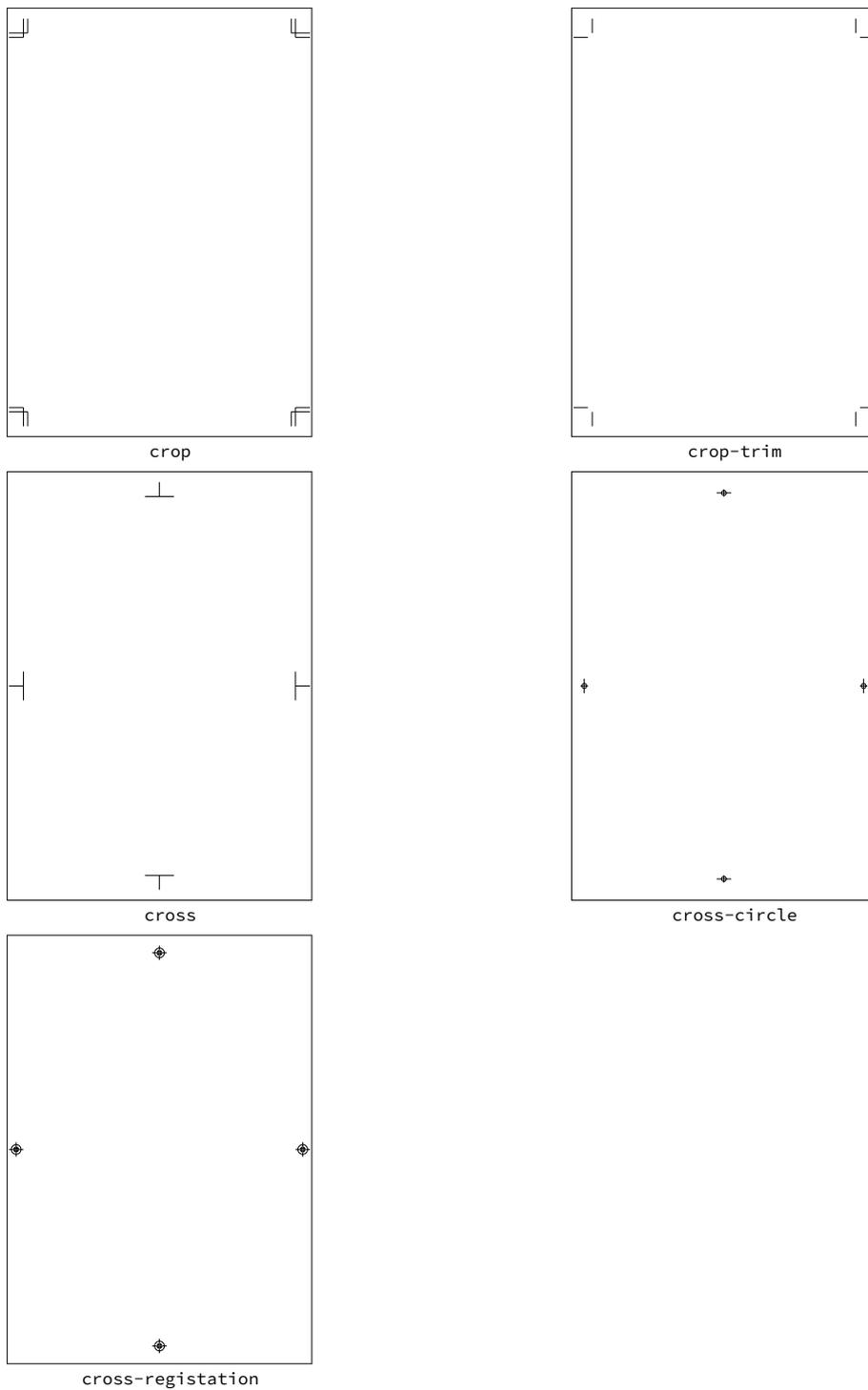
### トンボの種類を指定する

Property Name	axf:printer-marks
Value	[[ crop   crop-trim]    [ cross   cross-circle   cross-registrartion]    <uri-specification>+]   none
Initial	none
Applies to	fo:simple-page-master

出力するトンボの種類を axf:printer-marks プロパティへ指定します。crop でコーナートンボ、crop-trim で仕上り位置のコーナートンボ（内トンボ）、cross で十字センタートンボ、cross-circle で十字センタートンボに円、cross-registation で西洋トンボを表示します [図 4.1-1] (p.324)。

<uri-specification>として、独自のトンボ画像やカラーバーを用意して出力させることが可能です。URI は複数指定可能です。

axf:printer-marks="cross url('./color-bar.svg')"のように指定し、十字センタートンボとカラーバーの画像を同時に出力させることも可能です。



☒ 4.1-1 axf:printer-marks

## 見開き表紙の幅を指定する

Property Name	axf:printer-marks-spine-width
Value	<length>
Initial	0pt
Applies to	fo:simple-page-master

axf:printer-marks-spine-width プロパティへ 0 より大きな値を指定したとき、版面が左右の見開きであるとして、背表紙にあたる部分のセンタートンボを背幅トンボに変更して出力します。左右のページ幅を同一としてその中央位置に描画されます。

## トンボの線を修飾する

Property Name	axf:printer-marks-line-color
Value	<color>   auto
Initial	auto
Applies to	fo:simple-page-master

axf:printer-mark-line-color プロパティにトンボの線の色を指定します。印刷側で指定されているときや、ページの背景が黒色で区別したいときなどに有用です。auto のときはレジストレーションカラー、CMYK では 4 色すべてが 100% の値で印刷されます。

Property Name	axf:printer-marks-line-width
Value	<length>   auto
Initial	auto
Applies to	fo:simple-page-master

Option Property Name	printer-marks-line-width
Value	<length>
Initial	0.24pt
Applies to	<formatter-settings>

トンボの線幅を axf:printer-marks-line-width プロパティへ指定可能です。  
auto のとき、オプション設定 printer-marks-line-width プロパティへ指定した値が利用されま  
す。

Property Name	axf:printer-marks-line-length
Value	<length>   auto
Initial	auto
Applies to	fo:simple-page-master
Option Property Name	printer-marks-line-length
Value	<length>
Initial	10mm
Applies to	<formatter-settings>

トンボの長さを `axf:printer-marks-line-length` プロパティへ指定可能です。  
`auto` のとき、オプション設定 `printer-marks-line-width` プロパティへ指定した値が利用されま  
す。  
`axf:printer-marks="cross-circle"` などでは円のサイズも調整されます。

## 塗り足しが 0 のときの版面とトンボ間の余白を指定する

Property Name	axf:printer-marks-zero-margin
Value	<length>   auto
Initial	auto
Applies to	fo:simple-page-master
Option Property Name	printer-marks-zero-margin
Value	<length>
Initial	3mm
Applies to	<formatter-settings>

`axf:bleed*` プロパティの値が 0 のときの版面とトンボの距離を `axf:printer-marks-zero-margin` プ  
ロパティへ指定します。`auto` のとき、プロパティ `printer-marks-zero-margin` プロパティの値が利  
用されます。

### 4.1.4 オーバープリントを指定する

印刷時、色を重ねて印刷を行うオーバープリントを指定します。

背景画像とテキストなど、オブジェクトを配置するとき色が重なって配置される箇所がありま  
す。印刷では、この配置通りに重ねて色を置くか、重なった部分の下の色を除くかによって違いが  
生じることがあります。重ねて置く方法をオーバープリント、あるいはノセと呼びます。下の色を  
除く方法はトラッピング、あるいはヌキと呼びます。PDF 出力時にオーバープリントを指定する  
と、重なった箇所の前面の色と背面の色を保持します。

