Altova StyleVision 2021 Basic Edition

Altova StyleVision 2021 Basic Edition ユーザーマニュアル

All rights reserved. No parts of this work may be reproduced in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

公開日: 2015-2021

(C) 2015-2021 Altova GmbH

目次

1	はじめに	13
1.1	新規機能: バージョン 2021	
1.2	製品機能	
1.3	Altova 製品における Authentic View	
1.4	SPS とは?	
1.5	StyleVision のセットアップ	
1.6	用語解説	
1.7	このドキュメントについて	25
2	ユーザーインターフェイス	27
2.1	メインウィンドウ	28
	2.1.1 デザインビュー	29
	2.1.2 出力ビュー	29
2.2	サイドバー	
	2.2.1 デザイン概要	
	2.2.2 スキーマツリー	
	2.2.3 デザインツリー	
	2.2.4 スタイルレポジトリ	
	2.2.5 スタイル	44
	2.2.6 プロパティ	45
	2.2.7 メッセージ	49
	2.2.8 検索と置換	49
3	クイックスタートチュートリアル	51
3.1	新規 SPS の作成	52
3.2	(XML ソースの)動的なコンテンツの挿入	56
3.3	静的コンテンツの挿入	
3.4	コンテンツのフォーマット	68

3.5	自動計	↑算の使用	74
3.6	条件の)使用	78
3.7	グロー	-バルテンプレートとその他のコンテンツの使用	85
3.8	おしま	lv!	90
4	使用	方法の概要	91
4.1	SPS d	とソース	92
4.2	デザインの作成		93
4.3	XSLT	と XPath のバージョン	94
4.4	Interne	et Explorer の互換性	95
4.5	生成さ	れたファイル	97
4.6	StyleV	/ision におけるカタログ	98
4.7	StyleV	/ision 内のカタログ	102
	4.7.1	カタログの仕組み	102
	4.7.2	StyleVision 内のカタログ構造	103
	4.7.3	カタログのカスタム化	104
	4.7.4	Windows システムロケーションのための変数	105
5	SPS	コンテンツ	107
5.1	テキストとして XML コンテンツを挿入		108
	5.1.1	定義済みのフォーマットとともにコンテンツを挿入する	
	5.1.2	その他のコンテンツ	111
5.2	MS Wo	ord コンテンツの挿入	112
5.3		cel コンテンツの挿入	
5.4	ユーザ	・ ・一定義テンプレート	117
5.5	ユーザ	・ ・一定義要素、XML テキストブロック	120
	5.5.1	ユーザー定義要素	120
	5.5.2	ユーザー定義 XML テキストブロック	121
5.6	テーブ	`ル	123
	5.6.1	静的テーブル	124
	5.6.2	動的テーブル	126
	5.6.3	テーブル内の条件付き処理	130
	5.6.4	デザインビュー内のテーブル	131

	5.6.5	テーブルのフォーマット	133
	5.6.6	行と列の表示	137
	5.6.7	CALS/HTML テーブル	138
5.7	リスト		144
	5.7.1	静的なリスト	144
	5.7.2	動的なリスト	146
5.8	イメーシ	>	149
	5.8.1	イメージ: URI とインラインデータ	149
	5.8.2	イメージ型 と出力	151
	5.8.3	サンプル: イメージのテンプレート	153
5.9	フォーム	ムコントロール	154
	5.9.1	入力フィールド、複数行の入力フィールド	155
	5.9.2	チェックボックス	155
	5.9.3	コンボボックス	157
	5.9.4	ラジオボタン、ボタン	157
5.10	リンク		158
5.11	バーコー	− ド	159
5.12	レイアウ	フト モジュール	163
	5.12.1	レイアウトコンテナー	163
	5.12.2	レイアウトボックス	166
	5.12.3	ライン	170
5.13	変更機	能	172
6	SPS	構造	175
6.1	スキー	マソース	177
	6.1.1	DTD と XML スキーマ	178
	6.1.2	ユーザー定義スキーマ	183
6.2	複数の	ソースから XML データのマージ	186
6.3	モジュー	ール SPS	189
	6.3.1	利用可能なモジュールオブジェクト	190
	6.3.2	SPS の作成	193
	6.3.3	サンプル: Address Book	197
6.4	テンプレ	ンートと デザインフラグメント	202
	641	メインのテンプレート	202

	6.4.2	グローバルテンプレート	203
	6.4.3	ユーザー定義テンプレート	206
	6.4.4	変数テンプレート	209
	6.4.5	ノードテンプレート操作	209
	6.4.6	デザインフラグメント	212
6.5	XSLT	テンプレート	216
6.6	複数σ) ドキュメントの出力	218
	6.6.1	新規 ドキュメントテンプレートの挿入	219
	6.6.2	新規ドキュメント テンプレートと デザイン構造	220
	6.6.3	新規ドキュメント テンプレートの URL	220
	6.6.4	プレビューファイルと出力 ドキュメントファイル	223
	6.6.5	ドキュメント プロパティと スタイル	225
7	高度	な機能	227
7.1	自動計	†算	228
	7.1.1	・・・ 自動計算の編集と移動	
	7.1.2	サンプル:注文書	230
7.2	条件		233
	7.2.1	条件のセットアップ	
	7.2.2	条件の編集	236
	7.2.3	条件と自動計算	237
7.3	グルー	-プ化	238
	7.3.1	サンプル: group-by(Persons.sps)	240
	7.3.2	サンプル: group-by(Scores.sps)	242
7.4	並べ替	替え	245
	7.4.1	並べ替えのメカニズム	245
	7.4.2	サンプル: 複数のソートキーによる並べ替え	247
7.5	パラメ	ーターと変数	251
	7.5.1	ユーザー宣言パラメーター	251
	7.5.2	デザインフラグメントに対するパラメーター	252
	7.5.3	ソースのための SPS パラメータ	255
	7.5.4	変数	255
7.6	目次、	参照、ブックマーク	258
	7.6.1	目次に含まれるブックマークアイテム	260

	7.6.2	目次テンプレートの作成	267
	7.6.3	サンプル: シンプルな目次	272
	7.6.4	サンプル: 階層的ならびに連続的な目次	275
	7.6.5	ドキュメントボディー上の自動番号	279
	7.6.6	相互参照	282
	7.6.7	ブックマーク と ハイパーリンク	284
8	表示	方法と出力	290
8.1	定義済	「 「 「 「 「 「 「 「 「 「 」 「 「 」 「 」 「	291
8.2)エスケープ	
8.3	値フォ	ーマット (数値データ型のフォーマット)	295
	8.3.1	値フォーマットのメカニズム	295
	8.3.2	値フォーマットの構文	298
8.4	CSS 7		304
	8.4.1	外部スタイルシート	305
	8.4.2	グローバルスタイル	308
	8.4.3	ローカルスタイル	310
	8.4.4	スタイルの値の設定	312
	8.4.5	XPath を使用したスタイルプロパティ	315
	8.4.6	合成スタイル	317
9	追加	機能	321
9.1	パース	されていないエンティティ URI	322
9.2		. XSL-FO または FO ファイルからの新規作成	
9.3		·一定義 XPath 関数	
	9.3.1	XPath 関数の定義	
	9.3.2	ノードを検索するための関数の再利用	331
	9.3.3	XPath 関数内のパラメーター	332
9.4	日付の)使用	341
	9.4.1	- 日付のフォーマット	341
9.5	スクリ	プトの使用	344
	9.5.1	JavaScript 関数の定義	
	9.5.2	関数をイベントハンドラーとして割り当てる	346

	9.5.3	外部 JavaScript ファイル	347
9.6	HTML	のインポート	349
	9.6.1	HTML のインポートにより新規 SPS を作成	349
	9.6.2	スキーマと SPS デザインの作成	351
	9.6.3	要素/属性としてテーブルやリストを作成	353
	9.6.4	出力の生成	355
9.7	ASPX -	インターフェイスによる Web アプリケーション	356
	9.7.1	サンプル: Windows 7 にて動作する localhost	357
9.8	PXFフ	ァイル:SPS とその関連ファイルのコンテナー	359
	9.8.1	PXF ファイルの作成	359
	9.8.2	PXF ファイルの編集	363
	9.8.3	PXF ファイルのデプロイ	364
10	処理(の自動化	366
10.1	コマント	· ・ライン インターフェイス	367
	10.1.1		
	10.1.2	StyleVision Server	368
10.2	Raptor	XML の使用	370
	10.2.1	PDF 出力	
10.3	FlowFo	rce Server を使用した自動化	372
10.4	自動処理の方法		374
11	メニュ	∟ ーコマンドと レファレンス	375
11.1	デザイ:	ン ビューのシンボル	376
11.2	XPath :	式 の編集ダイアログ	380
	11.2.1	エバリュエーター	381
	11.2.2	デバッガ—	384
	11.2.3	式ビルダー	391
11.3	ツール	バー	395
	11.3.1	書式	397
	11.3.2	テーブル	397
	11.3.3	デザイン要素の挿入	399
	11.3.4	デザインフィルター	401

	11.3.5	標準	402
11.4	ファイル	・メニュー	404
	11.4.1	新規作成	404
	11.4.2	開く、再ロード、閉じる、すべて閉じる	410
	11.4.3	デザインを保存, すべて保存	415
	11.4.4	名前を付けて保存	420
	11.4.5	MobileTogether デザインファイルとしてエクスポートする	421
	11.4.6	生成されたファイルの保存	421
	11.4.7	FlowForce へのデプロイ	422
	11.4.8	Web デザイン	424
	11.4.9	プロパティ	424
	11.4.10	印刷プレビュー、印刷	425
	11.4.11	最後に開かれたファイル、終了	426
11.5	編集メニ	<u>-</u>	427
	11.5.1	元に戻す、やり直し、すべて選択	427
	11.5.2	検索、次を検索、置き換え	427
	11.5.3	スタイルシートのパラメーター	432
	11.5.4	マークアップの縮退/展開	432
11.6	表示メニ		434
	11.6.1	ツールバーとステータスバー	434
	11.6.2	デザインサイドバー	435
	11.6.3	デザインフィルター、ズーム	435
11.7	挿入メニ		437
	11.7.1	コンテンツ	437
	11.7.2	その他のコンテンツ	438
	11.7.3	フォームコントロール	438
	11.7.4	自動計算	439
	11.7.5	段落、特殊な段落	440
	11.7.6	イメージ	441
	11.7.7	水平線	443
	11.7.8	テーブル	444
	11.7.9	リスト	
	11.7.10	ブックマーク	446
	11.7.11	ハイパーリンク	447
	11.7.12	条件、出力ベースの条件	449

	11.7.13	テンプレート	450
	11.7.14	ユーザー定義テンプレート	451
	11.7.15	変数テンプレート	452
	11.7.16	デザインフラグメント	453
	11.7.17	レイアウトコンテナー、レイアウトボックス、ライン	453
	11.7.18	目次	453
	11.7.19	新規ドキュメント	454
	11.7.20	ユーザー定義アイテム	454
11.8	タグで囲	目むメニュー	456
	11.8.1	テンプレート	456
	11.8.2	ユーザー定義テンプレート	457
	11.8.3	変数テンプレート	457
	11.8.4	段落、特殊な段落	458
	11.8.5	リスト	458
	11.8.6	ブックマーク と ハイパーリンク	459
	11.8.7	条件、出力ベースの条件	460
	11.8.8	目次ブックマークと目次レベル	461
	11.8.9	新規ドキュメント	461
	11.8.10	ユーザー定義要素	462
11.9	テーブル	レメニュー	463
	11.9.1	テーブルの挿入、テーブルの削除	463
	11.9.2	テーブルヘッダー/フッターの追加	464
	11.9.3	行/列の追加と挿入	464
	11.9.4	行/列の削除	465
	11.9.5	左/右/下/上のセルを結合	465
	11.9.6	水平方向/垂直方向にセルを分割	466
	11.9.7	セルの境界とテーブル マークアップの表示	466
	11.9.8	テーブル プロパティ	467
	11.9.9	CALS/HTML テーブルの編集	467
	11.9.10	セル コンテンツの高さ揃え	468
11.10	プロパラ	ティメニュー	470
	11.10.1	リストの編集	470
	11.10.2	定義済みの値フォーマット文字列	471
11.11	ツールメ	<	473
	11.11.1	スペルチェック	473

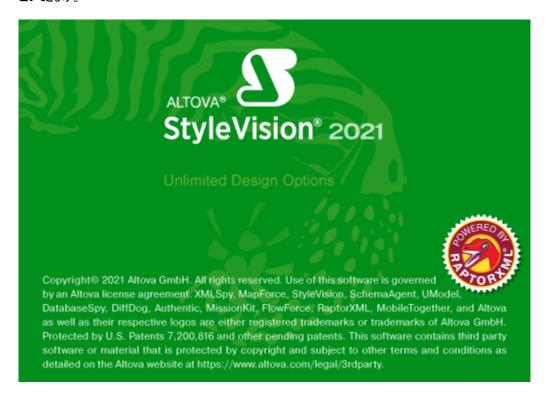
	11.11.2	スペル チェックのオプション	474
	11.11.3	カスタマイズ	477
	11.11.4	ツールバーとウィンドウの復元	482
	11.11.5	オプション	482
11.12	ウィンド	ウ メニュー	485
11.13	ヘルプノ	ـــــــــــــــــــــــــــــــــــــ	486
	11.13.1	目次、インデックス、検索	486
	11.13.2	ライセンス登録、注文フォーム、レジストレーション、最新情報のチェック	486
	11.13.3	他のコマンド	
12	付録		491
12.1	XSLT お	3よび XQuery エンジンに関する情報	492
	12.1.1	XSLT 1.0	492
	12.1.2	XSLT 2.0	492
	12.1.3	XSLT 3.0	494
	12.1.4	XQuery 1.0	494
	12.1.5	XQuery 3.1	497
12.2	XSLT &	: XPath/XQuery 関数	498
	12.2.1	Altova 拡張関数	499
	12.2.2	その他の拡張関数	571
12.3	DB によ	り生成された XML スキーマ内のデータ型	588
	12.3.1	ADO	588
	12.3.2	MS Access	589
	12.3.3	MS SQL Server	589
	12.3.4	MySQL	590
	12.3.5	ODBC	591
	12.3.6	Oracle	592
	12.3.7	Sybase	592
12.4	技術デ-	ータ	594
	12.4.1	OS とメモリ要件	594
	12.4.2	Altova XML バリデーター	
	12.4.3	Altova XSLT と XQuery エンジン	594
	12.4.4	Unicode のサポート	594
	12 4 5	インターネットの使用	595

12.5	ライセン	ノス情報	596
	12.5.1	電子的なソフトウェアの配布	596
	12.5.2	ソフトウェアのアクティベーションとライセンスの計測	596
	12.5.3	エンドユーザー使用許諾契約書	597
イン	デック	ス	598

はじめに 13

1 はじめに

Altova StyleVision 2021 Basic Edition は、StyleVision Power Stylesheet の編集を視覚的に行うはのアプケーションで。StyleVision®は、プラナフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10 とプラナフォーム更新済みのWindows Server 2008 R2 SP1 まけは以降で作動します。StyleVisionとAltova MissionKit の機能の一部は、Microsoft Office パッケージ(MS Access、MS Excel、MS Word)、バージョン 2007 まけは以降のアプケーションと統合することができます。



StyleVision Power Stylesheet (SPS) を以下の用途に使用することができます:

- XML ドキュメント をW YSIW YG ビューの Authentic View にて管理します。Authentic View は、Altova XMLSpy、Altova Style Vision、Altova Authentic Desktop とAltova Authentic Browserにて利用することができます。この機能により、XML ドキュメトをベース こた<u>電子フォーム</u>を簡単に作成することができます。
- SPS デザインをベースユたXSLT スタイルシートを生成します(XSLT 1.0、XSLT 2.0、とXSLT 3.0 がサポーされます)。XSLT スタイルシートは Style Vision の外部で XML ドキュメントを HTMLのファイル形式へ変換するために使用することができます。
- Style Vision 内から、HTML の形式で出力を直接生成します。

最終更新日: 2021 年 02 月 25 日

14

1.1 新規機能: バージョン 2021

バージョン 2021 リリース2

はじめに

• <u>検索と置き換え</u>機能がデザインビューロ搭載されました。デザインテキスト、スタイル、プロイティ、XPath 式内で入力するまだは 正規表現を構成する文字列を検索することができます。

バージョン 2021

•

はじめに 製品機能 15

1.2 製品機能

StyleVision の主な製品機能は、以下にリストされるとおりです:

一般的な 製品機能

StyleVision (学報された、主な機能を以下に示します:

• Enterprise ならびに Professional Edition は32ビットならびに64ビット版がご利用になれます。

ソース

SPS はXML スキーマやDTD をベースにデザインすることができます。デザインコはXML やCSS ファイルという。、その他のノースファイルを使用することもできます。ソースに関して以下の機能を使用することができます:

•

• HTMLドキュメントをXMLに変換することができます。

インターフェイス

一般的な GUI の機能を以下に示します:

- 複数のSPS デザイン 複数のデザインを同時に開くことができ、そのうちの1つがアクティブな状態となります。各 SPS デザインが異なるタブで表示されます。
- <u>デンプレートフィルター</u> デザインドキュメントの外観をカスタマイズすることができます。 現在編集されていない デンプレートの表示を無効にし、編集の効率を向上させることができます。
- <u>デザインビューでのマーケアップを隠す</u>: デザインビュー内にあるタグを隠し、縮退させることで、デザインビューのスペースを有効活用することができます。
- SPS のデザインを行う際に、対応するタブをクリックすることで、出力ビュー、そしてスタイルシートを表示することが できます。この機能にお出力やXSLTコードを素早く確認し、をテストすることができます。

出力

インストールされたエディションにより、様々な出力フォーマナがサポーされます。出力に関連する以下の機能がサポーされます

- XSLT バージョン 1.0、2.0、と3.0 のドージョンがサポートされます。
- Enterprise とProfessional Editions では、複数の出力フォーマナ (HTML) がサポーされ、単一のSPS デザインからこれらのファイルを生成することができます。
- XSLT ファイルと出力ファイルの両方をGUI から直接、ませまStyleVision Server から、生成、保存することができます。
- Altova により開発された PXF ファイルフォーマナ を使用することで、SPS ファイルを関連するノースならびにデータファイル とともに保存することができます。この形式により、SPS ファイル単体ではなく、SPS プロジェナト全体を簡単に受け渡すことができるようになります。
- ASPX インターフェイスによるWeb アプリケーション. この機能により、HTML ウェブページの更新を簡単に行うことができるようになります。 StyleVision では SPS から ASPX アプリケーションに必要な全てのファイルが生成されます。 ウェブページ(.aspx ファイル) か更新されると、XSLT により、更新箇所を含んだ)XML が動的にウェブページへ変換されます。

SPS デザイン機能

SPS のデザインに関する主な Style Vision の機能を以下に示します:

• SPS には、SPS 内に入力した <u>静的なテキスト</u> の他 <u>ソーズ・キュメト</u> から選択された<u>動的なテキスト</u>を含めることができます。

16 はじめに 製品機能

動的なコンテンツは、スキーマノース」あるノードをドラッグアイドロップすることでデザイン内に挿入することができます。デザイン要素(段落、リスト、イメージなど)を挿入した後に、スキーマソノーからXMLノードをデザイン要素へ割り当てることができます。

- 動的なコレテンツはデキスト、まけば、入力フィールド まけまコレポポックス といけらデータ入力デバイス として挿入することができます。
- <u>メインのテンプレート</u>により、<u>デザインの構造</u>が指定、管理されます。この構造は <u>グロー ジレテンプレート</u> と呼ばれるオプションのテンプレート要素にて修正することができます。
- グローバルテンプレートは個々のデータ型に対して作成することができ、各種の型に応じた処理を定義することができます。
- <u>複数のドキュメトの出力</u>: SPS により生成される出力を複数のドキュメントへ分割することができます。デザインコは新規ドキュメントテンプレートが作成され、その中にコンテンツが配置されます。各新規ドキュメントテンプレートにより、異なるドキュメントが出力されます。.
- ユーザー定義アンプレート: 指定されたXPath 条件式により、複数のアイテムコオしてテンプレートを生成することができます。これらアイテムは原子型の値割さまノードとなり、XPath 条件式により、より詳細なノードの指定を行うことができ、そのセクションコオする条件付けやフィリタリングを行うことができるようになります。
- ユーザー定義要素: この機能はデザイン内の任意の場所に(HTML、XSLT、XSL+O といか)表示言語要素を自由に挿入するために開発されました。
- ユーザー定義 XML テキストブロック: XML テキストブロックをデザイン内の任意の場所に挿入することで、これらブロックが、生成された XSLT スタイルシートの対応する場所にて生成されます。
- <u>デザインフラグメント</u> により SPS 内にあるテンプレートのモジューソルと再利用を行うことができるようになり、関数の使用と同じ様な方法により、複数の SPS からアクセスすることができるようになります(モジュー/レ SPS)を参照ください)。
- <u>SPS モジュール</u> 既にあるモジュールをその他の SPS モジュールへ追加することができ、ある SPS モジュールで定義されたオブジェケトを他のモジュールでも利用することができるようけないます。この機能により、複数の SPS にてモジュールオブジェケトを最利用することができるようけなり、メンテナンスを簡単に行えるようけないます。
- XSLT テンプレート: 生成されたスタイルシートへXSLT ファイルをインポートすることができます。XML インスタンスドキュメト内にあるノードがインポートされた。XSLT 内のテンプレートにマッチして、インポートされたテンプレートよ 提高し優先度のテンプレートが無し場合、インポートされたテンプレートが使用されます。更に、インポートされた。XSLT ファイル内部の名前付きテンプレートを、デザイン内部がら呼び出すこともできます。
- XSLT からの新規作成: SPS ファイルを XSLT まけは XSL FO ファイルから作成することができます。 XSLT 内にあるテンプレートの構造とスタイルが SPS にて作成され、 SPS コンポーネントを修正してコンテンンやフォーマットを SPS に追加することができます。
- ユーザー定義 XPath 関数: ユーザー定義のXPath 関数を作成し、ドキュメント内でXPath 関数の使用が許されている場所から、関数の呼び出しを行うことができます。
- レイアナコンテナー: レイアナコンテナーは、内部にデザイン要素を絶対位置指定により配置することができるブロックです。
- <u>青写真</u>: レイアナーカーフォームのイメージを配置することで、デザインの青写真として使用することができます。青写真を使用することで、既に存在するデザインを正確に再現することが可能になります。
- XMLドキュメトでは、データ構造を繰り返し作成することが簡単に行えます。例えば、ある部署には通常複数の従業員が勤務しています。この場合、各従業員のデータがデータ構造の中に収められ、その構造が従業員の数だけ繰り返されることではよっています。このような各データ構造に対する処理を一度定義して、関連するノード(この例では各従業員ノード)に対して適用することができます。。
- XSLT 2.0 と3.0 SPS 内に複数の内容のテーブルを挿入することができます。
- 繰り返し表示されるデータ構造を動的テーブルとして挿入することもできます。この機能により、テーブル構造に対してループを作成し、テーブル内部にて各行(まざまカラム)を生成することが可能になります。
- 繰り返される要素は、選択された(1つ以上の) ソートキーにより述べ替えを行うことができ、並べ替えられた要素セナが出力(HTML)。に送信されます。
- 変数: 変数をテンプレート上で宣言して、XPath 条件式により指定された値を受け取ることができます。これまで、変数の値は作成されたノードの選択に制限されていました。 バージョン 2010 にて導入された変数により、XPath 条件式を変数の値として指定することができるようになりました。
- 共通したデータコンテンツ(例えば属性値の共通値)やその位置をベースに、ノードの グループ化 を行うことがてきます。
- 条件アンプレート機能により、XML ドキュメトヤシステム環境という条件が満たされて時に、テンプレートのセトから処理する ものを選択することができます。この機能を使用することで、SPS ドキュメントの作成時に、ソースドキュメント内に含まれる情報や 未知の情報に対して条件を指定することができるようによります。利用することができる条件は、XPath 1.0 または XPath 2.0 条件式で扱うことのできる条件に限定されます。

はじめに 製品機能 17

• <u>自動計算</u>によりノースドキュメトから得られたデータを操作して、その結果を表示することができます。この機能は、数値に対して 計算処理を行うような状況(例えば主文書の合計金額を計算する)や、文字列(例えば イフンをスラッシュニ変更)に対して使 用することができ、コンテンソの生成を行うことができます。これらのデータ操作しま、XPath 条件式を使用することができます。 ネイ ティブ Java ならびに、NET 関数を、自動計算内の XPath 条件式にで使用することもできます。