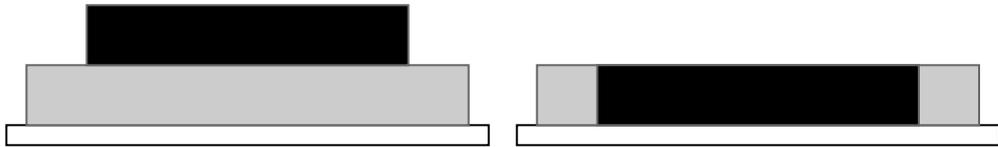


図 4.1-2 印刷時のノセ (左) とヌキ (右) のイメージ

Property Name	axf:overprint
Value	auto   <i>&lt;overprint&gt;</i> #
Initial	auto
Applies to	all formatting objects
Inherited	yes
Option Property Name	overprint
Value	<i>&lt;overprint&gt;</i> #
Initial	<i>empty</i>
Applies to	<i>&lt;pdf-settings&gt;</i>

Value	Value Composition
<i>&lt;overprint&gt;</i>	<i>&lt;opcolor&gt;</i> [[ <i>stroke</i>    <i>paint</i>    [ <i>full</i>   <i>nonzero</i> ]]   <i>none</i> ]?
<i>&lt;opcolor&gt;</i>	<i>&lt;color&gt;</i>   <i>k100</i>   <i>separation</i>   <i>all</i>

axf:overprint プロパティへオブジェクトのオーバープリントを指定します。PDF 出力でのみ有効です。

オプション設定 overprint プロパティへ、axf:overprint="auto"のときオーバープリントを適用する色を指定します。overprint="k100"と指定しておくで、FO のすべてのページで墨ノセが行なわれます。

axf:overprint	意味
<i>&lt;opcolor&gt;</i>	オーバープリントを適用したい色を指定する。
<i>stroke</i>	線の描画へオーバープリントを適用する。
<i>paint</i>	塗りの描画へオーバープリントを適用する。
<i>full, nonzero</i>	オーバープリントの適用時、カラー成分が0のときの動作を指定する。fullはそのカラー成分を0に、nonzeroは変更しない(色がないとみなされる)。CMYKのときのみ効果。
<i>none</i>	オーバープリントを適用しない。

*<opcolor>*としては透過色を除く*<color>*、*k100*、*separation*、*all*が指定可能です。

*k100*は *rgb-icc(#CMYK,0,0,0,1)* にオーバープリントを適用します。

*separation*は *rgb-icc(#Separation)* で示されるセパレーションカラーにオーバープリントを適用します。このとき、レジストレーションカラーは含みません。

*all*は透明度を持たないすべての色へオーバープリントを適用します。

<opcolor>のみ指定したとき、同時に stroke paint nonzero を指定したとみなされます。

オーバープリントからの除外指定を行うときは none を指定します。

オーバープリントの指定はカンマ区切りの組で複数指定できますが、複数に解釈できる色について、<color>、k100、separation、all の優先順位でマッチした指定が採用されます。

前面の色と背面の色の指定によってオーバープリントの効果は変わります。背景色がセパレーションカラーであるときは、ほぼ確実にオーバープリントを指定した前景色が有効になります。

---

#### 注

PDF を埋め込むとき、埋め込み元の PDF は処理されません。オーバープリントを行うのであれば、あらかじめオーバープリントを適用した PDF を埋め込んでください。

---

ほか、詳細についてはマニュアルをご覧ください。

## 4.1.5 PDF で印刷の不許可、印刷可能な解像度を指定する

印刷の不許可や、印刷を許可する解像度を指定した PDF を出力します。実際に印刷が不許可になるかどうかは PDF ビューアに依存します。

Option Property Name	printing-allowed
Value	high-resolution   low-resolution   none
Initial	high-resolution
Applies to	<pdf-settings>

オプション設定 printing-allowed プロパティへ PDF ファイルの印刷許可設定を指定します。none のときは印刷を許可しません。high-resolution は高解像度での印刷を許可します。low-resolution では低解像度での印刷のみを許可します。

low-resolution は PDF 1.4 以降で有効で、PDF 1.3 では high-resolution とみなされます。許可を制限するとき、owner-password プロパティへ値が設定されている必要があります。

## 4.1.6 Windows 版の印刷出力でプリンタの動作を指定する

Windows 版では、印刷出力のときページシーケンスを単位として両面印刷や用紙トレイを選択可能です。有効かどうかはプリンタにも依存します。

Property Name	axf:printer-bin-selection
Value	<string>   <integer>
Initial	環境に依存
Applies to	fo:page-sequence

プリンタのトレイ名またはトレイ番号を `axf:printer-bin-selection` プロパティへ指定します。このトレイ名はプリンタ設定ダイアログに出てくるもので、プリンタドライバから取得されます。トレイ番号もプリンタドライバから取得されるものです。プリンタドライバで取得できるものと一致しないとき無効です。

たとえば、ページサイズの異なるページが混在するときに指定することで、それぞれのページに応じたサイズでの印刷が可能です。

Property Name	<code>axf:printer-duplex</code>
Value	<code>&lt;integer&gt;</code>
Initial	環境に依存
Applies to	<code>fo:page-sequence</code>
Option Property Name	<code>SeparatePrinterDuplexJob</code>
Value	<code>true   false</code>
Initial	<code>true</code>
Applies to	<code>&lt;formatter-settings&gt;</code>

`axf:printer-duplex` プロパティに数値で指定します。プリンタドライバで指定されている値を指定するため、数値の対応はプリンタによって異なることもあります。多くのプリンタでは、1 が片面印刷、2 が両面印刷の長辺綴じ、3 が両面印刷の短辺綴じ設定となります。

オプション設定 `SeparatePrinterDuplexJob` プロパティでは、この印刷指定が切り換わる時、印刷を分割するか、一括で出力するかを制御します。`true` で分割、`false` で一括印刷を行います。

```
<!-- ページサイズが違う図表の印刷時、違うトレイを使うよう指定する -->
<fo:page-sequence master-reference="A4"
  axf:printer-bin-selection="1"
  axf:printer-duplex="2">
  ...
  <fo:page-sequence master-reference="A3broadtable"
    axf:printer-bin-selection="2">
    ...
  </fo:page-sequence>
</fo:page-sequence>
```

## 4.1.7 ユースケース：背表紙を含む表紙にトンボを付加する

背表紙や裏表紙を含めた表紙ページを、トンボを付加して作成します。

表紙、背表紙、裏表紙を1枚のページとして作成します [図 4.1-3] (p.330)。



図 4.1-3 背表紙を含む表紙にトンボを付加する

### 表紙のページ幅を指定する

```
<!-- 表紙のページマスタ -->  
<fo:simple-page-master  
  page-height="257mm" page-width="182mm + 182mm + 10mm"  
  margin="0mm" ...>  
  ...  
</fo:simple-page-master>
```

ページマスタに指定する `page-width` は表紙幅、裏表紙幅と背表紙幅の合計値です。背表紙幅の指定のため、事前に完成時の本の厚さを把握しておく必要があります。

## 注

<afx:spread-page-master>は2ページにまたがり存在する要素を想定した機能です。背表紙付きの表紙を1枚のページとして出力するときには適しません。

## トンボを付加する

```
<!-- 塗り足し領域とトンボの指定 -->
<fo:simple-page-master
  ...
  axf:crop-offset-top="(297mm -257mm) div 2"
  axf:crop-offset-bottom="(297mm -257mm) div 2"
  axf:crop-offset-left="(420mm - 364mm - 10mm) div 2"
  axf:crop-offset-right="(420mm - 364mm - 10mm) div 2"
  axf:bleed="5mm"
  axf:printer-marks-spine-width="10mm"
  axf:printer-marks="crop cross"
  ...>
  ...
</fo:simple-page-master>
```

仕上りの外側の領域を `afx:crop-offset*` で指定し、塗り足し領域を `afx:bleed*` で指定します。そして各種の印刷マーク（トンボ）の表示を `afx:printer-marks` で指定します。表紙以外のページサイズが JIS B5 で長辺が高さとして、表紙のページサイズは幅を倍にし、背表紙の幅を加えたサイズとなります。物理サイズが A3 サイズとなるように `afx:crop-offset-*` プロパティへ値を指定しました。

## ページ全体に塗り足しを考慮した背景を指定する

```
<!-- ページマスタに背景を指定する -->
<fo:simple-page-master ...
  background-image="url(./background.jpg)"
  axf:background-image-resolution="300"
  axf:background-size="cover"
  axf:background-position-horizontal="center"
  axf:background-position-vertical="center"
  axf:background-repeat="no-repeat">
  ...
</fo:simple-page-master>
```

背景は `page-width`、`page-height` プロパティで指定した値に `afx:bleed` で指定した値を反映した領域となります。

## 本文区画の境界線を描画する

```
<!-- 本文区画の境界線 -->
<fo:region-body
  margin="0"
  border="solid 2pt cmyk(0, 0 , 0 ,0)" />
```

説明のため本文区画のボーダーを表示しました。XSL 1.1 仕様では<fo:region-\*>への padding や border の幅は 0 なので境界線は描画できませんが、AH XSL Formatter では制限されていません。

## 表紙、裏表紙、背表紙をブロックコンテナで配置する

```
<!-- 表紙、裏表紙、背表紙をブロックコンテナで配置する -->
<fo:flow ...>
  <!-- 表紙 -->
  <fo:block-container absolute-position="absolute"
    width="182mm" top="33vh" left="182mm + 10mm -2pt">
    <fo:block ...>Cover</fo:block>
  </fo:block-container>
  <!-- 裏表紙 -->
  <fo:block-container absolute-position="absolute"
    width="182mm" top="33vh" left="0 -2pt">
    <fo:block ...>Back </fo:block />
    Cover</fo:block>
  </fo:block-container>
  <!-- 背表紙 -->
  <fo:block-container
    width="10mm" height="257mm"
    absolute-position="absolute" top="0" left="182mm - 2pt">
    <fo:block-container ... reference-orientation="-90">
      <fo:block text-align="center"
        background-color="cmyka(0, 0, 0, 0, 1)"
        axf:box-shadow="1mm 1mm 3pt 1mm cmyka(0,0,0,0,0.8),
          1mm -1mm 3pt 1mm cmyka(0,0,0,0,0.8),
          -1mm 1mm 3pt 1mm cmyka(0,0,0,0,0.8),
          -1mm -1mm 3pt 1mm cmyka(0,0,0,0,0.8)"
        font-size="6mm" line-height="1" ... >
        Title
      </fo:block>
    </fo:block-container>
  </fo:block-container>
</fo:flow>
```

表紙と裏表紙にブロックコンテナの絶対配置でテキストを配置しました。横組であれば、表紙は左端から裏表紙のページ幅と背表紙の幅を足した位置からになります。裏表紙は左端の始点です。背表紙は左端から裏表紙のページ幅の位置です。

- 2pt としているのは、本文区画のボーダーの幅の調整のためです。

背表紙のテキスト方向は、絶対配置のためのブロックコンテナの子に、`reference-orientation="-90"`を指定したブロックコンテナを指定しました。文字が小さくなるため、輪郭線ではなく背景色を指定しています。タイトルの背景色と画像の境界をぼかすため、ボックスシャドウを指定しています。

## 4.1.8 ユースケース：塗り足しを含めたツメを配置する

ブロックコンテナの絶対配置によって、塗り足し部分まで背景を置きます。

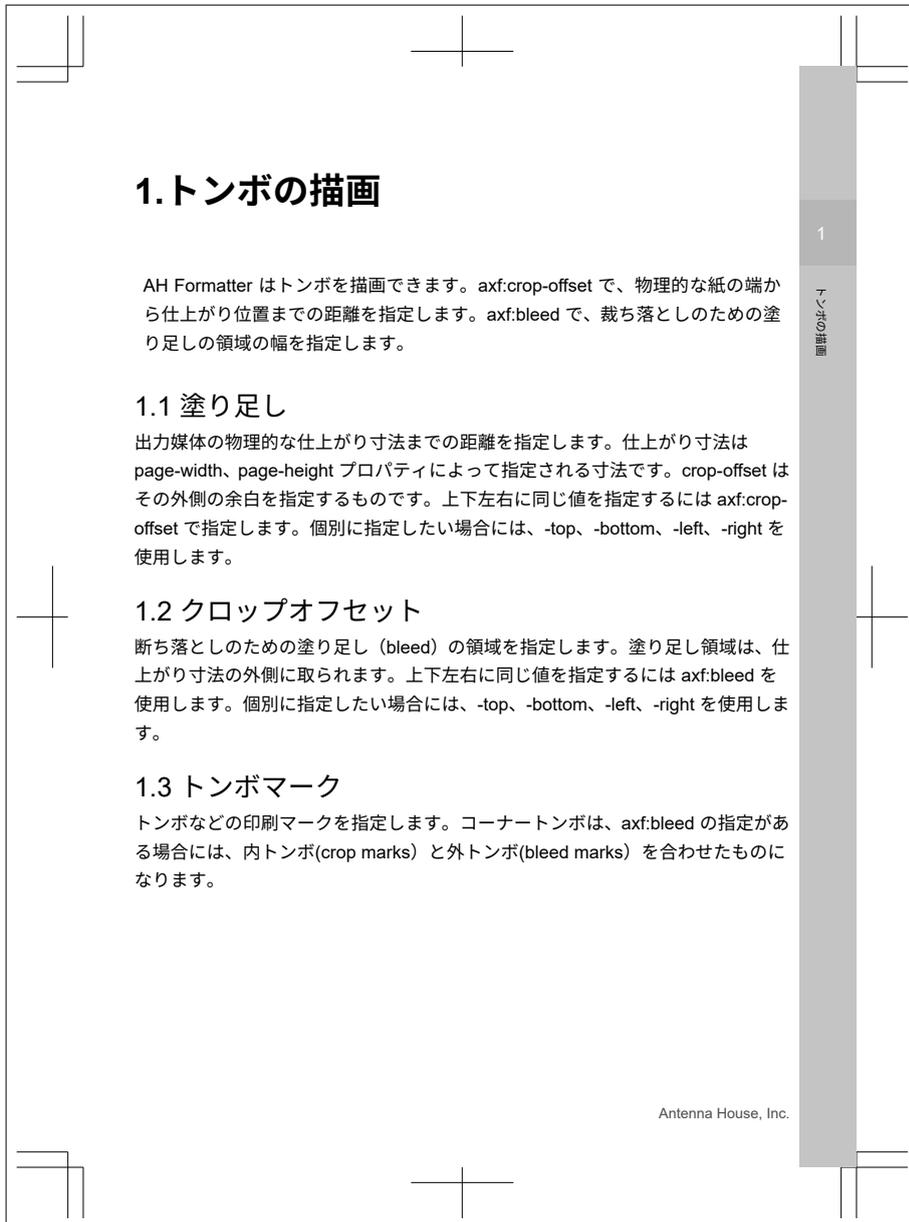


図 4.1-4 塗り足しを含めたツメを配置する

ページマスタへの背景であれば、`axf:bleed` を指定したときに背景範囲も拡大します。画像の解像度指定などを除けば塗り足しのために特別な変更は必要ありません。

しかし、`<fo:region-*>`の子として指定するヘッダやフッタ、ツメは特別に指定しなければ`<fo:region-*>`の内側で描画され、塗り足しの位置に空白が生じてしまいます。