- <u>イメージ</u>をデザイン内に挿入することができます。イメージのURI は静的(SPS 内で入力)ませば動的(ソースドキュメト内にあるノードから取得)、ませば静的と動的パトの組み合わせによい構成することができます。
- <u>インラインデータからのイメージ</u>: XML ドキュメト内のテキストとしてエンコードされた Base-16 や Base-64 から、イメージを生成することができます。この機能により、イメージをテキストとしてソース XML ドキュメント内に直接挿入することができるようしていました。
- 静的ならびに動的な、2種類の <u>リスト</u>を作成することができます <u>静的なリスト</u>では、各リストアイテムがSPS にて定義され、<u>動的なリスト</u>では、ノードがリストアイテムとして作成され、ノードに対応する全インスタンスの値がリストのアイテムとして作成されます。
- <u>静的と動的ないク</u>をデザイン内へ挿入することができます。ターゲル URI は静的(SPS 内で入力) おけは動的(ソースドキュメント内にあるノードから取得)、おけは静的と動的パートの組み合わせによい構成することができます。
- 静的<u>ブックマーク</u>を挿入することができます。これらブックマークは、ハイパーリンクのリンク先となるアンカーとして使用することができます。
- SPS 全体に対してパラメータを宣言することができます。パラメーターは名前と文字列の値により宣言され、SPS 内部にある
 XPath 条件式にて使用することができます。宣言したパラメーター値はデフォルト値として使用され、<u>StyleVision Server</u> にて
 与えられた値によりオーバーライドすることができます。
- <u>入力フォーマト</u> 入力フォーマト機能により、数値型のXML スキーマデータ型ノードのコンテンンを、その他の出力の表示を行 うためにフォーマトし直すことができます。入力フォーマト機能は、<u>自動計算の</u>結果に対しても使用することができます。
- <u>JavaScript 関数</u> %SPS%> にて使用することができ、ユーザー定義関数をHTML 出力にて使用することができるようになります。.
- GUI に用意されている数多くの <u>定義済み HTML フォーマル</u> を利用して、個々のSPS エンポーネントへ適用することが できます。
- 用意されている CSS テキストフォーマットやレイアウトプロパティを、<u>スタイルサイドバー</u>が適々のSPS ユンポーネントに対して適用することができます。
- 更に、SPSの グロー いなレベルにおける HTML セレクターに対して、または外部 CSS スタイルシートに対して CSS スタイルを定義することができます。これらスタイルルールは HTML 出力に対して適用され、フォーマットならびにレイアウトの柔軟性が大幅に拡張されます。
- Path 条件式を使ってもスタイルの割り当てを行うことができます。この機能により、XML ドキュメントからスタイルプログライの値を選択し、条件付きでプログライの値をセントすることができます。

□ 関連項目

- ユーザーインターフェイス
- 一般的な使用方法

Altova 製品における Authentic View 1.3

Authentic View は、次のAltova 製品で使用することのできるグラフィカルな XML ドキュメント エディターです:

- * Altova XMLSpy* Altova Authentic Desktop
- * Altova Authentic Browser
- * Altova StyleVision

はじめに SPS とは? 19

1.4 SPS とは?

StyleVision Power Stylesheet(またはSPS)は拡張されたXSLT スタイルシートです。HTML 出力ドキュメントのためにデザインをグラフィカルコ作成するためコ使用されます。

SPS は .sps ファイル拡張子で保存されます。

SPS のデザイン

SPS は、StyleVision 内部で視覚的に作成されます。スキーマ(DTD ませま XML スキーマ)をベースしており、。SPS のデザイノは柔軟性に富んでおり、動的ならびに静的なコンテンルを含めることができます。動的なコンテンツとは XML ドキュメト内にあるデータのことです。静的コンテンツとは SPS 内に直接入力されたコンテンのことです。動的コンテンツはテキストとして直接、ませは入力フィールドやコンボボックス、テーブルというコンポーネントの内部に含めることができます。更に、動的なコンテンツを(自動計算機能により)操作することもでき、ソースドキュメト内である条件が満たされた場合に表示するということも可能にないます。SPS では複数の場所に異なるコンテンルを配置することができ、イメージやソイパープンク、JavaScript 関数というこその他のコンポーネントを含めることもできます。SPSの各コンポーネントは必要に応じてフォーマナすることができます。

SPS とXSLT スタイルシート

SPS のデザインが完成したら、作成したデザインをベースユナミXSLT スタイルシートを生成することができます。StyleVision ではXSLT 1.0 とXSLT 2.0 とXSLT 3.0 がサポートされており、単一のSPS からHTML、RTF、XSL FO、そして2007 以降のW ord に対応した XSLT スタイルシートを生成することができます(XSL FO とW ord 2007 以降はEnterprise Edition でのチサポートされ、RTF はEnterprise とProfessional Edition で、HTML 出力はBasic Edition でのチサポートされます)。生成された XSLT スタイルシートにより、XSLT スタイルシートの生成に使用された SPS と同一の SPS をベース した XML ドキュメトの変換を行うことできます。 XSLT スタイルシートの使用方法に関する詳細は、生成されたファイルのセグションを参照ください。

SPS 出力

StyleVision では出力(Enterprise Edition ではHTML、RTF、XSL+O、PDF、Professional ならびにBasic Edision ではHTML)を直接生成することできます。出力ビューのタブにより、アクティブな SPS ドキュメールの出力を、StyleVision GUI にて直接表示することができます。目的の出力ファイルは、(i) StyleVision GUI の「ファイル 生成されたファイルの保存」コマンド まだは StyleVision Server コマンドを呼び出すことで生成することができます。

20 はじめに SPS とは?

Altova 製品内のAuthentic View

Authentic View は、次のAltova 製品で使用することのできるグラフィカルな XML ドキュメント エディターです:

- * Altova XMLSpy
- * Altova Authentic Desktop
- * Altova Authentic Browser
- * Altova StyleVision

□ 関連項目

• 一般的な使用方法

1.5 StyleVision のセットアップ

Altova StyleVision は プラナフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10 上で動作します。Altova Web サ小 から StyleVision をダウムードした後に、実行ファイル(.exe) をダブルクトグして、セナアッププログラムを起動します。セットアッププログラムこより StyleVision を目的の場所へインストールすることができます。 Altova XSLT Engines (1.0 と2.0) は、StyleVision 内部は搭載されており、内部的な全ての変換に使用されます。 StyleVision とおりに XSLT エンシをインストールする必要はおりません。。

以下のエンポーネントをインストールする必要があます:

● HTML プレビューそしてデザインビューを使用するために Internet Explorer 5.5 以上が必要になります。 Internet Explorer 6.0 以降では XML のサポートが強化されており、インストールが推奨されます。 Internet Explorer 9 もサポートされます。 (Excel テーブルとHTML ページエレテングなど、W ord ドキュメト (退場) 付けることのできるコンテングの W ord ドキュメトカらのエピーと貼り付け W ord 2007+ が必要になります

□ 関連項目

- 生成されナンアイル
- 一般的な使用方法

22 はじめに 用語解説

1.6 用語解説

このセグションでは、StyleVision GUI ならびにそのドキュメンテーションで使用される用語の解説を行います。用語は以下に示されるグループに分けられ、グループ内でアルファベット順に表示されます。

Altova 製品に関連する用語

Altova 製品に関連する用語を以下に示します:

Authentic 以下のAltova 製品にてXMLドキュメトのエディタービューを利用することができます: Altova XMLSpy; Altova View Style Vision; Altova Authentic Desktop; Altova Authentic Browser。Authentic View ならびに Altova 製品に関する詳細については、Altova Web サイトを参照ください。

SPS StyleVision Power Stylesheet を省略したもので、このドキュメンテーションでは StyleVision により作成された デザインド・キュメントのことを指し、sps とう拡張子により保存されます。詳細については SPS となっを参照してください。

グロー・ ジレノソ ファイルのセナ、フォルダーのセナ、そしてデータベースのセナーに対するエイリアスです。各リアイアスコは構成セナカ管ース まれており、各構成がリノースにマピングされます。StyleVision にてグロー・ ジレノノースが使用される場合、 StyleVision におけるアクティブな構成を変更することで、割り当てられたリノースを変更することができます。

一般的な XML の用語

このドキュメンテーションで使用される XML に関する用語を以下に示します:

- スキーマ (英数文字のを伴うスキーマは、一般的なスキーマを指します。 Style Vision では(大文字で表示される) XML スキーマと DTD がサポートされます。
- XML スキーマ XML スキーマこのドキュメンテーション ある(大文字で表示される) XML スキーマ とは、W3C の XML スキーマとは、W3C の XML スキーマとは、W3C の XML スキーマとは、W3C の XML スキーマとは、小文字で表示される) 全スキーマ のサブセナ となります
- URI とURL このドキュメンテーションでは、「ロケーター」として付け使用される識別子のことを表し、http スキーマを使用する識別子に対しても、更に適用範囲の広いURI が使用されます。

XSLT とXPath の用語

XSLT 1.0 とXPath 1.0 からXSLT 2.0 ならびにXPath 2.0 にかけて、幾つかの用語が変更されました。例えば XPath 1.0 でルナノードと呼ばれていたものが、XPath 2.0 では<u>ドキュメトノード</u>と名前が変更されています。このドキュメンテーションでは XSLT 2.0 ならびにXPath 2.0 に準拠した用語が使用されます。

- 絶対 XPath ユンテキストノードを含んで、るソノーのルートノードから始まる、文の条件式です。 Style Vision のダイアログにて、文条件式を入力する場合、絶対 XPath チェックボックスニチェックを入れることで絶対、文として条件式を入力することができます。 このチェックボックスのチェックが外されている場合、ユンテキストノードに対して相対的ない なが入力されます。
- コンテキストア コンテキストアイテムとよ、条件式が評価されるアイテム(ノードまけは文字列の値) に相対的なアイテムとはります。コンティテム キストノードとは、ノードのエンテキストアイテムのことを指します。コンテキストアイテムは条件式内部でも、例えば各ステッコンテキストノ プの位置やフィルター条件式の中で変更することができます。
- カレントノード カン・トノードとは現在処理されているノードのことです。サブ式を持たない条件式の場合、カレントノードはユンテキストノードと等価しないます。サブ式が存在する条件式の場合ではユンテキストノードが変わることもあります。current()

はじめに 用語解説 23

関数はXSLT 関数で、XPath 関数では無い点に注意してくたさい。

ドキュメント要整形式のXMLドキュメトでは、最も外側に位置する要素が、キュメト要素と呼ばれます。ドキュメト要素はド素 キュメトノードの子どは、整形式のXMLドキュメト内には、キュメトフ書が存在することがます。 GUI において、ドキュメト要素はレト要素と表示されます。

ドキュメントノードにおキュメント全体が含まれ、ドキュメントのソノー構造におけるルトノードといます。ルトノードは 一・ XPath 条件式ではソーにて表されます。StyleVision のスキーマソノーウィバウでは、「/ルト要素」として表示されます。

Style Vision-でのみ使用される用語

StyleVision のメカニズム、コンセプト、コンポーネントを表す用語を以下に示します:

- **青写真イメー** 青写真イメージは、<u>レイアナーコンテナー</u>の背景イメージとして使用されるもので、通常フォームをスキャンしたもの」ではまず。 青写真イメージをベースこSPS のデザインを行うことができ、フォームデザインの再度作成を行うことができます。.
- **動的アイテム** XML データノース内から得られたアイテムのことです。動的アイテムはテキスト、テーブル、リスト、そしてイメージや(URI か動的の場合)・ソイパーリンクとして作成することができます。
- グロー・ジル要 スキーマソレーウィド・ウ内にあるグロー・ジル要素リストに表示される要素のことです。XML スキーマンプロー・ジル要素と 素 して定義された全ての要素は、グロー・ジル要素リストにて表示されます。DTD において全ての要素はグロー・ジル要素と なるが、グロー・ジル要素のリストにて表示されます。 グロー・ジルテンプレート は、グロー・ジル要素に対してのみ定義されます。
- グロー・ジレテン グロー・ジレテンプレートは グロー・ゾル要素 1 ス対して定義することができます。 グロー・ジレテンプレートが一旦定義される プレーナ と ドキュメント 内に出現する要素がどコンあろうと、その要素に対してグロー・ジレテンプレートが使用されます。 グロー・ジレテンプレートの代わりに、ローカルテンプレートにてグロー・ジレ要素を処理することができます。
- レイアナーン レイアナンテナーとは、デザイン要素を絶対位置により配置することができるデザインブロックのことです。デザインがフォテナー ーノをベースとしている場合、レイアナーン・アナーとしてデザインを作成することができ、フォームの要素を絶対位置により配置することができます。それ以外の場合、デザインをフレー・マドで作成し、ドキュメト内にレイアナーン・テナーを配置することができます。
- ローカルテンプレートにより、<u>メインのテンプレート</u>内にある(グロー がはよび)非グロー がの) 要素がどのように処理されるレーナ かが定義されます。ローカルテンプレートは、メインのテンプレート内にある特定の要素出現に対して適用されます。ローカルテンプレートとは違い、グロー がテンプレート は、メインのテンプレートにおける要素の全出現に対して適用されます。
- *メインスキー*マ 割り当てられたスキーマソースの1つがメインスキーマとして使用されます。<u>作業用 XML ファイル</u>の ドキュメトノードがダインスキーマと関連付けられ、<u>メインのテンプレート</u>の開始点として使用することができます。
- メインのテンプ メインとなるエトリーポイトです。Style Vision において、このテンプレートは<u>ドキュメト要素</u>にマッチして、XSLT レーナ プロセッサーこと最初に評価されるものとなります。スキーマソレーウィドウでは、<u>ドキュメトノード</u>の子として表示されます。<u>メインのテンプレート</u>により、基本的なドキュメト構造が定義されるまか、入力ドキュメトがどのように処理されるかり定義されます。その他にも、<u>ローカルテンプレート</u>を含めることができ、<u>グローバルテンプレート</u>への参照を行うことができます。
- 出力 XSLT スタイルシートを伴う XML ドキュメントを処理することで生成された出力のことです。 Style Vision にお 生成 することのできる出力ファイルはwould be HTML 形式といます。 Style Vision にお 上成された XSLT スタイルシート として記述されます。
- **静的アイテム** XML データノースではなく、SPS から得られたアイテムのことです。静的アイテムコまデキスト、テーブル、リストのほか、(URI か静的な) イメージやハイパーレンク、ブックマークがあります)。
- SPS コンポーネトは (i) スキーマノード (例えば要素ノード) 、(ii) <u>自動計算</u> やデキスト文字列という。 ネント SPS コンポーネト、おけな (iii) SPS にて開始ならびに終了タブとして表示される <u>定義済みのフォーマト</u> のどれか しないます。

24 はじめに 用語解説

テンプレーナ ノードやノードのガループを処理するための処理セナとして考えることができます。

テンプレート XML ファイルは Style Vision (Enterprise ならびに Professional Edition) 内の SPS へ関連付 XML ファイル けられます。 SPS が Authentic View にて開かれた際に、新力に作成される SPS を伴った XML ドキュメトの開始データとして テンプレート XML ファイルが使用されます。 テンプレート XML ファイルは SPS がベースとして るスキー マンブして 妥当でなければないません。

ユーザー定義 スキーマソノーにてノードでもなければ、定義済みの要素やデザイン要素でもない要素で、ユーザーから指定された要素 要素 のことです。属性により要素の指定を行うことができます。

ユーザー定義 XPath 条件式にて指定されたシーケンスコ対して作成されるテンプレートです。 テンプレーナ

ユーザー定義 デザイン内の任意の場所にXML テキストブロックを挿入することができます。

XML テキスト ブロッ

テキスト フロッ ク

作業用 XML/XBRL ファイル

作業 XML/XBRL ファイルは Style Vision 内の SPS に関連付けられた XML データファイルで、Style Vision にて XML ドキュメトの出力やのプレビューを行うかが スML データが処理されることはおりません。2つ以上のグロー・ い要素が含まれるスキーマを SPS が、一スコーて る場合、どのグロー・ い要素が キュメント 要素 しなるのかが曖昧 しなってしまします。 作業用 XML/XBRL ファイルを割り当てることで、そのような曖昧性が解決されます(妥当な XML ドキュメントには ドキュメント 要素が つしか合まれない ため)。 XBRL の機能は Enterprise Edition でのみ利用できることに注意してください。

XML ドキュメ XML ドキュメント以下のためご使用されます: (i) 特定のXML ドキュメント表す、(ii) XML データノースを表すのこンナ 使用されます。

三 関連項目

- SPS 出む
- 一般的な使用方法

はじめに このドキュメントについて 25

1.7 このドキュメントについて

このドキュメンテーションは Style Vision に同封されているユーザーマニュアルで、 Style Vision に内蔵されているヘッレプシステムとしてご利用 になれるほか、 Altova Web サイトや、印刷用のPDF ファイルをダウンロードすることもできます。

ユーザーマニュアルは、以下のようなセクションにより構成されています:

- はかい SPS の説明や、Style Vision の主な機能やコンセプトを紹介します。
- ユーザーインターフェイスの説明: StyleVision GUI の概要を記します。
- チュートリアル: Style Vision は搭載されている機能の使用方法について説明します。
- 使用方法の概要: SPS の作成に使用されるスキーマノースや、デザイン処理の概要、Authentic View のデプロイヤプロジェクトといす。高度な機能の使用方法について記述します。
- <u>SPS ファイルエレテンツ</u>: 静的(スタイルシートベース) かつ動的(XML ドキュメントベース) なコンポーネントの作成方法と、SPS で編集する方法について説明します。
- <u>SPS ファイル構造</u>: SPS ファイルかどのような構造を持ち、どのようにモジュール化されているか説明し、StyleVision のテンプレート処理について記述します。
- SPS ファイルの高度な機能 計算結果の自動生成や、条件の設定、ユーザー定義された条件のグループ化や並べ替え、そして 出力ドキュメトにて目次や相互参照を作成するための方法について説明します。
- SPS ファイルプレゼ・テーション: SPS エレポーネトのフォーマナや、レイアナトについて説明します。
- SPS ファイル追加編集機能: SPS を更にパフルに使用するかが機能について説明します。その他のAltova 製品の機能を利用するグルー・ジャノースや、更なる検証、変数やプメーターなどについて記述されます。
- レファレンス:StyleVision で使用されている全てのシンボルヤコマンドに関する記述が収められます。
- 付録 Altova XSLT エンジンの情報、Style Vision の技術情報、そしてライセンスに関する情報が収められます。

使用方法

まずは Style Vision の概要を知るから、はじめに、ユーザーインターフェイスと使用方法の概要 セグション 引するセグションへ目を通すことをお勧めします。その後、チュートリアルをこなすことで、SPS の作成方法について実際に確認することができます。 SPS ファイルのセクション (SPS ファイルエレデンツ、 SPS ファイル構造、 SPS ファイルの高度な機能、 SPS ファイルプレゼンテーション、 SPS ファイル追加機能)では、 Style Vision にて用意されている様々な機能の使用方法について確認することができます。 レファレンスセグション コは全てのソール ドーアイコン、デザインシンボル、そしてメニューコマンドが、ツール ドーとメニューごと 記述されており、コマンドの確認などを行うために使用することができます。

Windows 7、Windows 8 およびWindows 10 でのファイルパス

このドキュメトのファイル・パスはすべてのオペレーティングシステムで同じではありません。以下の通信に注意してください

• *(マイ)ドキュメント フォルダー: デフ*ォルトでは以下の場所に存在します。Example ファイルはこのフォルダのサブフォルダに存在します。

Windows 7/8/10	C:\Users\ <username>\Documents</username>

アプケーション フォルダー: アプケーション フォルダーコは Altova アプケーションが存在します。 デフォルトでの、アプケーションフォルダの いは以下の通りです。

Windows 7/8/10	C:\Program Files\Altova\
64-bit OS 上での82 bit バージョン	C:\Program Files (x86)\Altova\

メモ StyleVision は プラケンオーム更新済みのWindows Server 2008 R2 SP1 ませは以降 でもサポートされています。

26 はじめに このドキュメントについて

サポート オプション

StyleVision、は関する質問や問題がある場合、以下のサポートオプションがご利用になれます:

1. <u>ヘリプ</u>ファイル(このドキュメンテーション)の確認。ヘリプファイルコはキーワードによる検索の他に、全文検索機能が搭載されています。

- 2. Altova Web サイト の良くある質問 とディスカッションフォーラム を確認してくたさい。
- 3. Altovaサポートセンターに連絡してくたさい。

頻繁に使用される略語

このドキュメンテーションでは、以下の省略形が頻繁に使用されます。

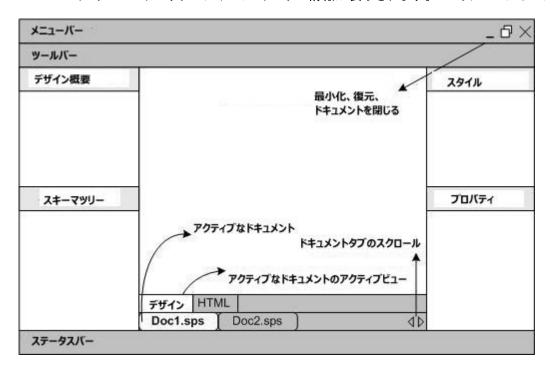
- **SPS**: StyleVision Power Stylesheet
- CSS: カスケーディングスタイルシート
- FAQ: 頻繁に寄せられる質問

ユーザーインターフェイス 27

2 ユーザーインターフェイス

StyleVision GUI は、以下の いから構成されます(下に表示されているユーザーインターフェイス/コはすべてのサイドバーは表示されていません):

- メニューバー: メニューをクリックすることで、そのメニュー内にあるアイテムが表示されます。メニューアイテムについては ユーザー・リファレンス メニュー・ドーコ は最小化、最大化、ドキュメトを閉じるためのドダンが表示されます。
- ツール・領域:様々なツール・ならびごとれらツール・コマイショートカナについてはユーザーリファレスにて記述されます。
- タブ化されたメインウィンドウ、その時開かれている SPS ドキュメントが表示されます。このウィンドウでは <u>SPS デザインの編集</u> そして XSLT スタイルシート と出力のプレビューを行うことができます。
- ステータスバー: アプリケーションのステータス情報が表示されます。64ビットバージョンの



メインウィードウとデザインサイドバーの詳細にひては、このセクションのサブセクションに記述されます。

メモ メニューバーならびにツールバーのタイトル部分をドラッグすることで、目的の場所へ移動することができます。

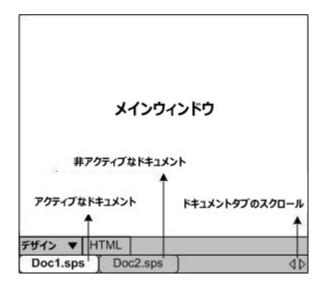
- 関連項目

- メニューコマンド セグション
- レファレンスセグション ツール ←

28 ユーザーインターフェイス メインウィンドウ

2.1 メインウィンドウ

メインウィンドウコはSPS デザイン、XSLT スタイルシート、そして出力プレビューが表示されます(以下のイメージを参照)。



メインウインドウ内の SPS ドキュメント

- StyleVision では複数の SPS ドキュメントを開くことができますが、同時に選択できるアクティブなドキュメントは1つだけです。開かれているドキュメントの名前は、メインウィンドウ下部にあるタブに表示され、アクティブになっているドキュメントのタブがハイライトされたかたちで表示されます。
- タブをクリックして、開かれているドキュメントをアクティブにすることができます。その他にも、ウィンドウメニューにあるオプションを使用することができます。
- 同時に開かれているドキュメントの数が多く、全てのドキュメントタブが表示されない場合、(ドキュメントタブバーの右側にある)スクロールボタンを使用することで、タブのスクロールを行うことができます。
- アクティブなドキュメントを閉じるには、アプリケーションウィンドウの右上にあるメニューバーの、ドキュメントを閉じるボタンをクリックします(または「ファイル」閉じる」を選択します)。

ドキュメントビュー

以下にあるビューにで、キュメントは表示されます。ビューは同時に1つしか選択することができません。

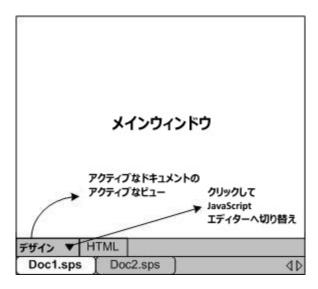
- <u>デザインビュー</u>: SPS のデザインと、SPS で使用する Java Script 関数の編集を行うことができます。ドロップダウンメニューの矢 印をクトックして、デザイン、ませは Java Script を選択することで、ドキュメントのデザインと Java Script エディターを切り替えることができます。
- <u>出力ビュー</u>(HTML 出力): それぞれのフォーマナトにて出力されるファイルのプレビューと、出力を生成するために使用される XSLT スタイルシートのプレビューカ張示されます。ドロップダウンメニューの矢印をクリックして、目的の選択を行うことで、出力のプレビューと XSLT スタイルシートの表示を切り替えることができます。

上にあるビューは、メインウインドウ下部に表示されるダブをクトックすることで切り替えることができます。選択されてビューのダブがハライトされます。

ユーザーインターフェイス メインウィンドウ 29

2.1.1 デザインビュー

SPS のデザインはデザインビューにて行います。デザインビューでは(i)サイドバー、キーボードからの入力、メニューやソールドーから選択することができるコンテンツ作成ならびに編集機能を使用、ませは(ii)サイドバーとメニューから選択することができる様々なフォーマト機能を使用することで出力ドキュメントのデザインを作成します。デザインビューは搭載されたこれらの機能について以下で説明します。



デザインビューは、JavaScript エディターへ切り替えることもできます。JavaScript エディターでは、SPS にて使用することができる
JavaScript 関数 を作成することができます。 JavaScript エディター へ切り替えを行うには、デザインタブ内にあるドロップダウンがなった
リックしてドロップダウンメニューから JavaScript を選択します。デザインビューへ切り替え直すには、JavaScript にあるドロップダウンが
ンをクトクして、ドロップダウンメニューからデザインを選択します。

デザイビューでは、メインテンプレート、グローブルテンプレート、ページ・イアナトンプレート、そしてデザインプラグメトといす。複数のテンプレートをSPS 内に収めることができます。 <u>ツールドーアイエン</u>として表示されている <u>テンプレート デザイン フィルター</u>を使用することで、どのテンプレートをデザインビューにて表示するか指定することができます。表示フィルターを使用することで、SPS の最適化と、SPS 間の表示を切り替えることができます。

マーケアップタグの表示

マーケア・プアイコンを使用することで、デザインビューにあるマーケア・プタグの表示を切り替えることができます(以下を参照)。



上にあるアイコンのうちどれかを選択することができます。左から、(i) 小さなデザインマーケアップ(名前なしのタグ)を表示、(ii) 大きなデザインマーケアップ(名前付きタグ)の表示を表します。小さなマーケアップが選択された場合、ノードへマウスを移動させた時に、ノードへの なか表示されます。