区画をはみ出すようにブロックコンテナを記述し、塗り足しを描画します [図 4.1-4] (p.334)。

## トンボを指定する

```
<!-- ページマスタにトンボを指定する -->
<fo:simple-page-master
  page-width="148mm" page-height="210mm"
  margin="0"
  axf:crop-offset-top="15mm" axf:crop-offset-bottom="15mm"
  axf:crop-offset-left="15mm" axf:crop-offset-right="15mm"
  axf:crop-area-visibility="true"
  axf:bleed="3mm"
  axf:printer-marks="crop cross"
  axf:printer-marks-line-color="rgb-icc(#CMYK, 0%, 0%, 0%, 100%)"
  axf:printer-marks-line-length="auto"
  axf:printer-marks-line-width="auto">
...
</fo:simple-page-master>
```

ページマスタに `axf:crop-area-visibility="visible"` と指定します。 `axf:crop-offset-*` と `axf:bleed`、`axf:printer-marks*` プロパティを指定し、トンボを描画しています。

## ツメ組の区画を指定する

```
<!-- ツメのために区画の優先順を変更する -->
<fo:region-before ...
  extent="10mm" display-align="before"
  precedence="false" ... />
<fo:region-end region-name="rgnEndRight"
  extent="10mm" display-align="before"
  reference-orientation="-90" />...
```

ツメの背景がヘッダと被らないように、`<fo:region-before>` 要素の `precedence` に `false` を指定しています。この指定により `<fo:region-end>` の描画が優先されます。

## ツメをブロックコンテナで絶対配置する

```
<!-- 塗り足し分ブロックコンテナを動かす -->
<fo:static-content flow-name="rgnEndRight">
  <fo:block-container position="absolute"
    top="-3mm" left="-3mm" right="-3mm"
    height="13mm">
    ...
  </fo:block-container>
</fo:static-content>
```

reference-orientation="-90"により、ページから見て右側がブロックコンテナの top、上側が left、下側が right で指定する値です。axf:bleed プロパティへ 3 mm の塗り足しを指定したため、3mm ずつずらして配置します。ブロックコンテナの基点は塗り足しを含まない位置のため、負の値を指定しています。高さは<fo:region-end>に 3 mm を足した 13 です。

## ツメを修飾する

```
<!-- ツメを修飾する -->
<fo:block
  line-height="1"
  background-color="rgb-icc(#CMYK, 39%, 8.9%, 0%, 7.5%)">
  <fo:inline-container width="12.8mm"
    height="8mm + 3mm"
    alignment-baseline="text-after-edge">
    <fo:inline-container>
      <fo:block-container ... reference-orientation="90">
        ...
        <fo:inline>1</fo:inline>
        ...
      </fo:block-container>
    </fo:inline-container>
  </fo:inline-container>
  <fo:inline-container>
    <fo:block font-size="6.5pt">
      <fo:inline baseline-shift="8pt">
        <fo:retrieve-marker retrieve-class-name="index-A" />
      </fo:inline>
    </fo:block>
  </fo:inline-container>
  ...
</fo:block>
```

ツメに表示する番号は多くとも 4 桁程度として、参照する章見出しも 1 行を超えないとします。line-height="1"として、余白の調整はインラインコンテナで行っています。章の番号は正立させるため、ブロックコンテナで reference-orientation="90"を指定し、ページと同じ方向に戻しています。章見出しの参照は、baseline-shift を指定して下側に余白をもたせています。

もし章ごとにツメの位置をずらしたいというときは、ページシーケンスを変更し、位置をずらしたツメを新たに記述していくのが確実です。

## 4.2 節 ウォーターマーク（透かし）を指定する

ページにウォーターマーク（透かし）を挿入します。

出力する PDF や印刷書類に権利や制限を明記するために透かしを挿入したいとき、ふたつの方針が考えられます。SVG 画像などをページ背景画像として指定する方法と、オプション設定で透かしのテキストを指定する方法です。

### 透かしとして背景画像を指定する

```
<!-- 背景画像で透かしを表現する -->
<fo:region-body ...
  background-image="watermark.svg"
  background-position-horizontal="center" />
```

ページマスタや本文区画に背景画像を指定し、透かしを表現します。「特定の文字列を繰り返し表示したい」、「異なるページサイズでも同じ透かし画像を使用したい」といったときには SVG 画像を用いるのがよいでしょう。background-image プロパティへ複数の画像を指定し、透かし用の画像の後にページ背景にしたい画像を別に追加することも可能です。

### オプション設定で透かしのテキストを記述する

Option Property Name	watermark-text, watermark2-text
Value	<string>
Initial	empty
Applies to	<formatter-settings>
Option Property Name	watermark-font-family
Value	[ <family-name>   <generic-family> ]#
Initial	sans-serif
Applies to	<formatter-settings>
Option Property Name	watermark2-font-family
Value	[ <family-name>   <generic-family> ]#
Initial	the watermark-font-family name
Applies to	<formatter-settings>

Option Property Name	watermark-font-style, watermark2-font-style
Value	normal   italic
Initial	normal
Applies to	<formatter-settings>

Option Property Name	watermark-font-weight, watermark2-font-weight
Value	normal   bold   <integer>
Initial	normal
Applies to	<formatter-settings>

透明度を含むテキストを出力に挿入する機能であるため、出力に透明度を含めない種類、バージョンの PDF には適用できません。

オプション設定による透かしテキストは2種類あります。watermark-\*プロパティはページを斜め方向に挿入される透かしテキストの制御、watermark2-\*プロパティはページ下部に挿入されるテキストの制御です。

watermark\*-text に透かしとして挿入するテキストを指定します。watermark-text では改行の文字参照 (&#10;) で区切り、複数行が可能です。

単純なテキストのみ指定可能です。単純でない、複雑なテキストとしては、単一のフォントで出力できないテキストや、タイ語やアラビア語など複雑なスクリプトがあります。

watermark\*-font-weight プロパティへ指定可能な数値は1~1000の整数値です。

透かしテキストのフォントファミリーはそれぞれ watermark\*-font-family で指定します。このときテキストに対し watermark\*-font-family プロパティを適切に指定していなければ表示が正しく行われなかもしれません。

## 注

文字数が増えると幅を狭めて表示するため、1行あたりの文字数が多いと可読性に影響します。

```
<!-- オプション設定による透かし -->
<fo:declarations>
  <axf:formatter-config
    xmlns:axs="http://www.antennahouse.com/names/XSL/Settings">
    <axs:formatter-settings
      watermark-text="Antenna House&#10; Watermark"
      watermark-font-family="'Impact'"
      watermark-font-weight="Bold"
      watermark-font-style="italic"
      watermark2-font-family="'Noto Serif CJK Jp'"
      watermark2-text="これもウォーターマーク"/>
    </axs:formatter-settings>
  </axf:formatter-config>
</fo:declarations>
```

```
</axf:formatter-config>  
</fo:declarations>
```

上の設定を適用した例が [図 4.2-1] (p.339) です。見やすくするため灰色の背景色を指定しました。

---

#### 注

watermark-text プロパティによるテキストは白抜き輪郭線、watermark2-text によるテキストは灰色で表示されるため、背景色によっては見えなくなることがあります。