2.1.2 出力ビュー

それぞれの出力ビュータブ(以下のイメージを参照)は、以下を表示します: 出力ビュータブにより、(i) SPS から生成された XSLT-HTML スタイルシートと (ii) XSLT スタイルシートを 作業用 XML ファイル に対して使用することで得られた HTML 出力のプレビューが表示されます。

30 ユーザーインターフェイス メインウィンドウ

HTML 出力ビュータブにあるドロップダウンボタンをクトックし、XSLT オプションませば出力プレビューオプションを選択することで、HTML 出力ビュータブの表示内容を XSLT+HTML スタイルシートと HTML 出力のプレビューで切り替えることができます。



XSLT ビュー

ビュー・コは、現在アクティブな SPS によりXSLT-for-HTML により生成された XSLT スタイルシートが表示されます。XSLT ビューが選択されるとびにスタイルシートは新たって生成されます。

出力ビュータブに表示されるスタイルシートには、行番号や展開/縮退アイエンも表示されます。左側の余白にある+ならびに-アイエンをクリックすることで、要素の展開/縮退を行うことができます。XSLT ビューに表示されたスタイルシートを編集することはできませんが、テキストの検索を行い(「編集 | 検索」を選択)、クトップボードへのエピーを行うことはできます(「編集 | コピー」を使用)。

メモ SPS に対して使用して得られた出力が表示されます。「ファイル」生成されたファイルの保存」コマンドを使用することで、個別に生成、保存することができます。

HTML プレビュー

HTML では、作業用 XML ファイルがHTML プレビュータブ かり トックされるさいに、新たった成されます。作成されるのは保存バージョンの作業用 XML ファイルで、— Authentic Viewl て編集される一時的なファイルではは、ことに注意してくたさい。

作業用 XML ファイル か割り当てられていない状態で、HTML 表示 タブニてHTML プレビューか表示された場合作業用 XML ファイルを割り当てるように促されます。DB をベース した SPS の場合、DB がロードされ、そこから一時的な(編集不可能な) XML ファイルが主成され、作業用 XML ファイルとして使用されるため、作業用 XML ファイルを割り当てる必要はありません。

メモ SPS から生成された出力ファイルは、個別に生成され、「「ファイル」生成されたファイルの保存」コマンドを使用することで、ファイルの保存を行うことができます。

□ 関連項目

- StyleVision のセットアップ
- 生成されたファイルの保存

2.2 サイドバー

サイドバー(おけはサイドバーウンドウ)はSPS のデザインを行うかのGUI コンポーネントで、アクティブなビューは関する情報を提供します。以下に示される各サイドバーことでは、このセクションのサブセクションこで解説されます。

- デザイン概要
- スキーマソノー
- デザインソノー
- スタイルポネリ
- スタイル
- <u>プロ ディ</u>
- メッセージ
- 検索と置換

ビューのレイアウト

ビューのレイアナとは、そのビューでどのサイド、ト・が利用可能なのかと、これらサイド、ドーがGUI にてどのように配置されているかのことを指します。レイアナトは個々のビューカテゴリーは大してカスタマイズすることができ、カスタマイズは(i)ビューはおける個々のサイド、ドーの表示(表示メニュー、おけまサイド、ドーのタイトルドーを右クトックして、「隠す」を選択する)と(ii)必要に応じてGUI にてサイド、ドーを配置するという2つから構成されます。この方法により定義されオレイアナトは、そのビューカテゴリーは対して保持されます。例えば、デザインビューにてスタイルサイド、ドーを除く全てのサイド、ドーカ表示されて、サナ場合、次回デザインビューレイアナトが変更されるまで、このレイアナカデザインビューに対して保持されます。出力プレビュー(HTML、RTF、PDF、Word ビューカテゴリーは以下のとおりです。(i)ドキュメントが開かれていない、(ii)デザインビュー、(iii)出力ビュー・

サイドバーウィイナウのドッキングとフロート

サイド、トウィイ・ウは Style Vision GUI 「コオレてドックするか、画面上「フロート状態」こすることができます。ウィイ・ウをドックするコオタイトルドーをドラッグし、ドラッグする際に表示される矢印「エドラックしてください。内側に表示される矢印ペドロップすることで、矢印が表示されたウィイ・ウの内側「コオレイドラッグされたウィイ・ウを配置することができます。外側にある矢印ペウィイ・ウをドロップすることで、矢印の表示されているウィイ・ウの縁「ボッキングが行われます。ウィイ・ウをフロート状態にする「コよ(i)タイトルドーをダブルクトックするか、(ii)タイトルドーをドラックして、画面上に表示される矢印以外の場所でドロップしてください。

その他にも、以下のメカニズムを使用することができます。ドックされたウイドウをフロート状態にするコよ、ドックされたウイドウの右上にあるメニューボタンをクトックして、フロートを選択します。このメニューへは、ドックされたウイドウのタイトル、ドーを右クトックすることでもアクセスすることができます。



フロート状態のナイドウをドッキングするコま そのナイドウのタイトル デーを右クトックして、表示されるメニューからドッキングを選択します。最後はドッキングが行われた場所にナイイドウルドッキングされます。

デザインサイドバーウィンドウを自動的に隠す

ドックされたウィイドウは自動的に隠すことできます。サイドバーウィイドウが自動的に隠れるように設定されている場合、GUI の端にタブというかたちで最小化されます。マウスカーノルをタブ上に配置することで、メイノウィイドウェウィイドウが表示されます。以下にあるスクリーシショナでは、スタイルタブの上にマウスカーノルを配置することで、スタイルサイドバーがインウィイドウ上に表示されます。



マウスカーノルを表示されたウンドウの外側へ まけおりとは別の場所に移動させると、そのウンドウがりとして GUI の端に再度収まります。

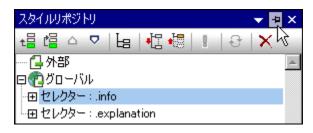
自動的に隠す機能は、頻繁には使用しないサイドバーを GUI に表示させず、必要な時にはオナアクセスするようは場合に便利な機能となります。この機能により、デザインサイドバーウィンドウへ簡単にアクセスすることができるとともに、メインウィンドウで更なるスペースを使えるようになります。

ウイドウを自動的に関すためコよ、ドッキングされたウイドウの右上にある自動的に関すがダン(ピンのかたちをしたアイコン)をケッケします。 それ以外にも、ウイドウのダイトル・そ右クックするか、ドッキングされたウイドウのダイトル・イニあるメニューポックをクックすることで表示されるメニューから、自動的に関すオプションを選択して、この機能を有効にすることができます。



ウンドウが自動的「穏されるよう」なります。

特定のナイドウニ対して自動的に隠す機能を無効にするコよ、マウスカーノルをそのタブへ移動してウイドウか表示された状態で、自動的に隠すがなンをクナックします。それ以外にも、ウイドウのタイトルドーを右クナックするか、ドッキングされたウイドウのタイトルドースあるメニューがシェクナックすることで表示されるメニューから、自動的に隠すオプションの選択を無効にすることでも、この機能を無効にすることができます。



メモ サイドバーウィンドウの自動的に隠す機能が無効になっている場合、そのウィンドウのピンアイコンが下に 向かって表示され、有効になっている場合は、ピンの向きが左方向の状態で表示されます。

サイドバーウンドウを隠す(閉じる)

サイドバーウィイウが隠されると、GUI 上から直接確認することはできません。サイドバーウィイウは最大化された状態(ドッキングまとはフロート状態)が、最小化された状態(自動的に非表示にする機能によりGUI の端に表示されるダブとして)でのみ表示されます。

ウイドウを隠すコよドッキングまけまプロート状態にあるウイドウの右上に表示されている閉じるボタンをクックします。それ以外にも、ウィンドウのタイトルドーを右クックするか、ドッキングされたウイドウのタイトルドーであるメニューボタンをクックすることで表示されるメニューから、隠すを選択することができます。

隠されている(閉じられている)状態のウイ・ドウを再度表示するコよ。 表示 メニューカら目的のデザインサイド・バーの名前を選択します。 デザインサイド・バーウィードウが、隠された時の位置と状態(ドッキングまだはエルート状態)で再度表示されます。

- □ 関連項目
- デザインビュー
- 表示メニュー

2.2.1 デザイン概要

デザイン概要サイドバーを使用することで、アクティブな SPS ロネルてスキーマノース グロー・ジレ ウメーター、SPS モジュール そして CSS ファイルを追加することができます。これらコンポーネトをまとめて表示することで、コンポーネトの集中管理を行うことができます。



スキーマノースの追加

空の SPSへSPS を追加することができます。ソースのヘッダー以下にある新しいソースの追加コマイドをクトックすることで、スキーマノースを追加することができます。この機能により表示されるポップアップメニューにより、XML スキーマ、DTD、XML ファイルにより生成されたスキーマ、お台はユーザー定義スキーマ



作業用 XML ファイル

スキーマが追加されると、そのスキーマがソースアイテム以下に表示されます。 XML アイテム内の作業用 XML ファイルのためコよ

モジュール、CSS ファイル、パラメータ、XSLT ファイルの追加

モジュール CSS ファイル、パラメーター、そして XSLT ファイルセグションの下部にある、新たに追加するコマンドをクリックすることで、対応するセグション:新たなアイテムを追加することができます。

デザイン概要の機能

以下の機能は、デザイン概要サイドバーにある各セクション(ソースやパラメーターなど)全てに共通して使用することができます:

- 各セクション名の左側にある三角形の矢印をクリックすることで、そのセクションを展開、または縮退することができます。
- ソース、モジュール、CSS ファイルセクション内にあるファイルセクションにはファイル名だけが表示されます。ファイル名へマウスを移動すると、ファイルのフルパスがポップアップにて表示されます。
- 灰色で表示されているアイテムはインポートされたモジュールを表しており、GUI にてアクティブしておっている SPS ファイル内には無いモジュールとなります。
- 各セクションの下部には新しい 〈アイテム〉 を追加コマンドが含まれており、そのセクションに対して新たなアイテムを追加することができます。例えば、新しいパラメーターを追加コマンドをクリックすることで、SPS とデザイン概要のパラメーターリストへ新たなパラメーターが追加されます。

セクション内にある各アイテムにはコンテキストメニューが含まれており、そのアイテムを右クリックするか、コンテキストメニューアイコン (アイテムの右側にある下向きのアイコン)をクリックすることで表示されます。

- コンテキストメニューに含まれている**削除**アイコンを選択することで、選択されたアイテムを削除することができます。このコマンドは、コンテキストメニューコマンドからも利用することができます。
- コンテキストメニューコマンドにある XMLSpy でファイルを編集を選択することで、Altova XMLSpy にてファイル を編集することができます。
- コンテキストメニューコマンドにある上へならびに下へコマンドは、モジュールセクション内に<u>複数のモジュール</u>が表示されており、そのうち1つのモジュールが選択されている際に有効になります。選択したコマンドにより、選択されているモジュールの位置を変更することができます。

ソース

ソースセグションのJストには、スキーマと、SPS をベースしている作業用 XML ファイル か表示されます。(右クナック、ませまレテキストメニューアイコン をクナックすることで表示される)コンテキストメニューヘアクセスし、適切な割り当 て... オプションを選択することで、これらのファイル選択を変更することができます。

モジュール

モジュールセグションイコは、アクティブな SPS にい使用されている <u>SPS モジュール</u>が表示されます。新しいモジュールを追加コマンドをクトグし、目的のSPS ファイルを選択することで、新たなモジュールをリストへ追加することができます。

<u>モンュールが表示される順序</u> には意味があるため、複数のモンュールが表示されている場合、(モンュールのエンテキストメニューからアクセスすることができる)上へ/下へコマンドによりモンュールの順序を変更することができます。コンテキストメニューはよ Style Vision にて選択されたコマンドを開くためのコマンドも用意されています。

メモ デザイン概要サイドバーではモジュールの概要を確認することができ、ファイルレベルでモジュールの管理を行うことができます。様々な <u>モジュールオブジェケト</u>(モジュール内部のオブジェケト)は デザインソノー サイドバーに表示されます。

CSS ファイル

CSS ファイルセグ・コイコは、アクティブな SPS により使用されている CSS ファイルも表示されます。新しい CSS ファイルを追加コマイ・をクトグして、目的の CSS ファイルを選択することで、新たな CSS ファイルがリストへ追加されます。 CSS ファイルな張示される順序 には意味があるさめ、複数のファイルが表示されている場合、(コンテキストメニューからアクセスすることができる)上へ/下へコマイ・が選択可能にないます。目的のコマイドをクトグウすることで、選択されたこCSS ファイルを移動することができます。コンテキストメニューコは、選択されたモンシュールを XMLSpy にて開くためのコマイ・も用意されています。

メモ デザイン概要サイドバーでは CSS ファイルの概要を確認することができ、ファイルレベルで CSS ファイル の管理を行うことができます。 CSS ファイル内にある様々な <u>CSS ルール</u>は、<u>スタイルポネリサイドバーに表示されます。</u>

パラメータ

/ ウメーターセグションコは SPS 内にあるグロー ジレ・ウメーターが表示されます。セグションの下部にある新たなパラメーターを追加コマイドを使用することで、新たなパラメーターを追加することができます。 / ウメーター名や値をダブルグ トックすることで、それぞれ名前や値を編集することができます。 / ウメーターを削除するコは、目的の ・ウメーターを選択して、コンテキストメニューから削除コマイドをグ トックする必要があります。

XSLT ファイル

XSLT ファイルセクションコま SPS 内にインポートされた XSLT ファイルが表示されます。これら XSLT ファイルに含まれている XSLT テンプレートは、グロー・ドレテンプレートとしてスタイルシートにて利用可能にないます。この機能の詳細については: XSLT テンプレートを参照くたさい。

□ 関連項目

- パラメータ
- <u>スキーマソレー</u>
- モジュールSPS
- デザインプラグメント
- スケノプトの使用

2.2.2 スキーマツリー

スキーマッリーサイドバーでは、以下のような操作を行うことができます(アのスクリーンショナ):

- スキーマに対して複数のルート要素(ドキュメント要素)を選択します。
- スキーマツリーからノード(要素、属性、グローバルタイプ)をドラッグして、デザインヘドロップします。これらノードは、出力に含められる XML コンテンツを表します。
- スキーマノース内にあるグローバル要素ならびに型が表示されます。グローバル要素ならびにグローバル型をグローバルテンプレートとして作成することができます。
- SPS にで使用されている全ての名前空間を表示します。
- デザインフラグメントの挿入と編集を行います。
- SPS に対してユーザー定義された XPath 関数 の挿入と編集を行います。



ルート要素

\$XML 以下にある各スキーマン対して、選択された<u>ルト要素</u>(まだは<u>ドキュメト要素</u>))が表示されます。このノストでは選択したスキーマン関連付けられている全てのルト要素が表示されます。各ルト要素を展開すると、モデルソノーが表示されます。これらルト要素内にあるノードから、メインテンプレートのエンテンンが作成されます。メインテンプレートのエントノードで、ドキュメントノードは、つでも変更することができます(その方法については以下を参照ください)。

スキーマス対してルート要素を選択するコよ、以下の操作を行います。ルート要素アイテムの右側にある選択 いけるとをクックします。この操作により、ルート要素の選択ダイアログが表示され、スキーマ内に含まれているグロー・い要素がシルート要素を選択することで利用可能しなるオプションの説明については、SPS 構造 スキーマノースを参照ください。



更に、追加した全てのスキーマ内にある全てのグロー・いと要素 スキーマ か全てのグロー・いと要素以下に表示されます。各グロー・いと要素に対しては、グロー・いデンプレートを作成することができます。

グローバル要素とグローバル型

グロー・シル要素ならびにグロー・シル型は、その他のテンプレートでも再利用することができるグロー・シルテンプレートを作成するかがます。更に、グロー・シル型はテンプレート内で直接使用することができます。

デザインフラグメント

ドキュメント内にある全ての<u>デザインフラグメント</u> はデザインフラグメントアイテム以下に表示され、デザインフラグメントアイテムを展開することで確認することができます。以下にあるデザインフラグメント機能を利用することができます:

- デザインフラグメントアイテムの追加アイコン をクリックすることで、デザインフラグメントを作成することができます。
- スキーマツリーにあるデザインフラグメントの名前をダブルクリックすることで、デザインフラグメントの名前 を編集することができます。
- 各デザインフラグメントの横にあるチェックボックスを有効、または無効にすることで、対応するデザインフラグメントを有効、または無効にすることができます。
- デザインフラグメントをスキーマツリーからデザインをドラッグすることができます。

デザインフラグメントを扱うための情報については、デザインフラグメントのセクションを参照ください。

ユーザー定義 XPath 関数

XPath 関数アイテムの追加アイコン をクックすることで、ユーザー定義 XPath 関数を追加することができます。XPath 関数を作成すると、その関数がスキーマソノー内に表示されます。XPath 関数をダブルクックすることで、その関数がダイアログにて開かれ、編集を行うことができるようしています。以下のXPath 関数を利用することができます:

- XPath 関数アイテムの隣にあるチェックボックスにチェックを入れる、またはチェックを外すことで、XPath 関数を有効、または無効にすることができます。
- XPath 関数を右クリックすることでコンテキストメニューが表示され、XPath 関数の名前変更や削除を行うことができます。

XPath 関数の詳細な操作方法については、ユーザー定義 XPath 関数 のセクションを参照くだとい。

名前空間

SPS 内で使用されている名前空間のノストが、プレフィックスととも「名前空間以下に表示されます。リスト内にある名前空間は(i)参照されたスキーマや、スキーマ内部で定義されている名前空間、そして(ii)デフォルトで新たって作成された。SPS に追加された名前空間となっています。このようなリストは、XPath 条件式を記述する際に便利な機能となります。更に、xpath-default-ns エイ・リーの値フィールドをダブルクアクして、名前空間の入力を行うことで、SPS 全体に対する XPath デフォルト名前空間を定義することができます。

メモ SPS や、SPS から生成された XSLT スタイルシートに対して名前空間を追加する場合、SPS がベースとしている XML スキーマのトップレベルスキーマ要素へ、名前空間を追加する必要があります。

ツールドーとスキーマソノーアイコン

以下のソール・ディコンは、一般的なスキーマソト・サイドバーコマ・ドレス対するショートカットとなります。

スキーマソノーで使用されるシンボル

スキーマソノーコおおシンボルを以下に示します。



子要素を含む要素。要素をダブルクトックし、左側にある + /- シンボルをクトックすることで、要素の 展開/縮退を行うことができます。



グロー・ ういの型は型を単純型まけば複合型となります。複合型はシアン色のアイコンにより表示され、単純型はブラナン色のアイコンにより表示されます。

| 関連項目

SPS 構造の作成 | スキーマノース

2.2.3 デザインツリー

デザインツリーサイドバーコよ SPS デザインの概要が表示されます。



デザインソノーのルートはよSPS の名前が表示され、マウスポインターをアイコン上に配置すると、ファイルの位置がポップアップで表示されます。デザインソノーにあるアイテムよ 以下のカテゴリーこ分類されます:

- <u>スクノナ</u>:StyleVision のJavaScript エディターによりSPS に対して定義された全てのJavaScript 関数が表示されます。
- メインのデンプレート: メインテンプレートの詳細な構造が表示されます。
- グローゾルテンプレート:現在のSPS 内にあるグローゾルテンプレートや、インクルードされた全てのSPS モジュールにあるグローバルテンプレートが表示されます。
- <u>デザインフラグメト</u>: デザイン内にある全てのデザインフラグメトか表示され、デザインフラグメトの作成、編集、名前変更、そして削除を行うことができます。
- XSLT テンプレート: インポートされた XSLT ファイル内にある XSLT テンプレートが表示されます。
- ユーザー定義 XPath 関数: 独自のXPath 関数を作成、編集、名前の変更、削除することができます。

ツール・アイコン

以下のソール・アイエンは、スキーマソトサイドバーはあー般的なコマイのショートカナとおります。



デザインス対してデザインフラグメート、メインテンプレート、おけましてアナトアイテムを追加します。アイコンの左側をケリックすることで、デザインフラグメートが追加されます。ドロップダウンを表す矢印をクリックすることで、デザインフラグメートや、その他のレイアウトアイテムを追加するかがのフトが表示されます。



選択されたアイテムを削除します。グロー・ジレテンプレートまたはレイアナトサブソノー内のアイテムが選択されている時に、アイコンが選択可能となります。



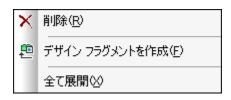
同期ソノーを切り替えます。有効になっている場合(アイエンの境界線が表示されている場合)、ツノー内のノードを選択することで、(i) デザイン内の対応するノードが選択され、(ii) デザインソノーウィドウ内にある同期ツノーアイエンが有効になっている場合、デザインソノー内の対応するノードが選択されます。アイエンが無効になっている場合、デザイン内の対応するノードが選択されます。アイエンが無効になっている場合、デザイン内の対応するノードが選択されることはおりません。



同期オプションが有効になっている場合、スキーマソノーにあるその他のアイテムが自動的に縮退され、デザイノにてアイテムが選択されます。このオプションオソーを同期オプションが有効によっている場合にゴジガ選択可能にないます。

デザインソノーの表示を変更する

デザインソノー内にあるアイテムを右クトックすることで表示されるコンテキストメニューから、デザインソノーの表示を変更することができます(以下のアクノーンショナを参照)。



削除コマイドにより、選択されたデザインソーアイテムをデザインソーか削除します。デザインフラグメントを作成により、デザイン内にデザインアラグメートを作成し、そのアイテムをデザインソートに追加します。全て展開コマイドにより、デザインソー内にある全てのアイテムを展開することができます。

スクリプトとメインのテンプレート

スクノプトロよ、インポートされたモジュール内にあるものを含め、デザイン内にある全てのスクノプトが表示されます。メインテンプレートロはメインテンプレートのソノーが表示されます。アイテムを右クトックして、削除を選択することで、ツノー内にあるアイテムを削除することができます。

グローバルテンプレート

グロー・シレテンプレート アイテムコよ 現在のSPS ならびに追加された全てのSPS モジュール内にある全てのグロー・シレテンプレートが表示されます。 現在のSPS にて定義されているグロー・シレテンプレートは黒色で表示され、 追加されたモジュールにで定義されたグロー・シレテンプレートはガレーで表示されます。 各グロー・シレテンプレートロよ 左側にチェックボックスが含まれており、 有効/無効を選択することができます。 グロー・シレテンプレートが無効化された場合、 デザインから削除されます。



現在のSPS 内にあるグロー・シレテンプレートをクトックし、ツール・コーある削除ポタンをクトックするか、コンテキストメニューから削除コマンドを選択することで、そのグロー・シレテンプレートを削除することができます。 デザインとソレーからコンポーネントが削除されます。

デザインフラグメント

デザインアグメト アイテムこより、現在のSPS ならがら追加された SPS モジュール内にあるデザインフラグメトが表示されます。現在の SPS 内にあるデザインフラグメトは黒色で表示され、追加モジュール内で定義されたデザインフラグメトはグレーで表示されます。各デザインフラグメトには、左側にチェックボックスが表示されており、対応するデザインフラグメトを有効/無効にすることができます。現在のSPS 内にあるデザインフラグメトを選択し、削除コマトをクリックするか、コンテキストメニューから削除コマトを選択することで、そのデザインフラグメトを削除することができます。コンポーネ・トがデザインとソーから削除されます。



デザインフラグメトアイテムの右側にある追加アイコン・をクトケクすることで、デザインフラグメトを追加することができます。各デザインフラグメトは展開、縮退可能なノードを備えたソーとしてデザインされます。デザインフラグメトツーを選択後に、ツール トの削除 ポタンをクリックするか、コンテキストメニューにある削除コマンドを選択することで、(現在のSPS にて定義された)任意のデザインフラグメトツーを削除することができます。デザインとソーからコンポーネントが削除されます。

XSLT テンプレート

デザインツリーサイドバーにある XSLT テンプレート以下には、インポートされた XSLT ファイルが表示されます(以下のスクリーンショットを参照)。



インポーされたXSLT テンプレートには種類がます: (i) match テンプレート、そして(ii) named テンプレート。デザインソレーでは、これら2種類のテンプレートが、(i) match テンプレートにある select 属性の値、(ii) named テンプレートにある name 属性の値ともにそれ それ表示されます。 XSLT テンプレートに関する詳細には、 XSLT テンプレートを参照ください。

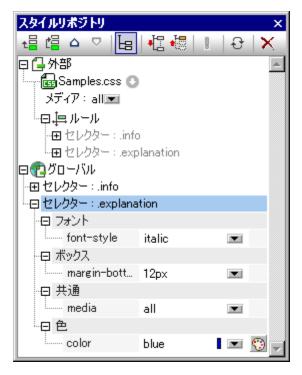
■ 関連項目

• デザインフラグメント

• スクノプトの使用

2.2.4 スタイルレポジトリ

スタイルリポジトリサイドバーでは、外部 CSS スタイルシートの割り当てや、SPS に対してグロー・ Vは CSS スタイルを定義することができます。 外部 CSS スタイルシートにあるスタイルレールとグロー・ VILに定義された CSS スタイルは、 HTML 出力 ドキュメントに対して適用されます。



スタイルノポシトリサイドバーコ**はブローバルと外部とい**名前で、2つのソノーが表示されます。外部ツノーコはSPS と関連付けられている外部 CSS スタイルシートのノストが含まれます。グローブルソノーコはSPS と関連付けられた全てのグローブレスタイルらきまれます。

スタイルパネリの表示構造は以下のようになっています:

```
外部
```

```
- CSS-1.css(マウスオーバーにより表示されるポップアップにて位置を確認できます)
- メディア(スタイルリポジトリウィンドウにて定義することができます)
- ルール(編集不可 - CSS ファイルにて編集する必要があります)
- セレクタ―-1
- プロパティー1
- ...
- プロパティーN
- ...
- セレクタ―-N
+ ...
+ CSS-N.css
グローバル
- セレクタ―-1
+ セレクタ―-1
- セレクタ―-1
- セレクタ―-1
```

+ セレクター-N

スタイルルールの優先順位

グロー シレスタイルレールと外部 CSS スタイルシートに記述されたスタイルレールの両方に、同じドキュメトコンポーネントを表すセレクターが 指定されている場合、グロー シレスタイルレールが、外部スタイルシートに優先して適用されます。複数のグロー シレスタイルレールが同一のドキュメトコンポーネントを指定している場合、最後に記述されたリールが適用されます。同様に、複数の外部スタイルシート内にあるスタイルシートが同一のドキュメトコンポーネントを指定している場合、最後のスタイルシート内にある最後のレールが適用されます。

スタイルルポントリにてスタイルを管理する

スタイルノポントリサイドバーでは、ツールバー上にあるアイコンを使用する、ませっぱフテキストメニュー内にあるアイテムを使用することで、以下の操作を行うことができます:

追加: 追加アイコン をクトックすると、選択されているソノー(外部まだはグロー がいにより、新たな外部スタイルシートエトリーを外部ツノーへ追加し、新たなグロー がスタイルエトリーをグロー がルソノーへ追加することができます。新規エトリーが、ツノー内にて既に存在するノストへ追加されます。追加コマイドはエンテキストメニューからも選択することができます。外部スタイルシートならびにグロー がスタイルを使用するための詳細な情報については、 CSS スタイルと作業 を参照くたさい。 デザイン概要サイドバーからも外部 CSS スタイルシートの追加や削除を行うことができます。

#A: 挿入アイコン により、(外部ツノー内にて)選択された外部スタイルシートの上部へ新たな外部スタイルシートエトリーを挿入し、(グロー・バルツノー内にて)選択されたグロー・バレスタイルエトリーの上部へ新たなグロー・バレスタイルエトリーを挿入することができます。 コマンドはエンテキストメニューからも利用することができます。 外部スタイルシート やグロー・バレスタイルを使用する詳細な情報については、次を参照してください CSS スタイルと作業.

上へ下へ: 上へアイコン ならびに下へアイコン により、選択された外部スタイルシートやグロー いスタイルを、ツー内部にてそれぞれ上や下へ移動することができます。 これらのコマイドを使用することにより、外部スタイルシートやグロー いスタイルの優先順位を操作することができます。 上へならびに下へコマイドはコンテキストメニューからも利用することができます。 スタイルの優先順位を変更する詳細方法については、 CSS スタイルと作業.

セレクターのスタイルを表示: 用意されている3つのビューの設定を使用することで、外部スタイルシートやグロー、いいに義されたスタイルシートの表示方法を指定することができます。空でないリスト と、全て展開 く 全て縮退 、コマドがソールドーとエンテキストメニューコマドにて利用することができ、空では、リストオブンコンを有効にすることで、値とともに定義されているスタイルプロッティイヨが表示されるようしています。このオブンコンが選択されていない場合、全てのスタイルプロッティが表示されます。全て展開ならば、全て縮退設定を、空では、リスト設定と組み合わせることで、選択されたセレクターのスタイル定義を展開、縮退することができます。これらのコマンドはコンテキストメニューからも利用することができます。

重要フラグの切り替え: 重要フラグの切り替えアイコン ● をクトックすることで、選択された CSS ルールス対して!important とら CSS 値をセルします。

すべてを再ローボ: すべてを再ローボ アイコン む により、全ての外部 CSS スタイルシートを再ロードします。

リセット: リセットアイコン と により、選択された外部スタイルシートまけまグロー ジレスタイルシートを削除することができます。

スタイルレポジトリ内にある CSS スタイルの編集

スタイルパネリでは、以下の編集メカニズムを使用することができます:

• CSS スタイルシートの追加や削除を行うことができ、各 CSS スタイルシートが適用されるメディアを指定することができます。操作方法については 外部 CSS スタイルシートを参照付さい。

スタイルリポジトリウィンドウでは、グローバルスタイルのセレクターとプロパティを直接編集することができます。操作方法については、CSS スタイルをグローバルス定義するを参照ください。

□ 関連項目

- デザイン概要
- CSS スタイルと作業
- スタイルサイドバー

2.2.5 スタイル

スタイルサイドバーでは、デザインビューコで選択されたSPS コンポーネントロネルて、CSS スタイルをローカルにて定義することができます。これはスタイル・ポネーリサイドバーコングロー・バルコ定義されたスタイルとは異なるものです。



スタイルサイドバーは 以下にある2つの ピンから構成されます:

- 左側には、スタイルカラムが表示されており、選択されたコンポーネントの型が表示されます。デザインビューでは複数のコンポーネントを同時に選択することができます。選択されたコンポーネントが、コンポーネントの型ごとにスタイルカラムに表示されます。同時に編集することができるのは、表示されているコンポーネントのうち1つだけです。選択されたコンポーネントに属するインスタンスが1つしか無い場合、そのインスタンスがスタイル適用に使用されます。同一コンポーネントの型に属する複数のインスタンスが選択された場合、選択された全てのインスタンスに対してスタイルが適用されます。スタイルは(選択されたコンポーネントの)各インスタンスに対してローカルにて適用されます。特定のインスタンスに対してのみスタイルの適用を行いたい場合、デザインビューにてそのコンポーネントインスンタンスを選択し、スタイルサイドバーにてスタイルの定義を行う必要があります。Shift キーが押下された状態で複数のコンポーネントを選択することで、連続した複数のコンポーネントを選択することもできます。コンポーネント型の選択に関する詳細については、次を参照してください CSS スタイルをローカルに定義する。
- 右側には、スタイル定義ペインが表示されており、スタイルカラムにて選択されたコンポーネントの CSS スタイルを定義することができます。スタイル定義ペインは、3つのビューで表示することができます。スタイ 〒

ル定義の詳細については、<u>CSS スタイル値の設定</u> を参照くたさい。XPath アイコン により、スタイル値のソースとして XPath 条件式を使用することができます。スタイルプロパティが選択されており、XPath アイコンが有効になっている場合、XPath 条件式をプロパティに対して入力することで、XPath 条件式の値がスタイル

プロパティの値として使用されます。これにより、XML ドキュメント内にあるノードの値を、プロパティの値としてランタイムに取得することができます。XPath アイコンが無効になっている場合、静的な値をプロパティの値として入力することができます。

定義表示のオプション

編集のエーズに合わせて、定義の表示を変更することができます。以下に示されるこれらの表示オプションは、ツール デーのドダンとコンテキスト メニューから選択することができます。

- 空でないリスト
 「とは、このオプションが有効になっている場合、左側のカラムで選択されているエンポーネントにおいて、値が与えられているプレ、ティ付きがアルファベナ順で表示されるようになります。オプションが無効になっている場合、すべてのプレ、ティが表示されます。このオプションより、特定種類のエンポーネントに対してどのようなプレ、ティが定義されているか、簡単に確認することができます。選択されたエンポーネント型に対して新たなプロ、ティを定義するによ、このオプションを無効にした状態で、空のプロ、ティヘアクセスする必要があります。
- すべて展開 左側で選択されたコンポーネントレ対して、すべてのプロッティを右側のペインに表示します。このオプションは、空でないリストオプションと併用することができます。
- すべて縮退 : 左側で選択されたことポーネントに対して、すべてのプロッティを縮退された状態で右側のペインに表示します。このオプションは、空でないリストオプションと併用することができます。

重要フラグの切り替えとソールドーアイゴンのノセト

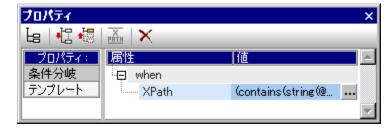
重要フラグの切り替えアイコン ● をクトックすることで、選択された CSS ルールコ対して!important とらCSS 値をセナトします。

□ 関連項目

- CSS スタイルと作業
- スタイルルポネリ
- 定義済みのフォーマ外

2.2.6 プロパティ

プロパティサイドバーにより、デザインビューにで選択された SPS コンポーネントのプロッティを定義することができます。



プロゲイサイドバーは、以下にある2つのパーツから構成されます:

サイドバー

- 左側に表示されるプロパティカラムには、選択されているコンポーネントの型が表示されます。同時に編集することができるのは、1つのプロパティだけです。コンポーネントとプロパティをグループ化する方法については、次のセグシンを参照してください、ユンポーネントとプロパティグループ。
- 右側に表示されるプロパティ定義ペインでは、プロパティカラムで選択されたコンポーネントに対してコンポーネントプロパティを定義することができます。プロパティ定義ペインは3つのビューにて表示することができます。各プロパティグループに、どのプロパティが収められているかの詳細については、以下の次のセクシュを参照してくたさい プロペティグループ。

定義表示のオプション

編集のエーズに合わせて、定義の表示を変更することができます。以下に示されるこれらの表示オプションは、ツール デーのポタンとコンテキスト メニューから選択することができます。

- 空でないリスト
 このオプタンが有効になっている場合、左側のカラムで選択されているエンポーネントにおいて、値が与えられているプロ・ティゼがグルファベナ順で表示されるようではます。オプタンが無効になっている場合、すべてのプロ・ティが表示されます。このオプタンにより、特定種類のエンポーネントに対してどのようなプロ・ティが定義されているが、簡単に確認することができます。選択されたエンポーネント型に対して新たなプロ・ティを定義するコよこのオプタンを無効にした状態で、空のプロ・ティヘアクセスする必要があります。

ツール・アイコンのナット

リナビナ アイコン 🔀 をグリックすることで、選択されているプロッティの値をデンオルトに戻します。

コンポーネントとプロケティグループ

選択されているデザインコンポーネントにより、表示されるプロッティグループが変化します。以下のテーブルに、SPS コンポーネントと それらコンポーネントに含まれるプロッティグループを示します。.

コンポーネント	プロ/ 守ィグループ
コンテンツ	コンテンツ, 共通: イベト
バ ス	テキスト; 共通: イベント
自動計算	自動計算: 共通: イベント
条件分岐	When
データ入力デバイス	共通 データ入力デバイス; イベント; HTML
イメ ー ジ	イメージ, 共通; イベント; HTML
リンク	リンク, 共通, イベト; HTML
テーブル	テーブル; 共通: イベト; HTML; インタラクティブ
パラグラフ	/ ^プ ブラフ, 共通: イベント; HTML

コンポーネントの型に対して以下の点に留意してくたさい

• コンテンツコンポーネントとは content ならびに その他のコンテンツ プレースホルダ のことです。ノードのテキストコンテンツや XML ドキュメントから得られたノードが表示されます。

- テキストコンポーネントとは、静的なテキストである文字列のことです。文字列は、テキストコンポーネントを除く全てのコンポーネント間に配置され、空白スペースを含めることもできます。
- データ入力デバイスとは、入力フィールド、複数行の入力フィールド、コンボボックス、チェックボックス、ラジオボタン、そしてボタンことで、プロパティにはデータ入力デバイスと、そのコンテンツを入力することができます。
- テーブルコンポーネントとは、デザイン内にあるテーブル構造のことです。テーブルコンポーネントには、その構成要素がサブコンポーネントとして表示されます。サブコンポーネントには行、カラム、セル、ヘッダー、そしてフッターがあります。
- 段落コンポーネントとは、定義済みのフォーマットのことです。

各プロゲイグループの説明を以下のテーブルニ示します。

プロパティグループ	説明
自動計算	自動計算が選択されている時に、このオプションが有効によいます。値フォーマットプロペティでは、数値や日付データ型といった。自動計算の書式を指定することができます。XPath プロペティでは、自動計算に使用されるXPath 条件式を指定することができます。
共通	共通プロ ティグループよ テンプレートと自動計算コレポーネトを除く全てのエレポーネトにて利用可能です。class (ケラス名)、dir (書き方の方向)、id (ユニークID)、lang (言語)、title (名前)プロ ティをコンポーネトに対して定義することができます。
データ入力デバイス	ユノボボックス、チェックボックス、そしてラジオボタインおける値の範囲を指定します。このプロペティグループは入力フィールドとボタイン対して適用されない点に注意してください。
1124	以下のケライアトサイドのHTMLイベトに対して、 <u>JavaScript関数</u> を有効にします: onclick, ondblclick, onkeydown, onkeypressed, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup.
HTML	データ入力ディス、イメージ、リンク、テーブル、パラグラフのコンポーネント型に対して利用することができます。 データ入力ディスとパラグラフ には複数の種類があり、テーブルにはサブコンポーネントが備えられている点に注意してください。 これらプロパティは対応する HTML 要素 (img、table、p、div など)に対してセントすることができます。 そのため、利用可能となるプロパティは、選択されたコンポーネントにより変化します。 これらプロパティー対応する値を XPath 条件式により選択することもできます。

更に、イメージ、リンク、ハラグラフと他の定義済みのフォーマナと条件分岐に固有のプロペティが用意されています。これらプロペティこれでは、それぞれのセグションで記述されます。

プロゲイの値の設定

プロ・ティにより、複数の方法でプロ・ティを入力することができます。

- カラムに値を直接入力する。プロパティを選択し、値カラムをダブルクリックした後に、キーボードから値を入力し、 Enter キーを押下するか、GUI をクリックすることで値を確定します。
- プロパティに対応したコンボボックスのドロップダウンリストから、値を選択する。コンボボックスの矢印をクリックして、ドロップダウンのプロパティ値オプションから値を選択します。

共通ならびにHTML グループ内のプロ ティによっては XPath 条件式を使用することでプロ ティへ値を与えることができます。XPath アイ

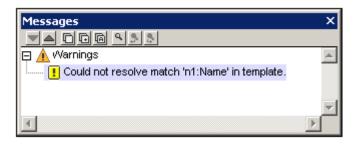
つ方効/無効を切り替えることで、XPath 条件式をプロ・ティ値のノースとして使用するか指定することができます。プロ・ティが選択されている状態で XPath アイエを有効にすると、このプロ・ティン対して XPath 条件式を入力し、その XPath 条件式の返り値を、プロ・ティの値として使用することができます。これにより、プロ・ティの値として XML ドキュメント内にある値を、ランタイムに取得することができるようごとります。 XPath アイエンが無効になっている場合、プロ・ティの値として静的な値を入力することができます。詳細については XPath を使用したスタイルプロ・ティを参照(たさい)。

プロゲイ値の修正と削除

- □ 関連項目
- スタイル

2.2.7 メッセージ

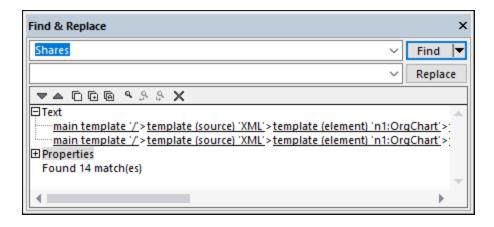
StyleVision が最初に開かれる場合、メッセージサイドバー(*下のスクリーンショット*)がシッセージウインドウのGUI に表示されます。メッセージサイドバーをオンとオフするコよ、表示 | メッセージをクトックします。



メッセージサイドバーは、デザインビューとAuthentic View 内の警告を表示します。警告は、不足している作業用 XML ファイル、デザイン構造内のエラーなど、SPS ドキュメトに関連した多くの点に関連してます。Authentic View 内では、警告は、基になるスキーマ、まだは、追加検証条件に従い、入力された XML データの有効性について説明しています。

2.2.8 検索と置換

検索と置き換えサイドバーによりデザインテキスト、スタイル、プロペティ、XPath 式内で直接入力するまだは正規表現を構成する文字列を検索することができます。デザインビューは検索は適用されます。検索メカニズムはメニューコマンド[編集 | 検索]によりアクセスされる検索ダイアログで使用されるものと同じです。 検索の結果はサイドバー内に表示されます。 結果をクリックしてデザイン内の対応するロケーションに移動することができます。 検索と置換サイドバーをオンとオフに切り替えるには、長家と置換しをクリックします。



検索と置換メカニズムに関する詳細はいピック検索、次を検索、置換の説明を参照してください。

検索

「検索対象」フィールドは検索する用語を挿入します。検索 ボダンのドロップダウン矢印をクリックして 検索オプションを選択します。以下のオプションから選択することができます:

- *選択の箇所:* (i) デザイン内のテキスト (ii) (<u>スタイル</u> サイド・ドーこて入力される) スタイル内 (iii) (<u>プロ 守ィ</u> サイド・ドーこて入力される) プロ、守ィ内。 Xpath 式内。
- 大文字と小文字/または単語の完全マッチ、これらはオンとオスさ切り替えるオプションです。
- 正規表現: エトリは正規表現とて扱われます。正規表現の使用方法に関しての詳細は検索、次を検索、置換を参照してく たさい。

結果

結果は(i) デザインテキスト(ii) スタイル(iii) プロペティと(iv) Xpath 式のためのグループに整理されます。(*上のスクリーンショナを参照してくたさい*)。各結果アイテムは検索されたアイテムの階層を表示するために構造されています。検索されたアイテムまだは階層内のアンケをクリックしてデザインビュー内のアイテムに移動することができます。

結果ペインお以下の左側からのコマドのためのアイコンを持っています: 次のおよび前のアイテムご移動する。 クトップボードにアイテムをコピーする。 結果ペイン内で検索する。 結果ペインをクリアする。

置換

置換 ポタノは検索されたすべてのアイテムを 置換 フィールド内の文字列と置き換えます。

3 クイックスタートチュートリアル

このチュートリアルでは、効果的な SPS を作成するこあさり注意すべき点を簡単に解説していきます。最初にSPS の作成とセナアップについて説明し、その後 SPS 内にエレテングを挿入する方法について、SPS エンポーネントのフォーマナ方法、そして、自動計算や条件という、SPS に搭載されている2つのパファルな機能について説明します。内容を読み進めることで、出力の構造について、また利用可能な機能について効率的な学習を行うことができます。

必要とされるファイル

このチュートリアルで使用されるファイルは、<u>(マイ)ドキュメトフォルダー</u>以下のC:\Documents and Settings\Susername>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Fュートリアル\QuickStart しれめられています:

- QuickStart.xsd: SPS のベースとして使用される XML スキーマファイルです。
- QuickStart.xml: 作業用 XML ファイルで、内部に含まれているデータが出力プレビューに表示されます。
- QuickStart.sps: 既に完成した SPS ファイルで、作成した SPS ファイルと比較することができます。
- QuickStart.css: チュートリアルで使用される外部 CSS スタイルシートです。
- NewsItems.BMP: SPS にて使用されるイメージファイルです。

チュートリアルの進め方

チュートリアルを最初から読み始め、そのままサブセクションを読み進めてくたさい。チュートリアルを始める前に、XSD ならびにXML ファイルの 内容とその構造を確認すると良いでしょう。XSD とXML ファイルを開いたままっしておくことで、チュートリアルを進めなから、それらファイルの 内容を確認することができます。QuickStart.sps 以外の名前で SPS ドキュメントを保存することで、最初から与えられている SPS ファイル を上書きすること無く、チュートリアルを進めることができます。各ステップを完了した後口よ、ファイルの保存を忘れないようにしてくたさい。

- リンク

• 次のステップ:新たな SPS の作成とセナアップ

3.1 新規 SPS の作成

このセクションでは、以下の項目について学習します

- 新規 SPS ドキュメトの作成方法。
- SPS /式してスキーマノースを追加する。
- SPSのXSLT バージョンを追加する。
- <u>作業用 XML ファイル割り当て</u>。
- 出力のエノコードの指定。
- SPS ドキュメトの保存。

このセケション内のファイル

このセグション内で参照されているファイルお以下の場所で見つけることができます: マ<u>ベドキュメトフォルダー</u>C:\Documents and Settings\<username>\My

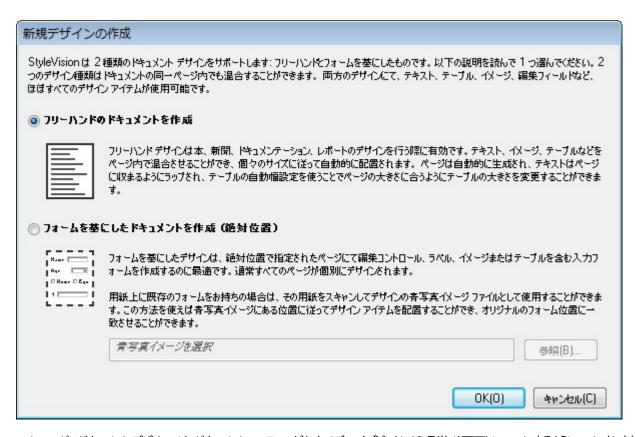
Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart:

- QuickStart.xsd はSPS がースにされてるXML スキーマファイルです。
- QuickStart.xml は出力プレニー内に表示されるデータのノースである作業 XML ファイルです。
- QuickStart.sps は完了済みのSPS ファイルです。このファイルを使用して作成したSPS ファイルを比較することができます。

新規 SPS ドキュメトの作成

新規 SPS ドキュメトを作成するコよ メニューオプションからファイル | 新規作成 | 新規作成(空) を選択するか、アプケーションソール ドースある 新規アイコン む をクトックします。

新規デザインを作成ダイアログボックスが表示され、(i) フレーハンドのドキュメントデザイン、または(ii) フォームをベース ユナギキュメントデザイン(コンポーネントが絶対位置で指定されます)から選択を行うようこ促されます。



フレー・ンドのドキュメトデザインでは、ドキュメトコンテンソが出力メディアオブジェケト(印刷物や画面)にフィナするようコンイアナされます。ドキュメトコンテンソ内のアイテムは相対的にしか配置することができず、絶対位置を指定することはできません。このオプションは、レポートや記事、書籍などのデザインに適しています。

フォーノをベースコンナギュメトロは単一のレイアナコンテナーが作成され、その中でデザインコンポーネントを絶対位置で指定することができます。レイアナコンテナーコは、絶対位置でレイアナボックスを配置することができ、各レイアナボックス内にギュメントコンテンンを配置することができます。既にあるフォームのデザインをSPS にて再現したい場合、オバンナルのフォームイメージを青写真イメージといて使用することができます。青写真のイメージをレイアナコンテナーの背景として含めることができ、フォームのデザイノニ役立てることができます。

ここではフレーハンドのドキュメントを作成します。フリーハンドのドキュメントを作成ラジオポタンをクックして、「OK」をクックしてくたさい。

SPS1.sps とう名前の新規ドキュメトが作成され、デザクビューに表示されます(以下のスクノーンショナを参照)。



<u>デザイビュー</u>にて空のメインテンプレートが表示されます。<u>デザイン概要</u> と<u>スキーマソノー</u> サイドバーにおいてスキーマエントリーは表示されません。

スキーマノースの追加

SPS 「対して QuickStart.xsd スキーマを使用します。以下の操作により、スキーマをスキーマノースとして追加することができます:

1. デザイン概要サイドバーから新しいソースの追加コマンドをクリックします。メニューが表示されるので、 XML スキーマ/DTD/XML を追加を選択します。



- 2. ファイルを開くダイアログボックスが表示されるので、<u>(マイ)ドキュメトフォルダー</u>C:\Documents and Settings\<username>\My
 - Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\QuickStart\t.xsd を選択し、開くをクリックします。
- 3. 作業用 XML ファイルを選択するように促されます。ファイルシステムからファイルを選択します。参照ボタンをクリックして、(マイ)ドキュメトフォルダー 内のC:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\QuickStart.xml を選択し、開くをクリックします。スキーマがデザイン概要サイドバー内部のスキーマソースとして追加されます(以下のスクリーンショットを参照)。更に、選択した XML ファイルが、スキーマに対する作業用XML ファイルとしてデザイン概要に表示されます。



以下の点に注意してください(i) デザイン概要内にある \$XML エントリーに、スキーマや 作業用 XML ファイルのリストが表示され、(ii) スキーマツリーサイドバーのルート要素ツリーには、スキーマ内に定義されたグロージレ要素 選択した、ルト要素(ドキュメト要素) が表示されます。このスキーマの場合、グロージ収要素 から選択した要素、この場合要素 presswatch、が表示され、(iii) スキーマ内にある グロージ収要素 がすべてのグロージ収要素ツナーに表示されます。

XSLT バージョンの選択

この例ではSPS に対してXSLT 2.0 を使用します。XSLT の、一ジョンを指定するコよ アプリケーションソール デーから 20 アイコンをクリックします。

作業用 XML ファイルの割り当てまけよ変更

これまでの操作により XML スキーマを SPS へ追加して、作業用 XML ファイルをスキーマへ割り当てまた。作業用 XML ファイルにより XML データのノースが SPS へ与えられます。作業用 XML ファイルの割り当て、変更、そして割り当て解除を行うづま、デザイン概要サイド・バー内にある 作業用 XML ファイルの行を右クトク(まざまち側にあるコンテキスト メニューアイコン をクトンク)して、コンテキスト メニューから目的のコマイ を選択します。 作業用 XML ファイルの割り当てが行われ、デザイン概要にてファイル名を入力することができます。他の操作を行う前に(マイ)ドキュメトフォルダー内にある C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\QuickStart.xml が作業用 XML ファイルとて割り当てられていることを確認してください。

出力のエンコーディングを指定する

オプションダイアログのデフォルトのエンコードタブ(ツール オプション)にてHTML エンコード Unicode UTF-8 に設定し、。

SPS ドキュメントの保存

上記のSPS のセナアップを終えた後に、MyQuickStart.sps (マイ)ドキュメトフォルダーC:\Documents and Settings\<username>\My

Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart フォルダーに MyQuickStart.sps とう名前でキュメトを保存して付さい。メニューコマ・ドファイル デザインを保存 」を選択するか、Ctrl+S を押下して、ダイアログにてファイル名を入力します。

- **」リンク**
- 次のセグション. 動的なコンテンソの挿入
- チュートリアルスタートページ

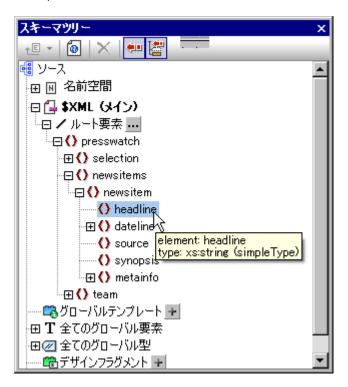
3.2 (XML ソースの)動的なコンテンツの挿入

このセグションでは、XMLドキュメト内にあるノードからデータの挿入を行う方法について説明します。スキーマソノーに表示されている要素や属性ノードをデザイン・ドラッグすることで、これらノードをコンテンソとして作成することができます。ノードがコンテンソとして作成されると、その中にあるデータが要素の子テキストノードや、その子孫テキストノードのコンテンソが文字列として出力されます。

要素コンテンソの挿入

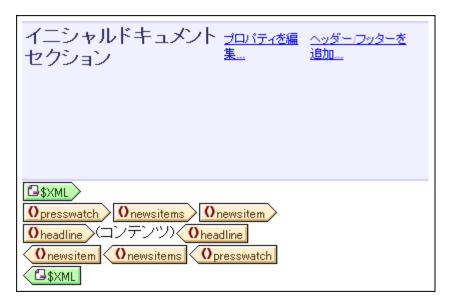
SPS において、以下の操作を行います::

1. <u>スキーマソノーサイドバー</u>において、スキーマソノーを展開して、newsitem 要素の子要素を表示します。(*下のスクリーンション*イ).