---

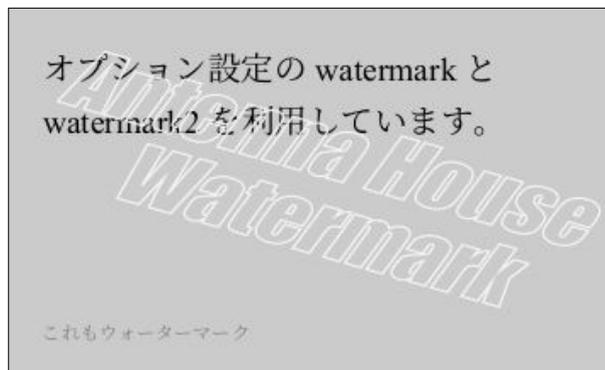


図 4.2-1 オプション設定による透かし

## 4.3 節 文書内の相対 URI の基点を指定する

文書内で相対パスで記述されている URI の基点を指定します。

Property Name	axf:base-uri
Value	<uri-specification>
Initial	empty string
Applies to	all formatting objects
Inherited	yes

FO 内の相対 URI について、その基点となる URI を axf:base-uri プロパティへ指定可能です。xml:base プロパティへ値が指定されたときも同様に処理されます。本番環境とテスト環境でリソースのパスが異なるときなどに有用です。

```
<!-- URIの基点を指定する -->
<fo:root ...
  axf:base-uri="https://antennahouse.co.jp/AHF/"
  ...
  <fo:basic-link external-destination="path/to/AHFbook.pdf">
  ...</fo:basic-link>
  <fo:wrapper ... axf:base-uri="/data/Antenna/images/" >
  ...
  <fo:external-image href="uri_sample.pdf" />
  </fo:wrapper>
</fo:root>
```

上の FO では<fo:root>要素では axf:base-uri プロパティへ https://antennahouse.co.jp/AHF/を指定しており、相対 URI の基点はこの値になります。

<fo:wrapper>要素で axf:base-uri プロパティに/data/Antenna/images/を指定しているため、この<fo:wrapper>とその子要素では基点の URI は/data/Antenna/images/となります。

## 4.4 節 オプション設定を FO に記述する

Element Name	<axf:formatter-config>
Property	src
Parent	fo:declarations

<axf:formatter-config>の子要素、または src プロパティに外部 XML ファイルとしてオプション設定の記述が可能です。FO でのオプション設定はネームスペースを別に持つことに注意してください。オプション設定のネームスペースを次に示します。

<http://www.antennahouse.com/names/XSL/Settings>

初期化にかかわるものなど、<axf:formatter-config>プロパティの指定からは無視されるオプション設定もあります。記述が可能であるのは、<formatter-settings>、<pdf-settings>、<text-settings>、<svg-settings>、<mif-settings>、<ps-settings>、<mathml-settings>、<cgm-settings> とその子要素です。

```
<!-- FOでのオプション設定の記述 -->
<fo:declarations>
  <axf:formatter-config
    axs="http://www.antennahouse.com/names/XSL/Settings">
    <axs:formatter-settings ... />
    <axs:pdf-settings ... />
  </axf:formatter-config>
</fo:declarations>
```

### オプション設定の指定を無効にする

Option Property Name	axf-formatter-config
Value	true   false
Initial	true
Applies to	<formatter-settings>

オプション設定ファイルで axf-formatter-config プロパティに false を指定したとき、FO での <axf:formatter-config>によるオプション設定の記述は無効です。

## 4.5 節 エラーメッセージ表示を抑制する

組版処理中に GUI やコマンドラインで表示されるエラーメッセージについて、抑制が可能です。

フォントのフォールバックや外部画像参照先が存在しないなど、エラーメッセージを出力するものの組版自体は可能なエラーのとき、AH XSL Formatter は該当のエラーメッセージを出力しながら組版を継続します。

Option Property Name	no-disp-warnings
Value	[ <decimal>   <hex>]*
Initial	empty
Applies to	<formatter-settings>

オプション設定 no-disp-warnings プロパティへ表示を抑制したいエラーメッセージのエラーコードを 10 進数または 16 進数で列挙することで、該当エラーメッセージ表示の抑制が可能です。

組版可能なとき表示の抑制を行う指定であり、処理が中断されるようなエラーではこの指定は無効となります。

## 4.6 節 バーコードについての拡張

バーコードジェネレータオプションを購入いただくことで、さまざまな種類のバーコードを生成可能になります。また、他の拡張と組み合わせることでさまざまな配置が可能です。

バーコードの種類によって指定可能なパラメータは異なります。仕様や各パラメータの詳細についてはマニュアルをご覧ください。パラメータやオプション設定の変更で可能なこと、難しいことの対処例を紹介します。

### GS1-128/EAN-128 の識別子を追加、変更する

AH XSL Formatter で未登録のアプリケーション識別子 (AI) を登録、登録済みの AI の変更が可能です。登録済みの AI についてはマニュアルをご覧ください。

Option Element Name	<GS1-128>
Property	AI, format
Parent	<formatter-settings>
Option Property Name	AI
Value	<string>
Initial	empty
Applies to	<GS1-128>
Option Property Name	format
Value	<string>
Initial	empty
Applies to	<GS1-128>

<GS1-128>要素の AI プロパティへ登録アプリケーションを、format プロパティへ書式を指定します。

### 線形バーコードのバーの高さだけを変更する

線形バーコードで、スキャナが読み取れる範囲で線幅や高さを変更することを考えます。希望の形によっては、ブロックを変形せず、バーコード生成時に渡すパラメータの調整によって解決するものもあります。長さや幅、余白のパラメータは単位付きの値で指定してください。

```
<!-- 線形バーコードのバーの高さだけを変更する -->
<fo:block>
  <fo:external-graphic
    src="data:application/vnd.ah-barcode;
```

```
type=EAN; h=5mm; text=auto,4901234567894">
</fo:block>
```



図 4.6-1 線形バーコードのバーの高さだけを変更する

## 線形バーコードのフォントを変更する

Option Property Name	barcode-text-font
Value	[ <family-name>   <generic-family>]#
Initial	OCRB, monospace
Applies to	<font-settings>

線形バーコードにはテキストを表示するものがあります。このとき利用されるフォントを変更する方法はふたつあります。オプション設定 `barcode-text-font` プロパティを変更する方法と、バーコードジェネレータでのテキストを非表示にし、別のテキストを重ねる方法です。前者は安全、簡便に行えます。後者は、バーコードとの距離を指定したりやバーコードと分離した回転などが可能ですが、制御が複雑になります。

オプション設定 `barcode-text-font` プロパティでは、バーコードジェネレータオプションで生成する線形バーコードに元コードのテキストを付加する、つまりパラメータ `text` が `none` 以外のときのフォントを指定可能です。

## バーコードを回転して表示する

生成されるバーコードは一般的な図表と同じく、親のブロックごと表示方向の変更が可能です。1次元バーコードのテキストだけ回転させたくないため、テキストを別ブロックにしました [図 4.6-2] (p.345)。

```
<!-- axf:transformで回転 -->
<fo:block
  axf:transform="translateY(5mm) rotate(90)">
  <fo:external-graphic
    src="data:application/vnd.ah-barcode;
    type=EAN; text=none, 4901234567894" />
</fo:block>
<fo:block text-align="center"
  axf:transform="translateY(10mm)"
  font-family="OCRB, monospace" >
  4901234567894
</fo:block>
```