- 2. headline 要素を選択します(マウスオーバーにより、要素データ型がポップアップで表示されます)。要素を <u>デザインビュー</u>へドラッグして、矢印カーソルが挿入状態に変わすと状態でメインテンプレートへドロップすることができます。
- 3. コンテキストメニューから、コンテンツの作成を選択します。headline 要素の開始ならびに終了タグが、デザイン内のドロップされた箇所に挿入され、コンテンツプレースホルダが内部に含まれます。headline タグの周りには、その祖先要素のタグが表示されます(スクリーンショットを参照)。

デザインにて、(Enter キーを押下することで)以下のスクリーンショットにあるように改行を挿入します。



HTML タブをクリックして、HTML 出力のプレビュー(下のスクリーンションナ)を表示します。HTML プレビューには newsitem の headline 子要素がテキストとして表示されます。

NanoNull Inc Launches Version 2.0 of NanoPower

NanoNull Inc Jumps 3% on Release of New NanoPower Version

NanoNull Shares Up 10% on Month Following New NanoPower Version

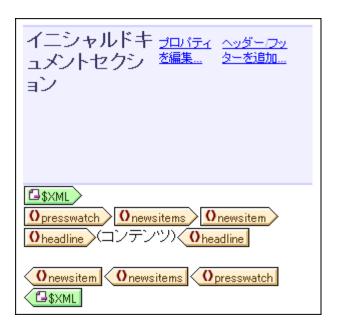
NanoDiamonds Project to Go Ahead

メモ デザイン要素の挿入ツールドーツールドーであるコンテンツを挿入をクリックして、デザイン内の位置をクリックし、スキーマセレクターツリーから目的の位置を選択することで、ノードのコンテンツを作成することもできます。

属性コンテンツを挿入する

コンテンソとして要素がデザインへ挿入された場合、属性のコンテンソは自動的に挿入されません。属性ノードをデザインへ挿入することで、属性の値が出力へ表示されるようによります。以下の操作をSPS に対して行なってください

1. カーソルを headline 要素タグの終端に配置して、Enter キーを押下します。この操作により、空の行が作成されます(スクリーンショットを参照)



2. スキーマツリーサイドバーにて、dateline 要素を展開します。



dateline 要素には date と place という2つの子要素が含まれており、place 要素には city と country という2つの属性が含まれていることに注意してください。

3. デザイン内の新たに作成した空の行に dateline 要素をドロップします。(*下のスクリーンショント*).



4. HTML プレビューへ切り替えて、dateline の出力を確認します(アのスクリーンショント)。

NanoNull Inc Launches Version 2.0 of NanoPower 2006-04-01

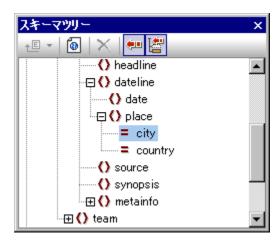
NanoNull Inc Jumps 3% on Release of New NanoPower Version 2006-04-01

NanoNull Shares Up 10% on Month Following New NanoPower Version 2006-04-25

NanoDiamonds Project to Go Ahead 2006-05-06

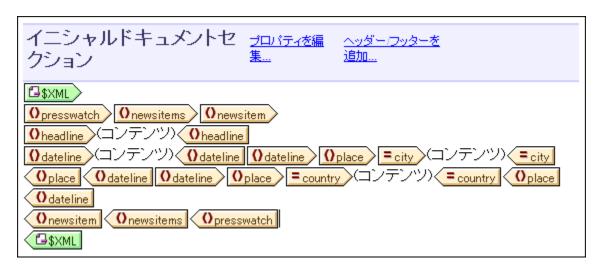
dateline 要素の子である date のコンテンツが出力されていることに注目してください。dateline の子である place に対してコンテンツの出力は行われていません。これは、place のデータが place 要素の属性(city ならびに country 属性)に含まれている一方で、属性の親要素が処理された際に属性のコンテンツは出力されない為です。

- 5. スキーマソノーサイドバーからdate 要素をドラッグして、dateline 要素の開始ならびに終了タグの間にドロップします。
- 6. スキーマンナーサイドバー内にある dateline/place 要素の city 属性を選択します(アのスクケーンショナ)。



- 7. @city 属性ノードを <u>デザインビュー</u> ペ・ラッグして、date 要素の終了タグ直後にドロップし、コンテンツを作成します。
- 8. @country ノードを <u>デザインビュー</u> ペドロップして、@city 属性の終了タグ直後にドロップし、コンテンツを作成します。.

上の操作を行った後に、SPS デザインは以下のようしているはずです:



HTML プレビュー は以下のように表示されます::

NanoNull Inc Launches Version 2.0
of NanoPower
2006-04-01BostonUSA

NanoNull Inc Jumps 3% on Release
of New NanoPower Version
2006-04-01New YorkUSA

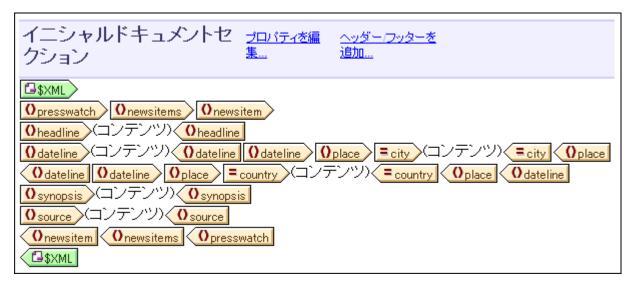
NanoNull Shares Up 10% on Month
Following New NanoPower Version
2006-04-25New YorkUSA

NanoDiamonds Project to Go Ahead
2006-05-06LondonUK

@city ならびに@country 属性の値が出力に含まれていることで注目してください。

更に動的なコンテンツを追加する

上記の方法により、XML データノースから得られた要素ならびに属性のエンセプトを、デザイン内の任意の場所へ挿入することができます。このセグションを終了する前に、以下のスグノージョントにあるように synopsis ならびに source 要素をデザインへ追加してくたさい



synopsis 要素がsource 要素の前に配置されていることは注意してくたさいスキーマ内の順番と異なっています)。synopsis とsource 要素をデザインにはいた後に、HTML プレニーから出力を確認してくたさい。メインのデンプレート内に配置されたノードの順番により、出力に表示される順番となります(ドキュメトの構造と出力ドキュメトに関する詳細については、デンプレートとデザインアグメトを参照ください。

その他にも、デザイン内に作成されるノードを含む形式(フォーム)について留意する必要があります。 HTML プレビュー では、デザイン内にある全てのノードが、テキスト文字列として出力へ送信されます。テキスト文字列として出力する代わりに、テーブルやコンボボックスといす。、その他の形式での出力を行うこともできます。このセグションでは、全てのノードを(コンテンツ)として作成してきな、これら全てのノードはテキスト文

字列として出力されます。 条件を使用する セクションでは、エンボドックスとしてノードを作成する方法について学習し グロー・バリテンプレート とその他のエンテンツの使用 セクションでは (動的な)テーブルとしてノードを作成する方法について学習します。

次のセグションを始める前に、ファイルを保存するようはしてくたさい。

ロリンク

- 次のセグション 静的コンテンツの挿入
- <u>前のセグション</u> 新規 SPS の作成とセナアップ
- チュートリアルスタートページ
- SPS ファイル構造

3.3 静的コンテンツの挿入

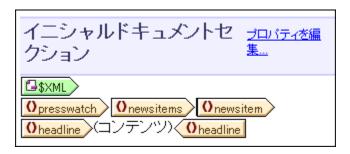
静的コンテンツ出ま、XML ソースから得られるコンテンツ出お違い、デザイン内に直接入力まけお挿入されるコンテンツのことです。 SPS デザインコは様々な静的コンポーネントを配置することができます。 このチュートリアルセグションでは、以下の静的コンポーネントを挿入します:

- イメージ
- 水平線
- 対

静的イメージを挿入する

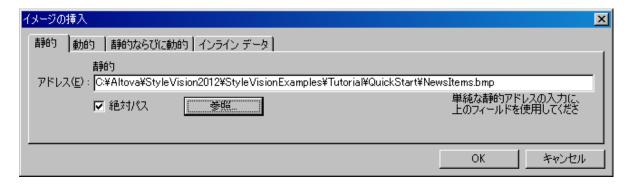
挿入する静的イメージは、<u>(マイ)ドキュメトフォルダー</u>以下の: C:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP
で、ドキュメトのヘッダーとして使用されます。以下の操作により、このイメージをドキュメトの先頭へ挿入することができます:

1. newsitems と newsitem の開始タグの間にカーソルを配置します(以下のスクリーンショットを参照)。



カーソルが newsitems 要素の内側、かつ newsitem 要素の外側に配置されます。そのため、newsitems 要素が処理される際に、1度だけ挿入されます(スキーマ内では newsitems 要素が1度だけ定義されているため)。

2. カーソル位置で右クリックして、<u>挿入 | イメージ</u>を選択します。イメージの挿入ダイアログが表示されます(以下の スクリーンショットを参照)。



- 3. 静的タブにて絶対パスのチェックボックスをクリックして、NewsItems.BMP を参照ボタンから選択します。
- 4. 「OK」をクリックして完了します。

HTML プレビューが以下のように表示されます:

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower 2006-04-01BostonUSA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed. NewTech Online

水平線を挿入する

以下の操作を行うことで、ドキュメントのヘッダーとドキュメントのボディーの間に水平線を挿入します:

- 1. 先ほど挿入した静的イメージの直後にカーソルを配置します。
- 2. その位置で右クリックして、挿入 水平線を選択します。水平線が挿入されます。

水平線のプロ ティを以下の方法でセナします:

- 1. <u>デザインビュー</u>にて水平線が選択されている状態で、<u>プロッティサイドバー</u>にあるラインコンポーネントを選択し、プロパティの HTML グループを開きます。
- 2. 水平線に対して color と size プロパティをセットします。
- 3. デザンビューにてラインが選択されている状態でスタイルサイドバーにあるラインコンポーネントを選択し、プロペティのボックスグループを開きます。margin-botton プロペティの値に12pt をセットします。
- 4. HTML プレビューインで出力を確認します。

各ニュースアイテムの最後に水平線を挿入することはします。newsitem 要素にある終了タグ直前にカーソルを配置することで、各 newsitem 要素の終端にラインが表示されるようになります。ラインの size プロノ ティー単位無しの数値を入力することで、水平線の太さを変更することができます(プロノ ティサイド・バー・エーラインを選択し、size の値に例えば3 を入力してみてくたさい)。

静的なテキストを挿入する

実は(XML ソースから動的なコンテングを挿入するセクションで Enter キーを押下することで、改行という静的なテキストが既に追加されています。このセクションでは、静的テキストとしてデザイン内に文字を追加します。

ここまでデザインした SPS の出力結果は、以下のようになります:

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower 2006-04-01BostonUSA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

NewTech Online

dateline 要素内にdate 要素とplace/@city ならびにplace/@country 属性のコンテンンが、空白スペース無しつ出力されている点に注目してください。空白スペースは静的テキストとしてデザインに追加することができます。デザインに、date 要素の直後にカーツルを配置して、コロンとスペースを入力します。その後、@city 属性の直後にコンマを入力します(以下のスクノーンショナを参照)。



出力は以下のようなはます

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower 2006-04-01: Boston, USA

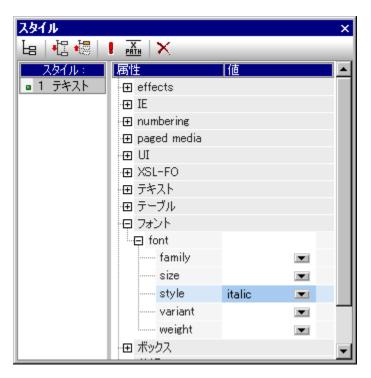
dateline 出力にてコロンとコンマ、そしてスペースが表示されます。これらテキストアイテムは、全てデザインへ直接挿入されたテキストアイテムです。

静的テキストを更に追加します。デザインにて、"Source: "というテキストをsource 要素の開始タグ直前に入力します。



静的なテキストをフォーマナする

静的なテキストをフォーマナするコよ、目的のテキストを・イライトして、ローカルのスタイルプロ・ティを指定します。デザイノニて入力した "Source: "というテキストを・イライトします。 スタイルサイドバー にて、1 テキストコンポーネントを選択します。 以下のスクリーシショナ にあるように、プロ・ティのfont グループを展開して、font-style プロ・ティーで(talic オプションを選択します。(下のスクリーンショナ)。



静的テキスト(ここでは "Source: ") かデザイノこてイタノック状態になります:



HTML プレビューの出力は以下のようけなります::

Summary of News Items

NanoNull Inc Launches Version 2.0 of NanoPower 2006-04-01: Boston, USA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

ソースアイテムと水平線アイテムの間にある垂直のスペースが狭すぎると思える場合、デザインにてノースと水平線の間に(Enter キーを押下することで)空の行を挿入してください。

操作が終わする後に、ファイルの保存を行います。

このセグションでは、静的コンテンソの挿入とフォーマナト方法について学習しました。次のセグションではCSS を使用することには、デザインコンポーネトをフォーマナする方法について学習します。

= リンク

- 次のセグション: コンテンツのフォーマット
- 前のセグション. 動的なコレテンソの挿入
- <u>チュートリアルスタートページ</u>

3.4 コンテンツのフォーマット

StyleVision はよ CSS をベース・したパフルで柔軟性に富んだ、スタイルメカニズムが用意されており、デザイン内にあるエンポーネトのフォーマトに使用することができます。StyleVision に搭載されたスタイルメカニズムの重要な点を以下に示します:

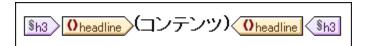
- CSS スタイルルールをブロックコンポーネントならびにインラインコンポーネントに対して定義することができます。
- <u>定義済みのフォーマナ</u> とは承継されるスタイルを含むブロックコンポーネトのことで、コンポーネトのグループに対するラッパーとして 使用されます。これら定義済みフォーマトの継承されたスタイルは、各コンポーネントに対してローカルでスタイルを指定することに り、オーバーライドすることができます。これは CSS のカスケード処理に従ってきのです。
- デザインではコンポーネントに対するクラス属性を宣言することができ、外部 ませまグローバレスタイルレールのセノクターとしてクラスを使用することができます。
- スタイルの指定は3つのレベルで行うことができます。優先順位の低いものから、(i) <u>外部スタイルシート</u>、(ii) グローブルスタイルレール と(iii) ローカルスタイルレールとなります。

このセグションでは、以下のことがらを学習します:

- 定義済みのフォーマルを割り当てる
- ユンポーネントにクラス属性を割り当てる
- 外部 CSS スタイルシートしてスタイルを定義して そのスタイルシートを SPS のスタイルノポネット追加する
- グローバルスタイルルールを定義する
- 選択された複数デザインコンポーネントに対してローカルスタイルを定義する
- 単一のエンポーネトレ対してローカルスタイルを定義する。

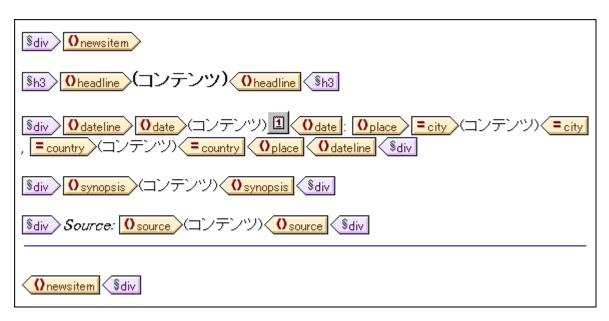
定義済みのフォーマトを割り当てる

定義済みのフォーマナ を割り当てることで 定義済みのフォーマナ のスタイルを、継承と らかたちでコンポーネトへ与えることができます。 デザインには、て headline 要素を選択し、メニューオプションから タグで 囲む | 特殊な段落 | 見出し3 (h3)」を選択します。 定義済みのフォーマナタグが headline 要素の周りに作成されます (下のスクノーンショナ)。



コンテンソのフォントプロ 守っか変化して、定義済みフォーマトの上と下に空白が追加されます。プロ 守ィ値はh3 定義済みフォーマトから継承されます。

定義済みのフォーマトを使用することで、ブロック内にあるエポーネトをグループとしてデザイン、それらエポーネトをブロックやグループとしてフォーマトすることができます。この中でも特にdiv フォーマトを使用することで、空白なしのブロックを作成することができます。デザインにて newsitem、dateline、synopsis、そして source ノードにdiv エンポーネトを割り当てます。デザインは以下にあるスクレーシンョナのようになります。静的テキストの"Source: "が source 要素を含む div エンポーネト内に含まれるようになり、newsitem 要素全体が div エンポーネ・ト内に含まれます。

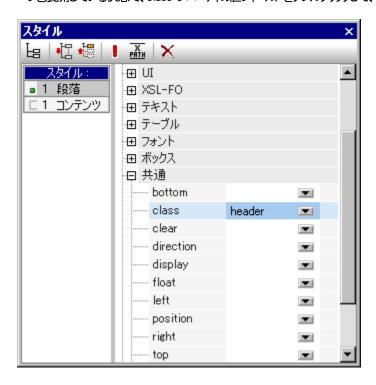


div ブロック内とともに、コンポーネトがブループ化されました。 <u>このセグションの終盤</u> では、グループ化されたコンポーネントのブロックス対してスタイルの割り当てを行います。

コンポーネントをクラス属性に割り当てる

スタイルはエンポーネトのケラスコ対して定義することができます。例えば、全てのヘッダーコ対して(フォントファミリーや色といす) 共通したプロ 守っを定義することができます。 定義を行う口は (i) 共通したプロ 守っを含んだエンポーネントを割り当て、(ii) そのケラスコ対してスタイルプロ 守っを割り当てる、という操作を行う必要があります。

デザインにて h3 タグを選択し、スタイルサイドバーにて 1 段落を選択することで、共通プロ ティグループを選択します。共通プロ ティグループを展開している状態で、class プロ ティの値フィールドをダブルクトックして、header という値を入力します。



h3 フォーマナのインスタンスがheader という名前のケラスへ割り当てられます。header ケラスコ対してスタイルプロ ティを定義した場合、これらのプロ ティがheader ケラスを持っている SPS 内の全てのエンポーネントロ対して適用されます。

外部 CSS スタイルシートをスタイルレポジトリへ追加する

外部 CSS スタイルシート内にあるスタイルレールは SPS デザイン内にあるコンポーネントへ適用することができます。外部スタイルシートのルールをコンポーネントに対して適用するナンガロは、スタイルシートをスタイルルポントリール追加することができます。 スタイルレポントリナイドバー (デザインビュー内の),にて、以下のような操作を行います:

- 1. 外部アイテムを選択します。
- 2. Zタイルポネリサイバーのソール・イスる追加ボタンをクリックします。ファイルを開くダイアログが表示されます。
- 3. (マイ)ドキュメトフォルダー以下にあるC:\Documents and Settings\<username>\My Documents¥Altova¥StyleVision2021¥StyleVisionExamples¥チュートリアル \OuickStart\OuickStart.css を開きます。

スタイルシートがスタイルノポネーリへ追加されます。この段階で関連する記述を以下に示します:

```
header

font-family: "Arial", sans-serif;
font-weight: bold;
color: red;
}

h3

font-size: 12pt;
}
```

header クラスとh3 要素のスタイルレールを組み合わせることで、以下のHTML 出力がheadline 要素に対して作成されます。

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

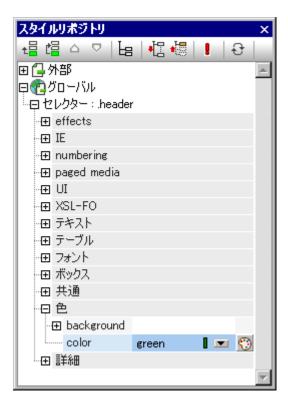
Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

グローバルスタイルルールを定義する

CSS セレクターを使用することで、グロー・ジスタイルレールをSPS 全体に対して定義することができます。ルールはスタイルレポトリサイドバー内で直接定義します。header クラスニ対して、以下のようにグロー・ジスタイルレールを作成します:

- 1. デザクビューがアクティブ さいている状態で、スタイルルポトリサイドバー内のブローバルアイテムを選択します。
- 2. ツールバーの追加ボタンをクリックします。この操作により、ワイルドカードセレクター (*) に対して空のルールが作成されます。
- 3. ワイルドカードの代わりに header と入力して、セレクターを置き換えます。
- 4. プロパティの color を展開して、color プロパティ値のドロップダウンリストから green を選択します(以下のスクリーンショットを参照)。



外部スタイルシートでも定義されているプロ・ティー対してグロー・ジレスタイルレールの定義を行うと、グロー・ジリレールのプロ・ティ値が優先的に使用されます。そのため、以下のHTML プレビューではヘッドラインのエレテンンが縁色で表示されます。外部スタイルシートにて定義されたその他のプロ・ティは、グロー・ジリレールにより上書きされることなく)そのまま使用されます(この例では、font-family とfont-weight)。

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

複数のコンポーネントノス対してローカルスタイルを定義する

複数のエンポーネントに対してローカルスタイルを同時に定義することができます。作成中のデザインにて、ニュースアイテムのテキスト全体に Arial フォントを使用したい場合、newsitem 要素を囲んでいる div エンポーネントをクックして、スタイルサイドバーのスタイルカラムにて、「1 段落」を選択し、プロ ティのフォントグループ内にある font-family IC Arial を選択します。この操作により、それから派生した形式全てか影響をうけることしています。

デザイノコで、dateline、synopsis、source ノードを囲んでいる3つのdiv ユンポーネントをクトックすることで選択します。 スタイルサイドバー にて 3 段落」を選択し、プログライのフォントグループコで、font-size を 10pt に指定します(h3 エンポーネントは必須項目のfont-size 12pt どなっているか、選択されませんでした)。

のスタイルカラムこで 1 段落」を選択します。プロ・ティのフォ・トグループ こて、font-weight をbold にセットして、font-style をitalic に指定します。プロ・ティの色グループ こて、color を gray に指定します。 dateline の出力は以下のようになります。 スタイルサイドバーのスタイルカラムこで 1 段落」を選択します。プロ・ティのフォ・トグループ こて、font-weight を bold にセットして、font-style を italic に指定します。 プロ・ティの色グループ こて、color を gray に指定します。 dateline の出力は以下のようになります。

2006-04-01: Boston, USA

div コンポーネントしつ対して定義されたスタイルが、div コンポーネント内にある静的なテキストしつ対しても適用されます。

単一のコンポーネントノン対してローカルスタイルを定義する

単一のエンポーネント「文がして定義されたコーカルスタイルは、SPS 内で定義されたより高次のエンポーネントをオー・デライドします。デザイン 「こて headline 要素を選択し、(スタイルプロ・ディの色グループ内にある color プロ・ディース対して) navy 色を割り当てます。ローカルで定義されたプロ・ディ(color:navy) により、header クラスコンプして定義されたプロー・ジレスタイル(color:green) がオー・デライドされます。

source 要素を囲んでいる div コンポーネントを選択します。 スタイルサイドントーにで 1 段落」アイテンを選択し、(スタイルプロ) 守ィの色グループ こある) color プロ ティーて gray をセナします。 スタイルプロ ティのフォントグループ こて font-weight の値を bold へ変更します。 これらの値は静的なテキスト 「マ対して適用されます。 前のセグションこて、font-style の italic か静的テキストの "Source:" 「マ対して適用され ナーのを思い出してください。 新たなプログライ (font-weight:bold とcolor:gray) が font-style:italic プログライとお別に追加されます。

デザインビューに source 要素の(コンテンツ)プレースポルダを選択します。スタイルカラムに「1 コンテンツ」を選択し、(スタイルプロ 守ィの色グループにある) color プロ 守ィこblack をセナします。その後、プロ 守ィのフォントグループにある font-weight に対して normal を選択します。新たなプロ 守ィが source 要素のノードに対応したコンテンソプレースポルダノードに適用され、div コンポーネントに対して定義されたプロ 守ィがナー ドライドされます(以下のスクノーンショナを参照)。

フォーマトを完成する

このセグションこでフォーマルを完成させるこあたり、synopsis 要素のdiv コンポーネントを選択し、ツールドーの定義済みのフォーマルコンドドックスこて p を選択します。この操作によりHTML のp 要素のスタイルがブロックに対して適用されます。HTML プレビューは以下のようにないます。

NanoNull Inc Launches Version 2.0 of NanoPower

2006-04-01: Boston, USA

Nanonull Inc today launched version 2.0 of its market-leading NanoPower line of hardware and software. The highlights of the new version of NanoPower are improved chip design capabilities in NanoSoft, the software used to design computer chips; and higher precision sand-filtering processes in NanoMeld, the hardware in which the transformation from sand to silicon chips is executed.

Source: NewTech Online

操作が完了した後に、ファイルを保存してくたさい。

- *□* リンク
- 次のセケション: 自動計算の使用前のセケンョン: 静的コンテンソの挿入
- チュートリアルスタートページ
- <u>CSS スタイルと作業</u>

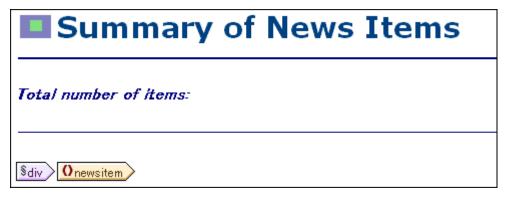
3.5 自動計算の使用

<u>自動計算</u>は、XML内にあるデータをベースに更なる情報を提示するナックのパファルなメカニズムです。このセグションでは、デザインに対して、XMLドキュメト内にあるニュースアイテムの総数と、ニュースアイテムによりが、一された期間という、2つの情報を追加します。どちらの情報もXMLドキュメト内に記述されていませんが、利用可能なデータから計算し、操作することで得ることができます。

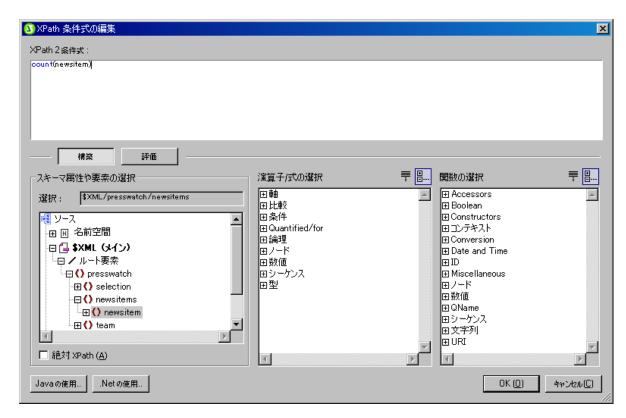
ユースアイテムのノードを数える

デザインにて、以下を行います:

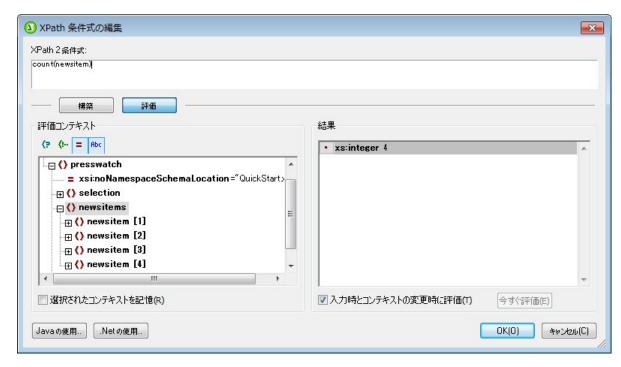
1. 以下のスクリーンショットに示されるように、空白行を挿入します(この場所に自動計算が配置されます)。 Return キーを押下して新たな行を追加して、その下に水平線を追加します(以下のスクリーンショットを参照)。



- 2. 上にスクリーンショットに示されているように、"Total number of news items: " という内容の静的なテキストを入力します。
- 3. 静的テキストに対してローカルなスタイルルールを適用します。 <u>コンテンソのフォーマナ</u>に記述されている操作を行います。
- 4. コロンの後にカーソルを配置して、「挿入 | 自動計算を挿入 | 値」を選択します。XPath 式 の編集ダイアログ ダイアログが表示されも明日(以下のスクリーシショナを参照)。デザインを右クトックして、コンテキストメニューからコマンドを選択することもできます。



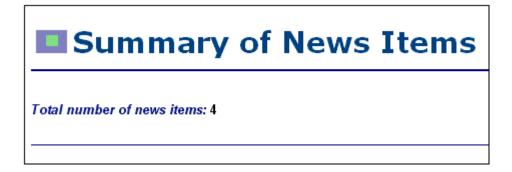
5. スキーマツリーにて newsitems がコンテキストノードとしてハイライトされていることに注目してください。条件式テキストボックスにて、count(newsitem) という条件式を入力するか、条件式テキストボックス以下にある入力へルパーペインを使用して条件式を構築することができます。関数のシーケンスグループにあるcount 関数をダブルクリックすることで条件式ペインに関数が入力されるので、関数のかっこ内にカーソルを配置して、スキーマツリーの newsitem ノードをダブルクリックしてください。評価ボタンをクリックすることで、XPath 条件式の戻り値を確認することができます。評価の結果が結果ペインに表示されます(以下のスクリーンショットを参照)。XPath 条件式の編集ダイアログに関する詳細については、次のセグシエを参照してください、XPath 式を編集。



6. 「OK」をクリックして完了します。デザイン内のカーソル位置に自動計算が挿入されます(以下のスクリーンショットを参照)。 ローカルスタイルには自動計算をフォーマットすることができます。



HTML 出力 は以下のようごなります:



ニュースアイテムによりか、一された期間を表示する

ニュースアイテムにより、ドーされた全ての期間は、一番古い記事の月日と、一番新しい記事の月日により求めることができます。以下にあるような XPath 条件式により、このような処理を行うことができます。以下に示される最初の条件式により、date ノードのコンテン・幼単純に出力され、2番目の条件式により、date ノードにある年や月だけが出力されるようによります。 どちらの条件式も使用することができます。

- concat(min(//date), ' to ', max(//date)).
- concat(month-from-date(min(//date)), '/', year-from-date(min(//date)), ' to ', month-from-date(max(//date)), '/', year-from-date(max(//date)))

デザイノにて、以下に示されるように静的テキストと自動計算が挿入されます。希望するローカルスタイルを適用することができます。

Total number of news items: =(AutoCalc)
Period covered by news items: =(AutoCalc)

HTML プレビューは以下のように表示されます:

Total number of news items: 4
Period covered by news items: 4/2006 to 5/2006

操作が完了した後にファイルを保存してくたさい。

□ リンク

- 次のセケション: 条件を使用する
- 前のセグション: コンテンソのフォーマ外
- チュートリアルスタートページ

条件の使用 3.6

QuickStart.xml でよ 各 newsitem 要素内にmetainfo 子要素が含まれており、その中に1つ以上のrelevance 子要素を含めることがで きます。各 relevance ノードコよ ニュースアイテムの類似性を表す見出しか含まれます。その他に

も、/presswatch/selection/byrelevance とらノードかわます。このノードのコンテンソコは関連する見出しか含まれており、どのニュース アイテムを表示するか決定することができます。例えば、byrelevance ノードのコンテンンがNanoPower となっている場合、NanoPower とら コンテンを含むrelevance ノードを持ってのニュースアイテムが表示されます。条件により、ノードをリックアップすることで。byrelevance ノ ードのコンテン:かテストされ、ユーザーの各選択に対して対応する処理や表示を行うことができるようになります。このセケションでは、ユーザー の選択にマッチした relevance 要素が含まれるこれらニュースアイテムを表示するための条件テンプレートを作成します。

以下のように操作を行います

- 1. which byrelevance ノードの値を表示するコンボボックスを作成します。コンボボックスのドロップダウンリス トに表示される値は、XPath 条件式により取得され、relevance ノードのユニークな値からなるリストを動的 に作成します。
- 2. newsitem 要素の周りに条件を挿入します。この条件により、byrelevance ノードのコンテンツにマッチした relevance 要素を含む全ての newsitem 要素が選択されます。条件の分岐に囲まれたコンテンツのことを、 条件テンプレートと呼びます。
- 3. 条件テンプレート内にて、ニュースアイテムの各 relevance ノードを表示します。
 4. byrelevance 要素にマッチした(relevance 要素のリスト内にある) relevance 要素をハイライトします。この操作 は、relevance 要素に対する条件を作成し、条件テンプレートに対してフォーマットを適用することで実現で きます。
- 5. newsitem 要素に対する条件内に、全てのニュースアイテムを選択する分岐を挿入します。

ユニーケなノード値を選択するコンポポックスを作成する

XMLドキュメトでは、ユーザーの選択が/presswatch/selection/byrelevance 内に含まれます。 ユルドドソフスとしてこのノー ドを作成します。 手順は 以下のとおりです:

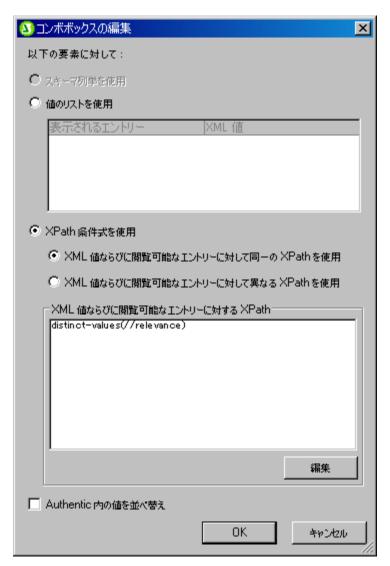
1. ドキュメントの先頭にある2番目の自動計算の下に"Select by relevance: "という静的テキストを挿入します (以下のスクリーンショットを参照)。

Total number of items:=(自動計算) Period covered by news items =(自動計算) Select by relevance:

2. (以下のスクリーンショットにて示される) スキーマソノー サイ・バーから byrelevance ノードをドラッグし、先ほと新た に入力した静的テキストの後にドロップします。



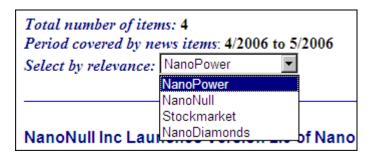
3. コンテキストメニューが表示されるので、コンボボックスの作成を選択します。以下のダイアログが表示されます。



- 4. コンボボックスの編集ダイアログにて、「XPath 条件式を使用」を選択し、「XML 値ならびに閲覧可能なエントリーに対して同一の XPath を使用」を選択します。XML 値ならびに閲覧可能なエントリーに対する XPath にて、distinct-values(//relevance) という XPath 条件式を入力します。この条件式により、XML ドキュメント内における全ての relevance 要素のユニークな値が選択されます。 relevanceノードの値は HTML コンボボックスに表示されるものの、HTML プレビューで値の選択を行なっても、XML ドキュメント内のノードコンテンツが影響を受けることは無い点に注意してください。HTML ドキュメントは XML ドキュメントを変換することで得られる出力で、入力を受け取る事はありません。ここでは、コンテンツの表現を行う手段としてコンボボックスを使用しました。
- 5. 「OK」をクリックして完了します。コンボボックスが挿入され、デザインは以下のように表示されます:



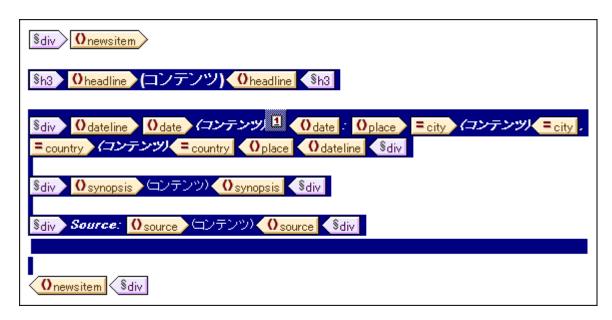
6. <u>HTML プレビュー</u> へ切り替えます。 エンドドックスの矢印をクトックすると、全てのrelevance ノードにあるユニークな値がリスト表示されます。 XML ドキュメントの内容が合致するか確認してみてください。 このリストは動的に生成され、 XML ドキュメントに新たに加えられた relevance 値も反映されます。



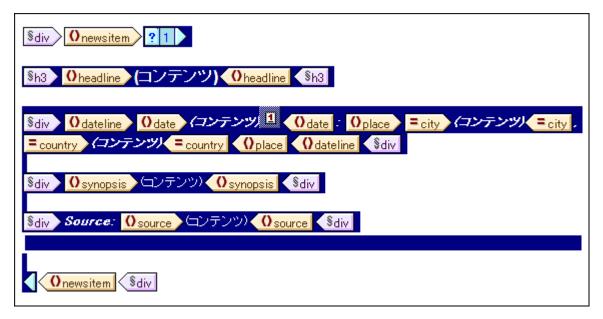
条件を挿入することで選択されたrelevance に対するニュースアイテムを表示する

/presswatch/selection/byrelevance 要素内でmetainfo/relevance 要素を持ったnewsitem 要素が選択されます。以下のように条件を挿入します:

1. デザインの newsitem におけるコンテンツを選択します(以下のスクリーンショットにおけるハイライト箇所)。



- 2. メニューコマンド(またはコンテキストメニューコマンド)から「 <u>タグで囲む | 条件</u> 」を選択します。これにより<u>XPath 式</u> の編集ダイアログが表示されます。
- 3. metainfo/relevance=/presswatch/selection/byrelevance という条件式を入力します。条件式により、現在の newsitem の metainfo/relevance 子孫が、(ユーザーの選択値である)/presswatch/selection/byrelevance に一致する場合に真となります。
- 4. 「OK」をクリックします。newsitem 要素のコンテンツの周りに条件が作成されます(以下のスクリーンショットを参照)。

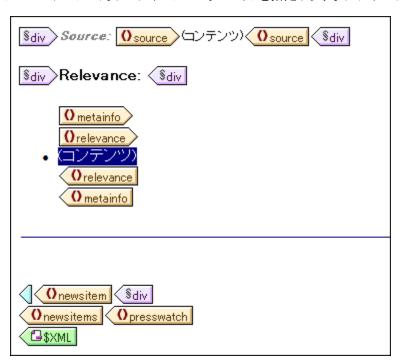


この段階では条件分岐がつしかない点に注目してくたさい。条件の結果が真となるニュースアイテムが表示され、条件の結果が真とならないニュースアイテムは表示されません。この場合、条件はフィルターとして機能します。このセグションの後半では、2番目の条件分岐を追加します。

relevance ノードをリストとして挿入する

各 newsitem 要素のrelevance ノードを表示するためづよ 以下の操作によりにれらノードをデザイン内に挿入します(以下のスクリーンショナを参照):

- 1. source 要素の div タグと条件テンプレートの終了との間に改行を作成します。
- 2. "relevance:"という静的テキストを入力し、その回りに定義済みフォーマットの div を作成します。
- 3. スキーマソノーサイドバーのレト要素ツノーからrelevance 要素をドラッグし、relevance: 静的テキストの下にドロップします。
- 4. リストを作成します(コンテキストメニューが表示されるので、箇条書きの作成を選択し、目的のフォーマットを選択します)。
- 5. リストのコンテンツに対してテキストのフォーマットを指定します。デザインは以下のようになります:

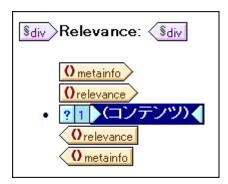


ここでHTML プレビューーこで、コンボドンクスの選択肢を変更することでrelevance の選択結果を確認してくたさい。以下を行います: (i) XMLドキュメト内のbyrelevance ノードの値を変更する。(ii) XMLドキュメントを保存する。(iii) Style Vision 内でSPS を開きます。

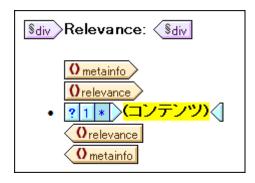
選択された relevance 要素を太字にする

ニュースアイテムこよっては、複数のrelevance 要素を含むものもあります。そのようは場合、ユーザーの選択に合致したrelevance をいイライトすることで、デザインを改善することができます。以下の操作によりデザインを拡張します:

- 1. デザイン内の relevance 要素を選択します。
- 2. =/presswatch/selection/byrelevance という XPath 条件式とともに、条件を挿入します。この操作により、分岐が1 つの条件が作成され、byrelevance 要素にマッチする relevance 要素が選択されます(以下のスクリーンショットを参照)。



- 3. コンテンツプレースホルダを選択し、ローカルのフォーマットとして(フォントグループから)太字を表す bold と、 (色グループから)背景色の background-color に yellow を(スタイルサイドバーにて)指定します。
- 4. 条件を右クリックして、コンテキストメニューから条件分岐のコピーを選択します。5. XPath 式 の編集ダイアログ にて、(ダイアログ右上にある)「それ以外」のチェックボックスをクトックします。
- 6. 「OK」をクリックして完了します。新たな(その他)の条件分岐が作成されます(以下のスクリーンショットを参 照)。この条件分岐では、byrelevance 要素にマッチしない全ての relevance 要素が選択されます。



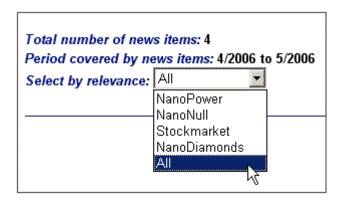
7. その他の条件分岐にあるコンテンツは、最初の条件分岐からのコピーとなります。コンテンツのプレースホルダ には太字で黄色の背景色が与えられます。コンテンツプレースホルダを選択し、これらのフォーマット情報を削 除してください。

これで、relevance 要素に対して以下のような動作を行う2つの条件分岐を持って条件が作成されました: relevance のコンテンソ が/presswatch/selection/byrelevance のエレテンソニマッチすればrelevance のエレテンツが太字に黄色の背景色で表示され、それ以 外の場合は(2番目の条件分岐により通常のフォーマナでコンテンツが表示されます。HTML プレビューに動作を確認してくたさい。

コンポポックスを修正して、2番目の条件分岐を挿入する

コンボドックス内の、全てのニュースアイテムを選択するオプションが表示されていません。以下の操作により、全てのアイテムを選択するオプショ ンを加えることができます:

- 1. デザインビューにてコンボボックスを選択します。
 2. プロパティサイドバーにてコンボボックスが選択された状態で、(コンボボックスグループ内にある)コンボボック ス入力値プロパティの編集ボタンをクリックします。
- 3. コルドックスの編集 ダイアログニて、XPath 条件式をdistinct-values(//relevance) から、distinct-values(//relevance), 'All' へ変更します。この操作により、"All" という文字列が XPath 条件式により返されるアイテムのシーケンスに追 加されます。
- 4. HTML プレニーにてコンボボックスのドロップダウンリストの選択肢を確認します(以下のスクリーンショットを参 照)。(*下のスクリーンショッ*ト).



値 All をbyrelevance ノードを入力することができます。 byrelevance ノードに "All" という値が含まれる場合、全てのニュースアイテムが表示されるようになります。

ニュースアイテムテンプレートを表示する条件には metainfo/relevance=presswatch/selection/byrelevance と う分岐だけ が与えられてします。 metainfo/relevance ノードに All とう値は無しため、 byrelevance ノードに All とう値が選択されても、ニュースアイテムは表示されません。 そこで、 All とう値に対してマッチする2番目の条件分岐を作成する必要があります。 条件の結果が真となって場合に、条件分岐にあるテンプレートが表示されます。 以下の操作をおこなってください

- 1. デザインビューにて、ニュースアイテムの条件を選択します。
- 2. 条件を右クリックして、コンテキストメニューから条件分岐のコピーを選択します。
- 3. XPath 式 の編集ダイアログ に/presswatch/selection/byrelevance='All'と入力します。
- 4. 「OK」をクリックして完了します。2番目の分岐が作成されます。

2番目の条件分岐には、最初の分岐と同一のエンテンンが含まれます。 byrelevance ノードのエンテンンがAII の場合、2番目の分岐にはニュースアイテムテンプレートの出力を行います。

操作が完了した状態でファイルを保存してくたさい。

ロリンク

- 次のセケション・グロー・バリテンプレートとその他のコンテンソの使用
- 前のセクション. 自動計算の使用
- チュートリアルスタートページ
- <u>条件</u>

3.7 グローバルテンプレートとその他のコンテンツの使用

グロー・バーテンプレートを使用することで、グロー・バルン要素の処理を指定することができます。この機能により、(1つの場所で定義された)グロー・バーテンプレートを、スタイルシート内の複数の場所で使用することができるようごなります。 グロー・バーテンプレート は2種類の方法で使用することができます:

- グローバルテンプレートのルールをローカルテンプレートへコピー。
- (メインテンプレート内にある)ローカルテンプレートから、ローカルにあるノードの処理をグローバルテンプレートへ渡すことができます。グローバルテンプレートが実行されると、メインテンプレートにて処理が再開されます。この場合、グローバルテンプレートが、メインテンプレートから呼び出された、または使用されたことになります。

メインテンプレートからグロー・ ジレテンプレート を呼び出すけるま 2つのメカニズムがおります:

- ローカルテンプレートからグローバルテンプレートへの参照。
- メインテンプレート内にあるその他のコンテンツ (rest-of-contents) 処理命令を、現在の要素の子孫要素(つまり現在の要素から見たその他のコンテンツ)へ適用。子孫要素に対応したグローバルテンプレートが存在する場合、その要素に対してグローバルテンプレートが適用されます。そうでない場合、内蔵のテンプレートがその要素に対して適用されます。要素に対する内蔵のテンプレートにより子要素が処理され、要素のテキストコンテンツが出力されます。結果として、子孫要素にある全てのテキストコンテンツが出力されます。要素の属性は出力されないという点に注意してください。

このセクションでは、その他のエレテンツ処理命令とグローノ、以要素のmember に対するグローノ、シリテンプレートを使用します。

rest-of-contents (その他のコンテンツ)処理命令を挿入する

スキーマの大まかな構造を以下のスクリーンショナに示します。



ドキュメト要素のpresswatch はよ(i) selection、(ii) newsitems、(iii) team とう3つの子要素が含まれて、ます。これまで作成したメインテンプレートでは/presswatch 要素が処理されます。presswatch 要素内ではnewsitems 要素だが処理されます。selection ならびにteam 要素は(selection はnewsitems 要素内部で処理されているものの) presswatch 要素内部で処理されません。presswatch へその他のコンテンツ処理命令を挿入することで、selection ならびにteam 要素が処理されるようけないます。

newsitems とpresswatch の終了タグ間にカーソルを配置し、メニューコマンドまけっぱンテキストメニューから挿入 1 その他のコンテンツを 選択することで、デザインへその他のエレテンン処理命令が挿入されます。その他のエレテンソプレース村以が挿入されます(以下のスクレーン ショナを参照)。



HTML プレビューを確認すると、テキストの文字列が表示されているはずです(以下のスクリーンショナを参照):

AllAndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.edwar

内蔵のテンプレートをselection ならどにteam 要素へ適用した結果、この文字列が表示されます。要素に対応した内蔵のテンプレートが 子要素の処理を行います。テキストノードに対する内蔵のテンプレートにより、テキストノード内にてテキストが出力されます。これら2つの内蔵 テンプレートを組み合わせることで、selection ならびにteam 要素以下にある全ての子孫ノードのテキストコンテンンが出力されます。 selection/byrelevance が得られたAll とりテキストが表示され、team/member 子孫ノードのfirst、last、email 内にあるテキストがド キュメト内の順序により出力されます。member のid 属性は出力されていないではご注目してください、属性はmember の子とは見なされ ません)。

selection は対してグローバルテンプレートを作成する

selection のエンテンツを出力に含める必要は無いため、selection に対して空のグロー・シリテンプレートを作成し、そのエンテンツが処理され ないようえます。以下の操作を行なってくたさい

- 1. デザインビューが表示されている状態で、スキーマソノーサイドバー内にある全てのグローン い要素のselection を右 クリックします。
- 2. コンテキストメニューから、グローバルテンプレートの作成/削除を選択します。selection に対するグロー バルテンプレートが作成されます(以下のスクリーンショットを参照)。



- 3. グローバルテンプレートにてコンテンツプレースホルダをクリックし、キーボードの Delete キーを押下しま
- す。コンテンツプレースホルダが削除されます。 4. HTML プレビューをチェックします。All というテキストが出力されたテキストに表示されなくなります(以下のスクリーンショットを参照)。

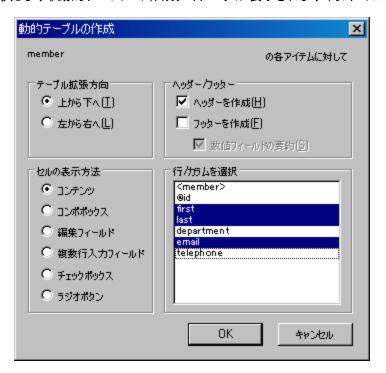
AndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.e

selection のグローバレテンプレートが空にていっているため、selection の子要素が処理されなくなります。

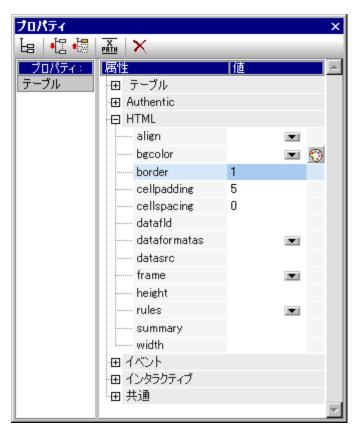
team/member /式けしてグローバルテンプレートを作成する

ここでメンバーの詳細情報を表示するテーブルを作成します。team 要素に対するグローバルテンプレート内にテーブルを作成します。以下の 操作を行なってくたさい

- 1. team 要素に対するグローバルテンプレートを作成します(スキーマツリーサイドバーにある全てのグローバル要 素にて team を右クリックし、グローバルテンプレートの作成/削除を選択します)。
 2. 全てのグローバル要素リストにて、team 要素を展開して、member 子要素を(デザイン内の)team グローバルテ
- ンプレートヘドラッグします。
- 3. team のグローバルテンプレートヘドロップした際にコンテキストメニューが表示されるので、テーブルの作成を 選択します。動的テーブルの作成ダイアログが表示されます(以下のスクリーンショットを参照)。



- 4. 属性/要素のリストにて、@id、department、telephone の選択を外し、「OK」をクリックします。動的テーブルが作 成されます。
- 5. カーソルをデーブルボディーのセルに配置して、<u>プロディサイドバーのテーブルが選択されている状態で、以下のスクト</u>シ ショナに示されているようにテーブルのプログラを変更します。



6. その他の値を、プロパティならびにスタイルサイドバーにて指定します。例えば、ヘッダー行へカーソルを配置した状態で、スタイルの trow を選択し、bgcolor プロパティの値を指定します。静的テキストの文字列であるヘッダーを編集することもできます。team 要素のコンテンツプレースホルダが表示されている場合、コンテンツ部分を削除してください。

HTML プレビューでは、テーブルが以下のように表示されます:

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j. ashe@nanonull.com

ロリンク

- 次のセクション. 払ま!
- 前のセグション: 条件を使用する
- チュートリアルスタートページ
- 出力の構造

3.8 おしまい!