ブロックコンテナで方向を変えることも可能ですが、この FO では `axf:transform` プロパティを利用しています。

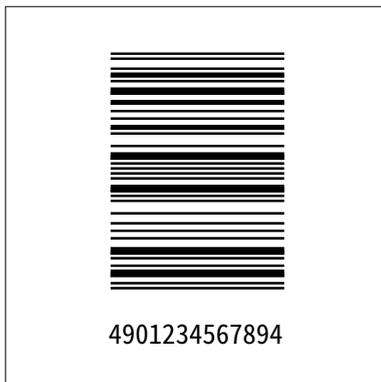


図 4.6-2 テキストを別のブロックにする

# 本書中の拡張仕様一覧

A	
axf:abbreviation-character-count.....	181
abbreviation-character-count.....	181
axf:action-type.....	50-53,276
goto.....	274
gotor.....	274
javascript.....	310
launch.....	274
reset.....	310
submit.....	310
uri.....	274
allow-javascript.....	312
axf:alttext.....	316
axf:annotation-author .....	265
axf:annotation-border-color.....	267
axf:annotation-border-style.....	267
axf:annotation-border-width.....	267
axf:annotation-color .....	266
axf:annotation-contents .....	271
axf:annotation-createdate .....	265
axf:annotation-file-attachment.....	266,272,273
axf:annotation-flags.....	270
axf:annotation-flags.....	293
Hidden.....	271
Invisible.....	271
Locked.....	271
LockedContents.....	271
NoRotate.....	271
NoView.....	271
NoZoom.....	271
Print.....	271
ReadOnly.....	271
ToggleNoView.....	271
axf:annotation-font-family.....	268
axf:annotation-font-size.....	268
axf:annotation-font-style.....	269
axf:annotation-font-weight.....	268
axf:annotation-height.....	269,272,273
axf:annotation-icon-name.....	267
axf:annotation-modifydate .....	265
axf:annotation-open .....	271
axf:annotation-position-horizontal. 270,272,273	
axf:annotation-position-vertical.....	270
axf:annotation-text-align.....	269
axf:annotation-text-color.....	268
axf:annotation-title .....	265
axf:annotation-type.....	264
FileAttachment.....	266,272,273
FreeText.....	268
axf:annotation-width.....	269,272,273
axf:append-non-end-of-line-characters.....	179
append-non-end-of-line-characters.....	179
axf:append-non-starter-characters.....	179
append-non-starter-characters .....	179
axf:assumed-page-number .....	151
auto-break-footnote.....	146
auto-fallback-font.....	231
axf:auto-letter-spacing.....	203
avoid-orphan-single-word.....	180
axf:avoid-widow-words.....	162
axf:avoid-widow-words-cjk-punctuation ...	163
avoid-widow-words-cjk-punctuation.....	163
axf-formatter-config.....	341

## B

axf:background-image-resolution .....	22-25,123,330-333
barcode-text-font.....	344
xml:base.....	340
axf:baseline-block-snap.....	26-30,35-41,185
axf:baseline-grid.....	26-30,35-41,184
axf:base-uri .....	340
axf:bleed.....	323,330-336
axf:bleed-bottom.....	323
axf:bleed-left.....	323
axf:bleed-right.....	323
axf:bleed-top.....	323
<fo:block-container>	
column-count.....	35-41,101,116,117
column-gap.....	35-41,101,116,117
bold-ratio.....	219
axf:bookmark-include.....	279
axf:border-bottom-left-radius .....	22-25,72
axf:border-bottom-right-radius .	22-25,72,92-95
axf:border-connection-form.....	72
axf:border-double-thickness.....	35-41,71
axf:border-radius.....	22-25,72,91,92,130
<border-style>	
dot-dash.....	70
dot-dot-dash.....	70
wave.....	31-34,70
axf:border-top-left-radius.....	35-41,72,92-95
axf:border-top-right-radius.....	72
axf:border-wave-form.....	70
axf:box-shadow.....	74,91,92
bpil.....	169
bpil-limit-chars.....	170
bpil-minimum-line-width.....	170
bpil-penalty-hyphenation.....	170

## C

cap.....	56
cc.....	56
ch.....	56
<fo:change-bar-begin>.....	86,97-100
check-tag-relationship.....	315
<color> .....	118,119
cmyk().....	119
cmyka().....	119
hsl().....	119
hsla().....	119
k100.....	119,129
rgba().....	22-25,119,130
color-compression.....	122
color-conversion .....	120
color-downsampling.....	125
color-downsampling-above-dpi.....	125
color-downsampling-target-dpi.....	125
color-jpeg-quality.....	122
axf:column-fill.....	102
axf:column-number-format.....	151
axf:column-rule.....	35-41
axf:column-rule-align.....	35-41,103
axf:column-rule-color.....	102
axf:column-rule-display.....	103
axf:column-rule-length.....	103
axf:column-rule-style.....	35-41,102
axf:column-rule-width.....	35-41,102
axf:condensed-text-align-last.....	192
condensed-text-align-last.....	192
axf:counter-style.....	42-45,211
axf:crop-area-visibility.....	322,334-336
axf:crop-offset.....	322
axf:crop-offset-bottom.....	322,330-336
axf:crop-offset-left.....	322,330-336

axf:crop-offset-right..... 322,330-336  
 axf:crop-offset-top..... 322,330-336  
 <axf:custom-property>..... 263

## D

dcm.....56,156  
 dd..... 56  
 default-CJK.....220  
 default-color.....219  
 default-font-size.....219  
 axf:destination-type ..... 275  
 axf:diagonal-border-color..... 73,110-113  
 axf:diagonal-border-style..... 73,110-113  
 axf:diagonal-border-width..... 73,110-113  
 display-align..... 110-113  
   justify..... 108  
 axf:display-alttext ..... 316  
 display-alttext..... 316  
 <axf:document-info>..... 258  
   author.....259  
   author-title..... 259  
   centerwindow..... 262  
   copyright-info-url..... 261  
   copyright-notice.....261  
   copyright-status..... 261  
   createdate.....259  
   description-writer..... 259  
   displaydoctitle.....262  
   document-title.....259  
   hidemenubar..... 262  
   hidetoolbar.....262  
   hidewindowui..... 262  
   keywords.....259  
   modifydate.....259  
   openaction.....262

pagelayout..... 26-30,262  
 pagemode.....262  
 subject.....259  
   xmp.....261  
 axf:document-info-include..... 278  
 dpi..... 57,124,125

## E

emu..... 56  
 emulated-italic..... 232  
 emulated-small-caps..... 232  
 encryption-level..... 253  
 encrypt-metadata.....253  
 EPS-processor.....128  
 error-if-overflow..... 187  
 error-on-pdfx-fault.....251  
 ex.....56  
 <exceptions>..... 175-177  
 axf:except-non-end-of-line-characters.....179  
 except-non-end-of-line-characters ..... 179  
 axf:except-non-starter-characters.....179  
 except-non-starter-characters ..... 179  
 Extensible Metadata Platform.....261

## F

fallback-glyph.....231  
 family-name-syntax.....230  
 axf:field-button-face..... 303  
 axf:field-button-face-down..... 303  
 axf:field-button-face-rollover..... 303  
 axf:field-button-icon..... 303  
 axf:field-button-icon-down..... 303  
 axf:field-button-icon-rollover..... 303  
 axf:field-button-layout..... 303  
 axf:field-checked..... 305