おめでとうございます!これでチュートリアルは終了です。以下にあるような、SPS を作成する際に最も重要な内容を学習したことづいます:

- ドキュメントの<u>構造を作成</u>する方法、(メインのテンプレート とグローバルテンプレート)。
- 動的と静的 SPS ユンポーネントを使用して、動的ならびご静的なコンテンンをデザイン内部へ挿入する方法。
- 外部スタイルシート、グロー・ゾレスタイルレール、とローカルスタイルレールにて、CSS スタイルを使用する方法。
- 自動計算を使用して、XML データ内部の情報を取得する方法。
- <u>条件</u> を使用して、XML データのフィルタレングや、XML から得られたデータン従って出力を得る方法。
- <u>グロー ゾリテンプレート</u> と <u>その他のエンテンツ</u>を使用する方法。

これら機能の詳細な説明については、以下にあるセグションを参照くたさい

- SPS ファイル コンテンツ
- SPS ファイル 構造
- SPS ファイル 高度な機能
- SPS ファイル: プレゼンテーション
- SPS ファイル 追加機能

上記のセクションでは、このチュートリアルでは触れられなかったその他の StyleVision 機能についても解説されます。

- リンク

- 次のセグション: コンテンソのフォーマナ
- 前のセクション. 動的なコンテンツの挿入
- <u>チュートリアルスタートページ</u>

使用方法の概要 91

4 使用方法の概要

目的

StyleVision で作成された SPS ドキュメトを使用することで、HTML に対応した XSLT スタイルシートを生成することができます。SPS から生成されたスタイルシートを使用することで、SPS のそれと同じスキーマをベース した任意の XML ドキュメートを変換することができます。

SPS 作成のステップ

SPS を作成するためのステップ概要を以下に示します:

- 1. 新たに作成された空の SPS に対してスキーで割り当てます。スキーマ出ま(i)スキーマファイル(DTD まけまXMLスキーマ)、(ii) DB から生成された XML スキーマ(Enterprise ならびに Professional Edition のみ)、(iii) XBRL タケノノミをベース ユナスキーマ(Enterprise Edition のみ)、(iv) ユーザー定義スキーマ(StyleVision にて直接作成)の、ずれかけなります。この操作はデザイン概要サイド・ドーにて行うことができます。まけま、ファイル | 新規作成コマンドにより、スキーマから直接新規 SPS を作成することもできます。
- 2. SPS に作業用 XML ファイルを割り当てます。作業用 XML ファイル により、 出力プレビューを生成する時に SPS にて処理 される XML データが与えられます。作業用 XML ファイル がデザイン概要サイドバー にて割り当てられ、新たなスキーマイン対する作業用 XML ファイルにより出力を StyleVision でプレビューすることができます。
- 3. 使用するXSLT の デンシンを選択します。
- 4. Internet Explorer の互換性 を選択して、インストールされている Internet Explorer の デュンニ合致するよう します。
- 5. デザインビュー では、様々なデザインコンポーネトにより SPS ドキュメントをデザインすることができます。 デザインの作業 はドキュメント 構造の作成と、表示プロペティの定義がら構成されます。
- 6. 出力のテストを行います。デザインへの修正が必要とされる場合、修正を行った後に SPS を再度テストします。
- 7. XSLT ファイルや 出力ファイル か必要とされる場合、 それらのファイルを 生成します。

□ 関連項目

SPS ファイル構造

92 使用方法の概要 SPS とソース

4.1 SPS とソース

新規 SPS ファイルを作成する

SPSドキュメトを新たに作成するコよ「ファイル」新規作成(Ctrl+N)」コマイ以下のオプシュンを選択するか、標準ツール・一にある新規デザインアイコンをファインを対します。新規SPSドキュメトが作成され、デザインエーに表示されます。新規ドキュメトには、SPSX.sps とう仮の名前が与えられ、X にはアプケーションが開始されてから作成された新規ドキュメトの順番に応じた整数が与えられます。

新規 SPS ドキュメトを作成した後にはSPS のノースファイルを割り当てる必要が別ます。

SPS (対してノースファイルを割り当てる

SPS に対して割り当てることのできるソースファイルおり下の2 種類あります:

- スキーマノース
- 作業用 XML ファイル

これらソースファイルの割り当ては、デザイン概要サイドバーにて行うことができます。割り当てを行う方法については、デザイン概要のセグションで記述されます。各種ソースファイルに関する重要な事柄を以下に示します。

スキーマノース

デザイドキュメトの構造を作成するコよ SPS へ少なくとも1つのスキーマノースファイルを割り当てる必要があります。スキーマノースよ デザイン概要サイドバー 内で割り当てられます。スキーマオよ XML スキーマファイル(.xsd ファイル)、XML ファイルは少生成された XML スキーマ、DTD、まけよ、ユーザー定義スキーマであることができます。作業用 XML ファイル を割り当てることができます。

メモ SPS や、SPS から生成された XSLT スタイルシートへ名前空間を追加したい場合、SPS がベースとしている XML スキーマ内にあるトップレベルの schema 要素へ名前空間を追加する必要があります。

作業用 XML ファイル

は 作業用 XML ファイルを割り当てることができます。 作業用 XML ファイル から StyleVision 内部の出力プレビューで使用される XML データソースが与えられるため、割り当てられているスキーマに対して妥当である必要があります。 作業用 XML ファイルは デザル概要サイバーにで割り当てられます。

□ 関連項目

スキーマソノー

使用方法の概要 デザインの作成 93

4.2 デザインの作成

SPS では以下のような事柄を指定することができます:

1. <u>コンテンツ</u> (XML ドキュメントまたは DB から得られた) のうち、どれが出力されるか。 更に、 コンテンツを直接 SPS へ挿入することで、出力に含めることができます。

- 2. 出力の構造がどのようご行われるか、および
- 3. 出力の様々ない ーソス対して適用される表示 (フォーマナ) プロンティの内容 です。

出力のコンテンツ

出力のコンテンソは以下のアイテムから得られます:

- 1. SPS が適用される XML ドキュメト スキーマソレーサイドバー内の関連するスキーマソレーから目的の XML データノードをドラックし、SPS 内の目的の場所にドロップすることで、 XML ドキュメト から得られたコンテンツが SPS に含まれます。
- 2. アプリケーションからアクセスすることのできる、拡張された(つまり、StyleVision 自動計算内にある XPath 2.0 の doc() 関数を使用することで、外部 XML ドキュメントソースから得られたコンテンツへアクセスすることができます。XPath 条件式内にある doc() 関数によりアクセスされた XML ドキュメントが、スキーマノースの関連付けにより参照されている必要はありません。
- 3. SPS そのもの。テキストならびに(イメージやテーブルといった)その他のコンテンツは、キーボードや GUI に備わっている機能を使用することで、SPS 内に直接挿入することができます。これら入力は XML ドキュメントとは独立したものです。.
- 4. XPath 条件式により処理された動的(XML ソースの)データ。処理は通常 自動計算により行わます。
- 5. HTML 出力に対して、コンテンツ生成のために JavaScript 関数 を使用することができます。

出力の構造

SPS デザインでは、以下の方法により 出力の構造 を指定することができます: (i) XML ドキュメトの構造とは独立した 入力レベルテンプレート (Style Vision のメインのテンプレート) にて出力構造を指定、(ii) (Style Vision のグロー・ ジレテンプレート 等の) 様々なノードに対してテンプレートルールを宣言し (iii) これら2つのアプローチを組み合わせたもの。 デザインビューでは メインのテンプレート とグロー・ ジレテンプレート を使用することで、出力ドキュメトに対して使用する目的の構造を取得することができます モジュール SPS とデザインアグメトを使用することで、SPS に更なる柔軟性が追加されます。

出力の表示(フォーマル)

デザインビューでは CSS スタイルにより表示プロ ティケデザインエンポーネントへ適用されます。スタイルの定義はエンポーネントに対してローカルレベルで行うことができるまか、ドキュメントレベルで宣言された HTML セレクターや、外部 CSS スタイルシートにて宣言された HTML セレクターに対して行うことができます。 更に 定義済みのフォーマント を使用することで、特定の HTML 要素をエンポーネントに対して適用することができます。 表示プロ ティの指定に関する詳細については、 表示の方法のセグションを参照ください。

□ 関連項目

- SPS ファイル構造
- デザインビュー

4.3 XSLT と XPath のバージョン

SPS とは XSLT スタイルシートと 目様の機能を持っています。各 SPS に対して 1.0、2.0、まだは 3.0 の XSLT バージョンを指定する 必要があり、適切なツール デーアイコンの まだは まだって まだって きだい まだって とで選択を行うことができます。この選択により、2つのことが決定されます:

- StyleVision に搭載されているどの XSLT エンジンが変換に使用されるのか。StyleVision はXSLT 1.0、XSLT 2.0、とXSLT 3.0 という異なるエンジンが搭載されています。
- どの XSLT (1.0、2.0、封は3.0) の機能がインターフェース上に表示され、SPS で使用できるようになるのか。例えば、XSLT 3.0 ではXPath 3.0 が使用されるが、(XSLT 1.0 で使用される)XPath 1.0 封は (XSLT 2.0 で使用される)XPath 2.0 に比べてより強力な機能を使用することができます。更に、目次機能といった SPS の機能は XSLT 2.0 とXSLT 3.0 でのみ利用することができます。

XSLT 変換

StyleVision におおるXSLT 変換は (i) インターフェースに <u>出力ビュー</u>を生成するか、そして(ii) <u>インターフェイス内部</u>と <u>StyleVision Server</u> からHTML) の<u>出力ファイルの生成と保存</u>を行うかめに使用されます。変換に使用されるXSLT エンジン(Altova XSLT 1.0、2.0、まけは3.0 エンジン) のドージョンは、SPS で選択されたのドージョンに以決定されます。

GUI における XSLT の機能

XSLT の各バージョンで利用することのできる機能の大半は、対応する XPath のドージョンに関連しています (XSLT 1.0 のために XPath 1.0、XSLT 2.0 のために XPath 2.0、XSLT 3.0 のために XPath 3.0 が使用されます)。 XPath 条件式は Style Vision 内の様々な場所 -特に 自動計算 と 条件テンプレート - で使用されており、インターフェースに備えられている機能を、XPath 条件式の構築に役立てることができます。 選択された XSLT のドージョンに従うかたちで、正しい XPath の機能をインターフェースで自動的に利用することができるようはいます。

- □ 関連項目
- 生成されたファイル

4.4 Internet Explorer の互換性

SPS デザインを(デザインビュー) 正確に表示し、プレビューを(HTML プレビュー) 出力するコよ StyleVision をお使い クエンピューター 上にInternet Explorer (IE) をインストーリする必要があます。 サポートされている IE の デジョンを以下に示します:

- Internet Explorer 5.5 おさ以降。
- Internet Explorer 6.0 おび以上では XML のサポートが改善されているため推奨されます。
- Internet Explorer 9 (IE9) 以上では、サポートされるイメージフォーマットの拡張や、新たな CSS スタイルな ど、更に機能が追加されています。デザインにてこれらの新たな機能を使用する場合、IE9 の使用を検討したほうが良いでしょう。

IE9 の機能-Style Vision におけるサポート

以下にあるIE9 以降の機能がStyleVision でサポートされます:

- 新たなイメージフォーマットのサポート: TIFF、JPEG、XR、SVG(SVGドキュメントは XML 形式で記述されており、SVG 名前空間内にある必要があります)。これらのイメージフォーマットは IE9 では表示されるものの、それより古いバージョンの IE では表示されません。出力にてサポートされているイメージのリストについては、イメージ型と出力を参照ください。
- 以下にある、(CSS3 を含む)新たな CSS スタイルのサポート。これらスタイルは、HTML 出力は対してのみ適用されます。
 - background-clip
 - background-origin
 - background-size
 - box-sizing
 - box-shadow
 - border-radius (border-*-radius)
 - font-stretch
 - ruby-align
 - ruby-overhang
 - ruby-position
 - overflow-x, overflow-y
 - outline (outline-color, outline-style, outline-width)
 - text-align-last (partial)
 - text-overflow (partial)
- 新たに追加された CSS の長さに関する関数: calc()。
- 新たに追加された CSS の色に関する関数: rgba()、hsl()、hsla()。
- 新たに追加された CSS の長さユニット: rem、vw、vm、vh、ch。
- IE9 にてサポートされる HTML5 要素を、ユーザー定義要素としてデザインへ挿入することができます。

デザインビューとIE バージョン

プログライダイアログにてIE のバージョンを指定することで、特定のIE バージョンに互換性のあるかたちでデザインビューを使用することができます。以下の影響が発生します。

• 選択された IE のバージョンでレンダリングすることができる CSS スタイルは全て、StyleVision のスタイル サイドバーに表示されます(IE9 を選択して、サポートされている CSS スタイルとデザインインターフェース で使用する場合、IE9 がインストールされている必要があります)。 例えば、IE9 がインストールされている 状態で IE9 が互換バージョンとして選択されている場合、IE9 でサポートされている CSS3 をデザインイン ターフェース内で使用することができます。 • 選択された IE バージョンに対応する HTML 要素は、<u>定義済みのフォーマナ</u> まだは ユーザー定義要素 として入力 することができます。HTML 要素は

Internet Explorer の互換性

使用方法の概要 生成されたファイル 97

4.5 生成されたファイル

StyleVision では、「ファイル 生成されたファイルの保存」コマドまけは StyleVision Server を使用し、XSLT スタイルシート と出力ファイルを生成することができます。

StyleVision からお以下のファイルを生成することができます:

- SPS デザインをベースにした XSLT スタイルシート。のために生成されます。
- 作業用 XML ファイル を処理して生成されるファイルは、SPS から生成された XSLT スタイルシート を使用して SPS 内で割り当てられます。

出力のマーケアップはSPS 内部に含まれています。出力に使用されるデータはXMLドキュメント 内部に収められています。マーケアップとデータは XSLT スタイルシートにより結び付けられます。StyleVision の出力ビューでは XSLT スタイルシートと実際の出力の両方をプレビューすることができます。

メモ SPS や、SPS から生成された XSLT スタイルシートに名前空間を追加する場合、SPS がベースとしている XML スキーマのトップレベルにある schema 要素にて名前空間を追加する必要があります。

Altova W eb サイト: 🔗 XML レポート

生成されたギャンメトに関する重要なポイントを以下に示します。

HTML 出力とスタイルシーナ: (1) 生成されたHTMLドキュメトのフォーマトとレイアナは、StyleVision 内部で確認することのできるほぼ同一のものになります。(2) HTML ファイル内のデータ入力デッイス(テキスト入力フィー)ドやチェックボックスなど)では入力が受け付けられることはありません。これらデータ入力デッイスは、Authentic View にて XML データの入力を行うためのもので、HTML フォーマットでも同じように表示されはするものの、データ入力のために使用することはできません。

■ 関連項目

ファイルメニュー 生成されたファイルの保存

4.6 StyleVision におけるカタログ

StyleVision では、OASIS XML カタログメカニズムのサブセトがサポートされます。カタログ機能により、一般的に使われているスキーマ(ヤスタイルシートなどのファイル)が、StyleVision によりローカルのユーザーフォルダーから取得されます。これにより全体的な処理スピードが向上し、(ネトワーグは接続していない状態でも)ユーザーはオフラインで作業することができるようになり、(URI の変更はカタログファイル内部に限定されるので)ドキュメントのポータビリティーが向上します

StyleVision におおカタログ機能は以下のように動作します:

RootCatalog.xml

StyleVision が起動すると、検索を行うカタログファイルのリストが含まれる RootCatalog.xml とり名前のファイルがロードされます。このファイルを変更し、nextCatalog 要素内にファイルを加えることで、検索を行うファイルを増やすことができます。各カタログファイルが検索され、カタログファイルこで指定されたマンピングによりURI が解決されます。

上記の表示にて、%AltovaCommonFolder%変数に共識別されたフォルダー以下のSchemas ならびにXBRL フォルダー内に、catalog.xml とら名前のカタログファイルが存在する点に注目してくたさい、%AltovaCommonFolder%変数の値は以下のテーブルにて与えられます)。

Altova Common フォルダー内にあるカタログファイルにより、(SVG やW SDL といす)広く使用されるスキーマや XBRL タケノノミの公開ならびにシステム識別子から、ローカルに保存された対応するスキーマのコピーが指し示されます。これらスキーマは、Style Vision のインストール時に Altova Common フォルダーへインストールされます。エラーを起こす可能性があるため、これらファイル内にあるマッピングを複製しないように注意する必要があります。

CoreCatalog.xml、CustomCatalog.xml、とCatalog.xml

上に示される RootCatalog.xml /こて、CoreCatalog.xml ならいこCustomCatalog.xml が表示されている点に注目してください

- CoreCatalog.xml には、Altova Common フォルダー内にあるスキーマを参照するための Altova 特有のマッピングが含まれます。
- CustomCatalog.xml は、独自のマッピングを作成するためのスケルトンファイルです。CustomCatalog.xml には、任意のスキーマに対するマッピングを追加することができますが、Altova Common フォルダー内のカタログファイルで記述されていないものに限られます。OASIS カタログ機能でサポートされている要素を使用するようにしてください(下記参照)。
- Altova Common フォルダーには多くの Catalog.xml ファイルが含まれています。各ファイルは、Altova Common フォルダー内にあるスキーマや XBRL タクソノミのフォルダー内部に収められており、それぞれ公開ならびにシステム識別子により、ローカル上に保存された対応するスキーマのコピーが参照されます。

スキーマカタログファイルとスキーマの場所

RootCatalog.xml ならびこCoreCatalog.xml ファイルは StyleVision アプケーションフォルダーへインストールされます。
CustomCatalog.xml ファイルは MyDocuments/Altova/StyleVision フォルダー以下に配置されます。各 catalog.xml ファイルは対応するスキーマフォルダー内に収められており、これらスキーマフォルダーは、 %AltovaCommonFolder% ¥Schemas ならびに %AltovaCommonFolder% ¥XBRL 以下に配置されます。

シェル環境変数 とAltova 変数

シェル環境変数をnextCatalog 要素内で使用することには、様々なシステム上の場所を指定することができます(上記のRootCatalog.xml を参照)。以下のシェル環境変数がサポートされます:

%AltovaCommonFolder C:\Program Files\Altova\Common2021

%DesktopFolder% 現在のユーザーのデスクトップフォルダーへのフル ス

%ProgramMenuFolder% 現在のユーザーのプログラムメニューフォルダーへのフル ス

%StartMenuFolder% 現在のユーザーのスタートメニューフォルダーへのフル &

%StartUpFolder% 現在のユーザーのスタートアップフォルダーへのフル ス

%TemplateFolder% 現在のユーザーのテンプレートフォルダーへのフル 次

%AdminToolsFolder% 現在のユーザーの管理ソールを保管するファイルシステムフォルダーへのフル 次

%AppDataFolder% 現在のユーザーのアプリケーションデータフォルダーへのフル 次

% 全てのユーザーのアプリケーションデータを含むファイルフォルダーへのフル 🎗

CommonAppDataFolde

r%

%FavoritesFolder% 現在のユーザーのお気に入りフォルダーへのフル ス

%PersonalFolder% 現在のユーザーの個人用フォルダーへのフル ス

%SendToFolder% 現在のユーザーのSendTo フォルダーへのフル 🎗

%FontsFolder% システムフォントフォルダーへのフル 穴

%ProgramFilesFolder% 現在のユーザーのProgram Files フォルダーへのフル ス

%CommonFilesFolder% 現在のユーザーのCommon Files フォルダーへのフル 穴

%WindowsFolder% 現在のユーザーのWindows フォルダーへのフル 🎗

%SystemFolder% 現在のユーザーのSystem フォルダーへのフル ス

%LocalAppDataFolder% ローカル(非ローミング)アプリケーションコマオするデータリポトリとして機能するシステムフォルダーへのフルタ

%MyPicturesFolder% ピクチャライブラノ(マイピクチャ)フォルダーへのフルス

カタログの動作方法: DTD

カタログにより、DTD への呼び出しは通常ローカルURI ペルダイレクトされます。カタログファイルにより、公開ませるシステム識別子から対応するローカルURI へのマッピングが行われます。XML ファイル内のDOCTYPE 宣言が読み込まれた場合、公開ませるシステム識別子が、カタログファイルによりローカルリソースへマッピングされます。

広く使用されているスキーマの場合、PUBLIC 識別子が通常既に定義されており、カタログファイル内のURI を適切なローカルコピーヘマッピングすることさけい必要にないます。XML ドキュメントが解析(パース)されると、その中にある PUBLIC 識別子が読み取られます。カタログファイル内に目的の識別子がある場合、カタログファイル内の対応する URL がロードされ、その場所からスキーマが読み取られます。例えば、以下にある SVG ファイルを考えてみましょう

このドキュメトか読み取られると、そのPUBLIC 識別子に対するカタログが検索されます。カタログファイルコお以下のエナリーか含まれます.