axf:field-checked-style.....	305	axf:float-float-margin-y.....	137
axf:field-default-text.....	50-53,298	axf:float-margin-x.....	42-45,136
axf:field-description.....	296	axf:float-margin-y.....	42-45,136
axf:field-editable.....	308	axf:float-min-wrap-x.....	135
axf:field-flags.....	297	axf:float-min-wrap-y.....	135
axf:field-font-size.....	295	axf:float-min-x.....	42-45
axf:field-format.....	50-53,300	axf:float-move.....	42-49,133
axf:field-format-category.....	50-53,299	axf:float-offset-x.....	97-100,137
axf:field-lock-document.....	50-53,309	axf:float-offset-y.....	35-41,137
axf:field-maxlen.....	50-53,299	axf:float-reference.....	42-45,132
axf:field-multiline.....	50-53,302	axf:float-wrap.....	135
axf:field-multiple.....	307	axf:float-x.....	35-45,97-100,134
axf:field-name.....	50-53,295	alternate.....	103
axf:field-name-suffix-page-number.....	295	axf:float-y.....	22-25,35-45,134,147,148
axf:field-readonly.....	50-53,297	axf:flush-zone .....	166
axf:field-required.....	298	<font-alias>.....	233
axf:field-scroll.....	50-53,302	<axf:font-face>.....	229
axf:field-selected.....	307	axf:font-feature-settings.....	228
axf:field-submit-coordinates.....	310	font-size-adjust	
axf:field-submit-method.....	310	<string> .....	220
axf:field-text-align.....	50-53,298	font-stretch.....	97
axf:field-top-index.....	307	<number>.....	218
axf:field-type.....	50-53,294	<percentage>.....	218
checkbox.....	305	font-stretch-mode.....	218
combobox.....	307	font-variant	
listbox.....	307	<font-variant-alternates>.....	221
radio.....	305	annotation().....	226
signature.....	309	character-variant().....	226
text.....	298	historical-forms.....	226
axf:field-value.....	295,306,308	ornament().....	226
fixed-width-space-treatment.....	207	styleset().....	226
axf:float.....	131	stylistic().....	226
axf:float-centering-x.....	35-45,136	swash().....	226
axf:float-centering-y.....	136	<font-variant-caps>.....	221,222
axf:float-float-margin-x.....	137	<font-variant-east-asian>.....	221

<east-asian-variant-values> .....	227	grayscale-downsampling-target-dpi.....	125
jis04.....	26-30	grayscale-jpeg-quality.....	122
<east-asian-width-values> .....	227	<GS1-128>.....	343
ruby.....	26-30,246,247	gs-options.....	129
<font-variant-ligatures>.....	221,225,226		
<font-variant-numeric>.....	221,224	<b>H</b>	
<font-variant-position>.....	221,227	hair-space-width.....	208
axf:footnote-align .....	144	axf:headers.....	317
axf:footnote-keep.....	145	<hyphen>.....	175-177
axf:footnote-max-height .....	146	axf:hyphenate-caps-word.....	178
<axf:footnote-number>.....	139	axf:hyphenate-hyphenated-word.....	174
<axf:footnote-number-citation>.....	139	<hyphenation-info>.....	175
axf:footnote-number-format.....	140	<axf:hyphenation-info>.....	176
axf:footnote-number-initial.....	139	hyphenation-keep-mode.....	174
axf:footnote-number-reset.....	139	axf:hyphenation-minimum-character-count.	173
axf:footnote-position.....	42-45,141	HyphenationOption.....	176
axf:footnote-stacking.....	145	hyphenation-push-character-count	
force-page-count		auto.....	177
doubly-even.....	64	hyphenation-remain-character-count	
doubly-even-document.....	64	auto.....	177
end-on-doubly-even.....	64	hyphenation-TeX.....	176
even-document.....	64	axf:hyphenation-zone.....	173
odd-document.....	64	<hyphen-char>.....	175-177
<axf:form> .....	294	<hyphen-min>.....	175
<axf:formatter-config>.....	341	hyphen-min.....	177
<axf:form-field> .....	50-53,276,294		
<axf:form-field-event> .....	50-53,311	<b>I</b>	
<axf:field-type>.....	276	ic.....	56
		axf:image-resolution.....	123
<b>G</b>		axf:image-smoothing.....	124
ghostscript.....	128	import-annotation-type	
gr.....	56	3D.....	272
grayscale-compression.....	122	All.....	272
grayscale-downsampling.....	125	Caret.....	272
grayscale-downsampling-above-dpi.....	125	Circle.....	272

FileAttachment.....	272	axf:inline-overflow-align .....	26-30,194
FreeText.....	272	internal-destination	
Highlight.....	272	<number-with-fragment>.....	262
Ink.....	272	issue-scale-to-fit.....	121
Line.....	272		
Link.....	272	<b>J</b>	
Movie.....	272	jamo-ligature.....	225
Other.....	272	joboptions.....	128
Polygon.....	272	jpt.....	56
PolyLine.....	272	justify-leader.....	115
Popup.....	272	axf:justify-nbsp.....	208
Richmedia.....	272	justify-rowspan-height.....	109
Screen.....	272		
Sound.....	272	<b>K</b>	
Square.....	272	axf:kansuji-grouping-letter.....	211
Squiggly.....	272	axf:kansuji-letter.....	210
Stamp.....	272	axf:kansuji-style.....	26-30,210
StrikeOut.....	272	axf:keep-together-within-dimension.....	195
Text.....	272	axf:keep-together-within-inline-dimension..	195
Underline.....	272	axf:kerning-mode.....	221
axf:indent-here.....	181		
<axf:index-page-citation-range-ff-suffix> ...	154	<b>L</b>	
<axf:index-page-citation-range-f-suffix> ....	154	latin-ligature.....	224
axf:initial-letters.....	35-41,156,159	axf:layer.....	255
axf:initial-letters-color.....	156	axf:layer-settings.....	254
axf:initial-letters-end-indent.....	35-41,159	leader-alignment	
axf:initial-letters-first-line-head-height.....		center.....	114
.....	35-41,158	end.....	114
axf:initial-letters-leading-punctuation.....	160	start.....	114
axf:initial-letters-leading-punctuation-position..		axf:leader-expansion.....	115
.....	161	axf:left-page-master-reference .....	22-25,61
axf:initial-letters-leading-punctuation-shift..	161	axf:letter-spacing-side.....	203
axf:initial-letters-text-align.....	157	lh.....	56
axf:initial-letters-width.....	157	axf:ligature-mode.....	224
axf:initial-volume-number.....	278,282-285	linear-gradient().....	31-34,92,126



no-accessibility.....	319
no-content-copying.....	254
no-disp-warnings.....	342
no-fill-form.....	312
<non-eol-words>.....	175
non-starter-ideographic-space .....	180
axf:normalize.....	212
normalize.....	212
axf:normalize-exclude.....	212
axf:number-transform.....	26-30,42-45,209
axf:number-type .....	151

## O

object-compression.....	252
oblique-skew.....	229
axf:origin-id .....	153
axf:outline-color.....	281
axf:outline-expand.....	281
axf:outline-external-destination.....	280
axf:outline-font-style.....	281
axf:outline-font-weight.....	281
axf:outline-group.....	279
axf:outline-internal-destination.....	280
axf:outline-level.....	279,282-285
axf:outline-title.....	279,282-285
axf:output-volume-break.....	277
axf:output-volume-filename.....	278
<axf:output-volume-info>.....	277,282-285
overflow	
condense.....	186
replace.....	186
axf:overflow-align.....	194
axf:overflow-condense.....	31-34,189
axf:overflow-condense-limit-font-size.....	190
axf:overflow-condense-limit-font-stretch.....	191

axf:overflow-condense-limit-letter-spacing..	191
axf:overflow-condense-limit-line-height.....	191
axf:overflow-limit.....	187
axf:overflow-limit-block.....	187
overflow-limit-block.....	188
axf:overflow-limit-inline.....	188
overflow-limit-inline.....	188
axf:overflow-replace.....	188
axf:overprint.....	327
overprint.....	327
owner-password.....	253

## P

<fo:page-sequence>.....	64,66-68
pair-kerning.....	221
pass-through.....	123
PDF/A.....	250
PDF/UA.....	250
PDF/X.....	250
axf:pdftag.....	314
pdf-version.....	251
axf:physical-page-number .....	152
axf:poster-content-type.....	288,293
axf:poster-image.....	288,293
axf:printer-bin-selection.....	328
axf:printer-duplex.....	329
axf:printer-marks.....	323,330-336
axf:printer-marks-line-color.....	325,334-336
axf:printer-marks-line-length.....	326,334-336
printer-marks-line-length.....	326
axf:printer-marks-line-width.....	325,334-336
printer-marks-line-width.....	325
axf:printer-marks-spine-width.....	325,330-333
axf:printer-marks-zero-margin.....	326
printer-marks-zero-margin.....	326

printing-allowed.....328  
 axf:punctuation-spacing.....201  
 punctuation-spacing.....201  
 axf:punctuation-trim.....26-30,200  
 punctuation-trim.....200  
 pvh.....56  
 pvmax.....56  
 pvmin.....56  
 pvw.....56  
 pxpi.....58

## Q

q.....56

## R

radial-gradient().....126,130  
 rasterize-resolution.....121  
 real-value-limit.....256  
 real-value-limit-modify.....257  
 rem.....56,59  
 axf:repeat-cell-content-at-break .....106  
 axf:repeat-footnote-in-table-footer .....106  
 axf:repeat-footnote-in-table-header .....105  
 repeating-linear-gradient().....50-53,127  
 repeating-radial-gradient().....127  
 axf:repeat-page-sequence-master.....66  
 responsive-svg-size.....126  
 axf:retrieve-table-rows.....109  
 axf:reverse-diagonal-border-color....73,110-113  
 axf:reverse-diagonal-border-style....73,110-113  
 axf:reverse-diagonal-border-width...73,110-113  
 axf:reverse-page.....65  
 reverse-page.....256  
 axf:reverse-page-number .....151  
 axf:right-page-master-reference .....22-25,61

rlh.....56,59  
 <axf:ruby>.....26-30,234,244-247  
 axf:ruby-align.....237,244-246  
 ruby-align.....237  
 <axf:ruby-base>.....26-30,234,244-247  
 axf:ruby-color.....235  
 axf:ruby-condense.....26-30,239,244-246  
 axf:ruby-font-family.....235  
 axf:ruby-font-size.....236,246,247  
 axf:ruby-font-stretch.....236,244-247  
 axf:ruby-font-style.....235  
 axf:ruby-font-weight.....236,246,247  
 axf:ruby-limit-overhang.....239,244-246  
 axf:ruby-limit-space.....244-246  
 axf:ruby-minimum-font-size.....26-30,236  
 axf:ruby-offset.....238,244-247  
 axf:ruby-overhang.....244-246  
 axf:ruby-positon.....238,244-246  
 axf:ruby-small-kana.....240  
 <axf:ruby-text>.....26-30,234,244-247

## S

axf:scope.....317  
 <script-font>.....230  
 SeparatePrinterDuplexJob.....329  
 axf:show-controls.....289  
 <fo:simple-page-master>.....68,69  
 size.....58,59  
 small-caps-emulation-always.....223  
 small-caps-emulation-size .....223  
 small-caps-emulation-x-height.....223  
 axf:soft-hyphen-treatment.....182  
 <axf:space-between-digit-and-punctuation>196  
 <space-between-digit-and-punctuation>.....196  
 <axf:space-between-punctuation-and-digit>196

<space-between-punctuation-and-digit>.....	196	afx:tyext-align-string.....	112
<afx:space-end-punctuation>.....	196	text-and-lineart-compression.....	252
<space-end-punctuation>.....	196	afx:text-autospace.....	198
<afx:space-start-punctuation>.....	196	text-autospace.....	198
<space-start-punctuation>.....	196	afx:text-autospace-width.....	198
<afx:spread-page-master> .....	22-25,61	text-autospace-width.....	198
<afx:spread-region> .....	22-25,61	afx:text-combine-horizontal .....	215
afx:suppress-duplicate-footnote .....	145	afx:text-emphasis-color.....	242
afx:suppress-duplicate-marker-contents .....	149	afx:text-emphasis-font-size.....	242
afx:suppress-folio-prefix .....	154,282-285	afx:text-emphasis-font-stretch.....	243
afx:suppress-folio-suffix .....	154	afx:text-emphasis-font-style.....	242
afx:suppress-if-first-on-page.....	185	afx:text-emphasis-font-weight.....	243
<b>T</b>			
<afx:tab>.....	205,282-285	afx:text-emphasis-offset.....	244
afx:tab-align.....	205,282-285	afx:text-emphasis-skip.....	243
afx:table-auto-layout-limit .....	107	afx:text-emphasis-style.....	241
table-auto-layout-limit.....	108	afx:text-indent-if-first-on-page .....	164
<afx:table-cell-repeated-marker> .....	106	afx:text-justify.....	204
table-omit-footer-at-break		text-justify-mode.....	202
column.....	105	afx:text-justify-trim.....	202
table-omit-header-at-break		afx:text-kashida-space.....	204
column.....	105	text-kashida-space.....	204
afx:table-summary.....	318	afx:text-line-color.....	87
afx:tab-overlap-treatment.....	206	afx:text-line-style.....	87
tab-overlap-treatment.....	206	afx:text-line-width.....	87
afx:tab-stops.....	205	afx:text-orientation.....	214
afx:tab-treatment.....	206	text-orientation-mode.....	215
tag-base-name.....	315	afx:text-overflow.....	193
tag-derived-name.....	315	afx:text-replace.....	214,216,217
<tag-element>.....	315	text-shadow.....	31-34,90
tagged-pdf.....	314	textshadow-blur-cannot-embed-font.....	90
<tag-role-map>.....	315	textshadow-resolution-minimum-dpi.....	90
afx:text-align-first.....	35-41,165	textshadow-resolution-pixel-per-em.....	90
afx:text-align-string.....	108	afx:text-stroke.....	22-25
		afx:text-stroke-color.....	22-25

axf:text-stroke-width.....	22-25
text-transform	
capitalize-lowercase.....	213
fullsize-kana.....	26-30,213
fullwidth.....	213
fullwidth-if-vertical.....	31-34,213
axf:text-underline-position.....	88
thin-space-width.....	208
axf:transform.....	31-34,74
matrix().....	74,75
rotate().....	74,75,344
scale().....	74,75
scaleX().....	74,75,95-97
scaleY().....	74,75
skew().....	74,75
skewX().....	74,75,95-97
skewY().....	74,75
translate().....	74,75
translateX().....	74,75,95-97
translateY().....	74,75,95-97,344
axf:transform-origin.....	31-34,76
transparency.....	120
transparency-color-space.....	120

## U

<unbreakable-words>.....	172
use-launch-local-file.....	274
user-password.....	253

## V

axf:vertical-underline-side .....	89
vertical-underline-side.....	89
vh.....	56
viewport-length-units-mode.....	57
vw.....	56

## W

watermark2-font-family.....	337
watermark2-font-style.....	338
watermark2-font-weight.....	338
watermark2-text.....	337
watermark-font-family.....	337
watermark-font-style.....	338
watermark-font-weight.....	338
watermark-text.....	337
axf:word-break.....	171
axf:word-wrap.....	172

## X

XMP.....	261
xmp-padding.....	261

## アンテナハウス株式会社

---

多言語・大容量の書籍、マニュアルなど、複雑なレイアウトを XSL-FO・XML/HTML と CSS によって組版可能な、組版・印刷ソフトウェア「Antenna House Formatter」を世界中で販売しています（XSL-FO と CSS を混在させることはできません）。PDF や XML をはじめとするドキュメント関連のデータ有効活用するためのコンピュータソフトを企画・開発・販売する会社です。XML 技術の啓蒙と技術者育成の支援も行っています。

## Antenna House XSL Formatter 拡張仕様使いこなしガイド

---

---

2021 年 5 月 11 日 初版

著 者 アンテナハウス株式会社  
発 行 所 アンテナハウス株式会社 CAS 電子出版  
住 所 東京都中央区東日本橋 2 丁目 1 番 6 号  
電話番号 03-5829-9021  
W E B <https://www.antenna.co.jp>  
E メール [cas-info@antenna.co.jp](mailto:cas-info@antenna.co.jp)

---

---

Copyright 2021 Antenna House, Inc.

ISBN978-4-900552-82-1