この場合、PUBLIC 識別子に対応するマッチが存在するかめ、SVG DTD へのルックアップがschemas/svg/svg11.dtd URI ヘルダイレケト され(この ぬけがログファイルに対して相対的となります)、参照されたローカルファイルがDTD として使用されます。カタログ内のPublic ID に対するマッピングが存在した。場合、XMLドキュメト内のURL(上記の例の場合、http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd)が使用されます。

カタログの動作方法: スキーマ

StyleVision では、カタログによりXML スキーマのリダイレクトを行うことができます。XML インスタンスファイルでは、以下の例にあるように、XML ドキュメントのトップレベルドキュメント要素内にある xsi:schemaLocation 属性にスキーマへの参照が行われます。例えば、

xsi:schemaLocation="http://www.altova.com/schema/orgchart OrgChart.xsd"

通常、(上の例にて太字で示されている)属性値のURI はま、実際に存在するスキーマの位置が記述されます。カタログによりスキーマが参照された場合、実際に存在するXML スキーマを指し示す必要は無いものの、xsi:schemaLocation 属性の妥当性を満たすために何らかの値を入力する必要があります。属性のURI 部分はよ、例えばhoge などの値を入力することでも、妥当性を確保することができます。xsi:schemaLocation 属性値の名前空間部分(上の例におけるhttp://www.altova.com/schemas/orgchart)により、カタログでは、以下のエトリーにより名前空間部分をベースにしたスキーマの識別が行われます。

<uri name="http://www.altova.com/schemas/orgchart" uri="C:\MySchemas\OrgChart.xsd"/>

によりサポートされるカタログサブセット

CustomCatalog.xml(又はStyleVision)にて読み取ることができるその他のカタログファイル)にてエナリーを作成する場合、以下にある OASIS カタログ仕様の要素だけを使用するようにしてください。各要素と、その属性値に関する説明を以下に示します。より詳細な解説に

ついては、XML カタログの仕様を参照くたさい。各要素にはxml:base 属性を含めることができ、その要素のベースURI を指定するのに使 用することができます。

- <public publicId="PublicID of Resource" uri="URL of local file"/>
- <system systemId="SystemID of Resource" uri="URL of local file"/>
- <uri name="filename" uri="URL of file identified by filename"/>
- <rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String to replace StartString"/>
- <rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement string to locate resource locally"/>

殆どのスタイルシートに見られるように公開識別子が存在してい場合、system 要素によりシステム識別子をURL へ直接マメピングすること ができます。更にuri 要素を使用することで、URI を他のURI ヘマソピングすることもできます。rewriteURI ならびにrewritsSystem 要素に より、それぞれURI やンステム識別子の書き換えを行うことが可能になります。この機能により、使用されるファイル ひを変更し、ターゲトの フォルダーを変更することが可能にないます。これら要素に関するより詳しい情報については、XML カタログの仕様を参照くたさい。

詳しい説明

カタログに関する詳細は、次を参照してくたさい XML カタログの仕様。

- □ 関連項目
- 生成されたファイル

102 使用方法の概要 StyleVision 内のカタログ

4.7 StyleVision 内のカタログ

The XML カタログ構造によりローカルフォルダーからファイルを抽出することが可能しまり、一変更のためにカタログファイルURI のみか必要してよるため、に処理スピード全体が向上し、ドキュメントのポータビリティも向上します。詳細に関しては、カタログの機能のセクションを参照してください。

Altova のXML 製品はDTD とXML スキーマなど共通に使用されるファイルに素早くアクセスレロードするかとファイットでは使用します。このカタログ構造はカスタム化することが可能で、またユーザーにお拡張可能です。セグション Style Vision 内のカタログ構造 と力タログのカスタム化内で説明されています。セグション Windows システムロケーションのナックのカックを数 は共通のシステムのロケーションのナックに Windows 変数をリストしています。共通で使用されているフォルダーをロケートするからについて変数はカタログファイル内で使用することができます。

このセクションは次のサブセクションに整理されています。

- カタログの仕組み
- StyleVision 内のカタログ構造
- カタログのカスタム化
- Windows システムロケーションのオーめの変数

これらのカタログの詳細に関してはXML カタログ仕様)を参照してくたさい。

4.7.1 カタログの仕組み

カタログをDTD とXML スキーマをりダイレクトするために使用することができます。両方のケースのメカニズムの後ろにあるコンセプトは同じですが、違いの詳細は以下に説明されている通りです。

DTD

カタログはDTD への呼び出しをローカルのURI にダイレクトするために使用されます。これはマメピング、カタログファイル、ローカルURI により必要とされる、グラックませっぱンステム識別子により達成されます。XML ファイル内のDOCTYPE 宣言が読み取れれると、ノグラックませっぱンステム識別子が必要とされるローカルリノースをカタログファイルマッピングによりロケートします。

人気のあるスキーマニ関しては PUBLIC 識別子は通常定義済みです。このためカタログファイル内のURI のみが正し、ローカルコピーコマリして PUBLIC 識別子 をマップします。XML ドキュメトカ解析されると内部の PUBLIC 識別子が読み取られます。この識別子がサタログファイル内で見つかると、カタログファイル内の対応する URL が シックアップされ、スキーマかこのロケーションから読み取られます。 例えば、次の SVG ファイルが Style Vision 内で開かれるとします:

カタログはこのSVG ファイルのPUBLIC 識別子を検索します。カタログファイルは以下のエトリを含んでいます:

<catalog>

```
cpublic publicId="-//W3C//DTD SVG 1.1//EN" uri="schemas/svg/svg11.dtd"/>
...
</catalog>
```

この場合 PUBLIC 識別子のためこ一致するものがおけます。この結果 SVG DTD のためのルクアップが、カタログファイルは相対的な) URL schemas/svg/svg11.dtd に ダイレクトされます。これは SVG ファイルのためこ DTD として使用されるローカルファイルです。 カタログ内で Public ID のためのマッピングが存在したは場合、 XML ドキュメト内の URL が使用されます(上の SVG ファイルのサン プルの場合、これはインターネト URL です: http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd)。

XML スキーマ

StyleVision 内では XML スキーマとカタログを使用することができます。XML インスタンスファイル内で、スキーマへのレファレンスがXML ドキュメントのトップレベルの要素のxsi:schemaLocation 属性内で発生します。例えば、

```
xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"
```

xsi:schemaLocation 属性の値にはつの部分が決定す:(上で緑色で表示されている)名前空間のパトとハイライトされている) URI パート。名前空間の部分は代替リンースをマップするかと対しが内で使用されています。例えば、以下のかタログエントリは上のスキーマレファレンスを代替のロケーションのスキーマログイレクトします。

```
<uri name="http://www.xmlspy.com/schemas/orgchart" uri="C:\MySchemas\OrgChart.xsd"/>
```

通常 xsi:schemaLocation 属性の値のURI 部分は実際のスキーマのロケーションの、父です。しかしなから、スキーマかかタログを介して参照される場合、URI の部分は実際のXML スキーマをポイントする必要はなく、xsi:schemaLocation 属性の文法の妥当性が保持されるために存在する必要があます。Foo の値は、例えば、属性の値の部分が妥当であるため十分です。

4.7.2 StyleVision 内のカタログ構造

StyleVision が開始されると、(以下のリスティングで示されるような構造を持った)RootCatalog.xml とうファイルがロードされ、ファイル内部に記述されているカタログファイルが検索されます。このファイルを修正して、検索するカタログファイルを増やすこともできます(各カタログファイルはnextCatalog 要素にて記述されます)。各カタログファイル内に記述されているURIが、ファイル内で指定されたマンピングに従って解決されます。

Listing of RootCatalog.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog"
    xmlns:spy="http://www.altova.com/catalog_ext"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:entity:xmlns:xml:catalog Catalog.xsd">
    <nextCatalog catalog="%PersonalFolder%/Altova/%AppAndVersionName%/CustomCatalog.xml"/>
    <!-- Include all catalogs under common schemas folder on the first directory level -->
    <nextCatalog spy:recurseFrom="%CommonSchemasFolder%" catalog="catalog.xml"
spy:depth="1"/>
    <nextCatalog spy:recurseFrom="%ApplicationWritableDataFolder%/pkgs/.cache"
catalog="remapping.xml" spy:depth="0"/>
    <nextCatalog catalog="CoreCatalog.xml"/>
    </catalog>
```

上のリスティングはカスタムカタログ(named CustomCatalog.xml)と(W3C XML スキーマとSVG スキーマ)などの一般的に使用されるスキーマをロケートするカタログのセットを参照します

● CustomCatalog.xml は(変数 %PersonalFolder% によりロケートされる)個人用フォルダーにあります。自身のマンピ

104 使用方法の概要 Style Vision 内のカタログ

ングを作成するスケルトンファイルです。共通のスキーマフォルダー内のカタログファイルこと、指されているい必要とするスキーマのため にマッピングを CustomCatalog.xml に追加することができます。OASIS カタログメカニズムのサポートされる要素を使用してこれを行います(次のセグションを参照)。

- (共通ご使用されるスキーマのセナを含む変数 & Common Schemas Folder & を使用してロケートされる) 共通のスキーマフォ ルダーこれらのスキーマフォルダの内部は保存された対応するスキーマのユピーをポイントする URI に対する ブリックおよびまけるシステム識別子をマップする catalog.xml ファイルです。
- CoreCatalog.xml はStyleVision アプケーションフォルダー内にあり、XMLドキュメトのAltova のAuthentic ビューを生成するが、対け、使用される StyleVision パフースタイルシートなど StyleVision 一固有のプロセスにより使用されるスキーマとスタイルシートをロケートするが、使用されます。

ロケーション変数

RootCatalog.xml 内で使用されている変数は以下の値を持っています、上のリスティング:

%PersonalFolder%	現在のユーザーの個人的なフォルダー、例えばC:\Users\ <name>\Documents</name>
%CommonSchemasFolder%	C:\ProgramData\Altova\Common2021\Schemas
% ApplicationWritableDataFolde r%	C:\ProgramData\Altova

力タログファイハとスキーマのロケーション 多種のカタログファイルの場所に注意してくたさい。

- RootCatalog.xml とCoreCatalog.xml はStyleVision アプケーションフォルダー内にあます。
- CustomCatalog.xml はMyDocuments\Altova\StyleVision フれば、内にかます。
- Catalog.xml 入るはそれぞれ特定のスキーマフォルダー内にあり、これらのスキーマフォルダーは共通のスキーマフォルダー内にあります。

4.7.3 カタログのカスタム化

CustomCatalog.xml(おさまStyleVisionにお読み取られる他のカタログファイル)内でエトリを作成する場合OASIS カタログ 仕様の以下の要素のみを使用してくささい。下の各要素は属性の値の説明と共にリストされています。詳細に関してはXML カタログ仕様)を参照してくささい。各要素はその要素のベースURI を指定するために使用されるxml:base 属性を取ることができます。

- public publicId="PublicID of Resource" uri="URL of local file"/>
- <system systemId="SystemID of Resource" uri="URL of local file"/>
- <uri name="filename" uri="URL of file identified by filename"/>
- <rewriteURI uriStartString="StartString of URI to rewrite" rewritePrefix="String
 to replace StartString"/>
- <rewriteSystem systemIdStartString="StartString of SystemID" rewritePrefix="Replacement string to locate resource locally"/>

以下の点に注意してくたさい

- 多くのスタイルシート同様/ ヴァクが識別子が存在しない場合、システム識別子をsystem 要素を介して直接 URL にマップすることができます
- uri 要素を使用してURI を他のURI にマップすることができます。
- rewriteURI とrewriteSystem 要素によりURI おはよステム識別子の開始の部分をそれぞれ書き換えることが可能になります。これによりファイル 次の開始部分が置き換えられこの結果他のディノケリをターゲットにすることができます。これらの要素の詳細に関してはXML カタログ仕様を参照してくたさい。

リリース2014 以降StyleVision はXML Catalogs specification (OASIS Standard V1.1, 7 October 2005) 仕様を厳密 に順守しています。この仕様は厳密にいでかりID まけまシステムID) 外部識別子ルグアップを(ハヴックID まけまシステムID を持たさいURI) URI ルックアップと区別しています。名前空間 URI は単純いで ヴックID まけまシステムID ではなく URI と考えられ外部識別子ルックアップではなく URI ルックアップとして使用されるべきです。バージョン2014 以前のStyleVision バージョンではスキーマ名前空間 URI は < public マッピングを使用して訳されてきました。2014 以降から < uri > マッピングが使用されています。

```
v2014 以前: <public publicID="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>
V-2014 以降: <uri name="http://www.MyMapping.com/ref" uri="file:///C:/MyDocs/Catalog/test.xsd"/>
```

StyleVision が参照されたスキーマを検索する方法

xsi:scemaLocation 属性を介してXMLドキュメト内で参照されるスキーマ(以下に表示されて、ます)。
xsi:schemaLocation 属性の値は2つの部分が決ます、(緑色で表示されている)名前空間のパトとハイライトされている) URIパト。

xsi:schemaLocation="http://www.xmlspy.com/schemas/orgchart OrgChart.xsd"

以下はStyleVision が参照されナースキーマを検索するステップです。スキーマオ最初のステップでロードされます。

- 1. xsi:schemaLocation 値のURI 部分のためのカタログをルックアップします。rewriteURI マッピング内を含みマッピングが検索されるとスキーマのロードのために結果のURI を使用します。
- 2. xsi:schemaLocation 値の名前空間部分のためのカタログをルックアップします。rewriteURI マンピング内を含みマンピングト検索されるとスキーマのロードのために結果のURI を使用します。
- 3. スキーマのロードのためにxsi:schemaLocation 値のURI 部分を使用します。

XML スキーマ仕様

XML スキーマ仕様情報は Style Vision 「モントされてお」 XML スキーマ(.xsd) ドキュメントの妥当性はこの内部情報に対してチェックされます。 XML スキーマドキュメント内では、このためレファレンスは XML スキーマ仕様を定義するスキーマフょって口ざいません。

%AltovaCommonSchemasFolder%\Schemas\schema フォルダー内のcatalog.xml ファイルは古いXML スキーマ仕様を 実装するDTD へのレファレンスを含んできます。XML スキーマドキュメントをこれらのスキーマス対して検証してはおりません。参照されるファ イルはドキュメントをこれらの古い 奨励に従い作成する場合に編集の目的で StyleVision に入力ヘッレ や情報を与えるセダに含まれています。

4.7.4 Windows システムロケーションのための変数

nextCatalog 要素でおプロ環境変数が使用され、システムコ関係する場所への、よか参照されます(上のRootCatalog.xmlの内容を参照)。以下のプロ環境変数がサポートされます:

```
%PersonalFolder% 現在のユーザーの個人的なフォルダーへのフル 久 例えばC:\Users\<name>\Documents
%
CommonSchemasFol
der% C:\ProgramData\Altova\Common2021\Schemas
%
ApplicationWrita
bleDataFolder% C:\ProgramData\Altova
```

106 使用方法の概要 Style Vision 内のカタログ

```
AltovaCommonFold
               C:\Program Files\Altova\Common2021
er%
              現在のユーザーのデスクトップフォルダーへのフルス。
%DesktopFolder%
ProgramMenuFolde
               現在のユーザーのプログラムメニューフォルダーへのフル・ス。
r%
%StartMenuFolder
               現在のユーザーの開始メニューフォルダーへのフル・ス。
              現在のユーザーのスタートフォルダーへのフルス。
%StartUpFolder%
%TemplateFolder% 現在のユーザーのテンプレートフォルダーへのフル 🎖。
AdminToolsFolder
               現在のユーザーの管理ソールを保管するファイルシステムディレクトリへのフルパ、
              現在のユーザーのアプリケーションデータフォルダーへのフル・ス。
%AppDataFolder%
CommonAppDataFol
               全てのユーザーのアプリケーションデータを含むファイルディレクトリへのフルペ。
der%
%FavoritesFolder
               現在のユーザーのお気に入りフォルダヘフルス。
%PersonalFolder% 現在のユーザーの個人フォルダーへのフル 次。
              現在のユーザーの送信フォルダーへのフル・ス。
%SendToFolder%
              システムフォントフォルダーへのフルパ。
%FontsFolder%
ProgramFilesFold
               現在のユーザーのプログラムファイルフォルダーへのフルペ。
er%
CommonFilesFolde
               現在のユーザーの共通ファイルフォルダーへのフルペ。
              現在のユーザーのWindows フォルダーへのフル 久。
%WindowsFolder%
              現在のユーザーのシステムフォルダーへのフルス。
%SystemFolder%
9
Local App Data Fold ローカル(非ローミング) アプリケーションのためのデータレポネリとしての役割を果たすファイルシステムディントリ
er%
               へのフルダ。
MyPicturesFolder
              ピクチャフォルダーへのフルス。
```

SPS コンテンツ 107

5 SPS コンテンツ

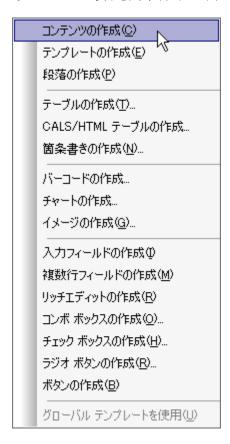
- <u>テキストとして XML コンテンルを挿入</u>:関連するノード(要素、属性、型、おけま CDATA)をドラッグすることで、デザイン内に XML データを挿入し、(コンテンツ) おけまくの他のコンテンツとして作成します。
- MS W ord コンテンソの挿入
- ユーザー定義デンプレート
- ユーザー定義要素、XML テキストブロック
- <u>テーブルの作業</u>: テーブルは(i) SPS デザイナーにより、(静的テーブル)やXML ドキュメトのサブ構造を使用することでSPS デザイン内に、ませは(ii) Authentic View ユーザーにより挿入することができます。
- <u>リストの作成</u>: SPS デザイン内にリスト構造が入力される静的リストや、XML ドキュメトのサブ構造がリストとして作成される動的リストにより、パフフルなデータ並び替え機能を使用することができます。
- <u>イメージの使用</u>:様々な方法を使用することで、ターゲル URI (静的、動的、これらの混合、パースされていないエンティティ URI)から画像を SPS デザイン・挿入することができます。
- <u>データ入力デ・イス(ませょフォームエトロールの使用</u>:入力フィールドヤエンポドックスと・ケセデータ入力デ・イスにより、 Authentic View にて XML データを入力することができます。この機能により、ユーザーへのヘッレプや入力の制約が提供されます。 XML ドキュメト内にある個々のソードをデータ入力デ・イスとして作成することができます。
- リンク
- バーコード
- レイアナ モジュール
- 変更機能: この機能を使用することで、テンプレートにマッチしたノードの選択を行い、別のコンテンツ型へノードを変更することが可能にないます。

□ 関連項目

- SPS ファイルの高度な機能
- SPS ファイル追加機能

5.1 テキストとして XML コンテンツを挿入

対応するスキーマノードをスキーマソノーウンドウからドラッグし、デザイノコでドロップすることで、XML ドキュメント内にあるノードから得られたデータを、デザイノコ挿入することができます。スキーマノードをデザイノコドロップすると、デザイノコでこのノードをどのように作成するのか選択するためのメニューが表示されます(以下のスクノーシンョナを参照)。



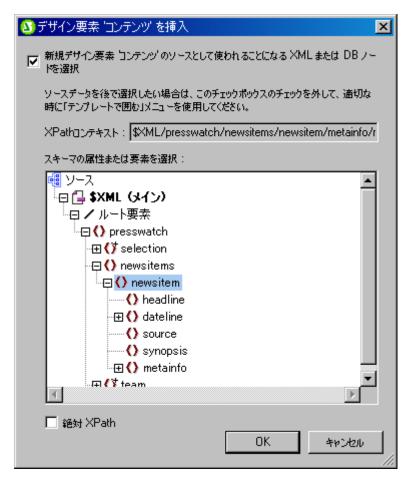
スキーマノードの種類

スキーマソノーサイドバーからデザインペドロップすることができるスキーマノードは(i) 要素ノード、(ii) 属性ノード、(iii) データ型ノードの、ずれかないます。

コンテンソの挿入ツール・アイコンを使用する

<u>デザイン要素の挿入 ツール ドー</u>アイコンを使用することでも、デザイン内にあるノードのコンテンンを挿入することができます。以下の方法によりコンテンクを挿入することができます:

- 1. コンテンソの挿入アイコンを選択します。
- 2. コンテンツを挿入するデザイン内の箇所をクリックします。デザイン要素コンテンツを挿入ダイアログが表示されます (*下のスクリーンショッ*ト)。



- 3. デザイン内における挿入箇所のコンテキストが XPath コンテキストフィールドにて表示されます。コンテン ツを作成するノードを選択します。
- 4. 「OK」をクリックします。コンテンツプレースホルダが作成されます。選択されたノードがコンテキストノード では無い場合、コンテンツプレースホルダの周りに、選択されたノードへのパスまでのテンプレートタグが 作成されます。

ノードのテキストコンテンンを出力する

ノードのテキストコンテンンを出力するコよノードをコンテンソとして作成します。ノードがコンテンソとして作成されると、デザインドキュメントにて ノードが以下のように表示されます:



上のスクレーシンョナでは Desc 要素がエンテンツとして作成されました。出力ではDesc のテキストコンテンツが表示されます。Desc 要素 にBold やItalic といた子要素が含まれる場合、子要素のテキストコンテンル、Desc の一部として出力されます。Desc 要素の属性ノー ドは子ノードとして見なされないがめ、属性ノードのエレテンソがDesc コレテンソの一部として出力されることはありません。属性内のエレテンソ を処理するコよ、属性ノードを明示的に挿入する必要があります。

CDATA セクション

CDATA セクションがXMLドキュメント内に存在する場合、それらが出力にないます。.

∱10 CData This is CDATA Text. € 10 CData

メモ Authentic View では、CDATA セクションを入力フィールド(テキストボックスならびに複数行のテキストボックス)へ挿入することはできません。Authentic View 内に表示される要素内部にテキストコンテンツコンポーネントとして挿入することはできます。

このセクション

このセクションのサブセクションでは XML コンテンンをテキストとして挿入するときに考慮するべき、その他の事柄について説明します:

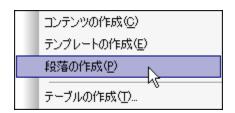
- ・ ノードが挿入された際に、ノードのテキストコンテンツを 定義済みのフォーマナによ直接マーケア・プする。
- ノード内に明示的に含まれていない子孫ノードを処理に含める次を参照してくださいその他のエレテンツ.

メモ ノードのコンテンツプレースホルダを削除することで、**空のテンプレートルール**を作成することができます。空のテンプレートルールを使用することで、処理されない(つまり表示させない)ノードを定義することができます。

- 関連項目
- デザインビューシンボル
- テンプレートとデザインフラグメント

5.1.1 定義済みのフォーマットとともにコンテンツを挿入する

ノードのテキストコンテンソは、StyleVision の定義済みのマーケアップとともに、直接挿入することができます。スキーマソノーウンドウから目的の場所にノードをドラッグアンドドロップして、表示されるメニューから段落の作成を選択します(以下のスクリーンショントを参照)。



定義済みのフォーマトタグを選択し、その他の定義済みフォーマトを<u>ツール・内のエンボボックスのフォーマト</u>から選択、ませはメニューコマイ・の「挿入 | フォーマット」を使用することで、定義済みのフォーマトを変更することができます。



プロ・ディウィドウ内にある段落グループの段落タイププロ・ディの値を変更、まけまと一ドテンプレートの<u>コンテキストメニューコマンド「タグ</u>で囲む | 特殊な段落」から段落タイプを変更することで、定義済みのフォーマットを変更することができます。

各種の段落には、それぞれ書式に関する高度な機能が含まれています。pre フォーマナ型では、改行が空白スペースに変換されること無く、そのまま表示されます。

- □ 関連項目
- デザインビューシンボル
- 定義済みのフォーマット

5.1.2 その他のコンテンツ

rest-of-contents (その他の) コンテンソプレースカレダは、作成されたテンプレートに対する要素以下にある。その他の子要素全てに 対して適用されます。以下の例を考えてみましょう:

- parent 要素にはchild1 からchild4 まで4つの子要素が含まれます。
- parent 要素に対するテンプレート内では ある処理がchild1 とchild4 子要素に対して明示的に定義されています。

この場合、child1 とchild4 だけが処理されます。child2 ならび child3 子要素が処理されることはありません。parent の要素に対してその 他のエンテンソプレースヤルダが挿入された場合、テンプレート内部で明示的に定義された処理ルールにより、child1 ならびに child4 だけでは なく、child2 ならびにchild3 子要素も処理の対象になります。これら子要素に対してグロー・ジリテンプレートが定義されている場合、グロー / ゾレテンプレートが使用されます。 そうでは 場合、 (要素、 属性、 テキストノード (文計して使用される) 内蔵のデフォルト テンプレートが適用さ れます。

重要なメモ その他のエレテンソニ対してどのソードが選択されているか留意する必要があります。

- 上の例で記述されたように、全ての子要素ノードならびに子テキストノードは、その他のエンテンソプレースカリダにより選択されます(XML) ドキュメト内にある不正な子ノードも処理されます)。
- 属性ノードは選択されません。これらは子ノードとして見なされず、XPath 軸にも含まれません。
- 子要素のグロー・ジリテンプレートが親テンプレートで使用される場合、子要素がローカルで使用されたとは見なされません。そのため、その 他のコンテンソプレースホルダにより子要素も選択されることづなります。しかし、子要素のグロー・バリテンプレートが「ローカルのコピー」となっ ている場合、ローカルで使用されたことしてい、子要素がその他のコンテンソにより選択されることはありません。

空のテンプレートルールを作成し、そのノードの(コンテンツ)プレースホルダを削除することができます。 空のテンプレートルールにより、処理(出力)を行わないノードを定義することができます。

■ 関連項目

- テキストとして XML コンテンツを挿入
- デザインビューシンボル
- <u>デンプレートとデザインフラグメント</u>

5.2 MS Word コンテンツの挿入

Microsoft Word 2007+ がマンノにインストールされていると、コンテンンをWord ドキュメトから、デザイノに静的コンテンツとして貼りつけることができます。Word コンテンソは、適切に対応するデザインコンポーネント内挿入され、テキスト書式プロバティは、Word コンテンソから移動されます。ます。 例えば、Word 段落ブロック内のテキストコンテンソは、パラグラスのコンポーネント内に挿入され、テキストの書式は保管されます(下のスクリーンショナを照)。

Accelerate XML Development

Today, eXtensible Markup Language (XML) technologies play a critical role in all software development projects. XML has received widespread support and adoption in the computer industry because of its simplicity, extensibility, interoperability, and flexibility, all of which stem from its power to represent data independent of programming language, platform, or operating system. In XML-based applications, XML is used alongside complementary technologies such as XML Schema, XSLT, XQuery, Web services, and others. Today's developer needs a tool for creating, editing, and debugging these XML-related technologies in an efficient, standards-based manner.

Altova XMLSpy 2007 delivers all the power you need to create the most advanced XML applications, yet at the same time it's flexible enough to allow you to work with XML using the views and options that best suit your specific requirements and preferences. XMLSpy 2007 increases productivity by allowing you to develop higher quality, standards-conformant XML-based applications more quickly than ever before.

Word コンテンツ





Today, eXtensible Markup Language (XML) technologies play a critical role in all software development projects. XML has received widespread support and adoption in the computer industry because of its simplicity, extensibility, interoperability, and flexibility, all of which stem from its power to represent data independent of programming language, platform, or operating system. In XML-based applications, XML is used alongside complementary technologies such as XML Schema, XSLT, XQuery, Web services, and others. Today's developer needs a tool for creating, editing, and debugging these XML-related technologies in an efficient, standards-based manner.



Sp Altova XMLSpy 2007 delivers all the power you need to create the most advanced XML applications, yet at the same time it's flexible enough to allow you to work with XML using the views and options that best suit your specific requirements and preferences. XMLSpy 2007 increases productivity by allowing you to develop higher quality, standards-conformant XML-based applications more quickly than ever before.



Word コンテンソは、デザイン内に貼り付けられています。適切な段落フォーマトが適用され、書式は保管されます。

メモ Word コンテンソコ加え、Word ドキュメントとして貼り付けることのできるコンテンソは、Style Vision デザイン内に貼り付けることができます。これは MS Excel テーブルとHTML ページコンテンソを含みます。

火モ W ord ドキュメント全体から静的コンテンソを含むSPS を作成するコよ、ファイル 新規作成 W ord 2007+ から新規作 成 コマンドを使用して新規 SPS を作成します。

サポートされるWord 機能

次のWord 構造と書式は、Word コンテンカデザイノコピーして貼り付けられた場合サポートされます

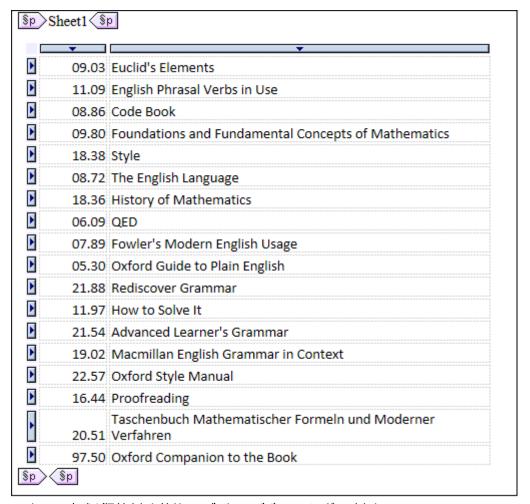
- フォーマナされたテキスト
 - 異なるフォント、サイズ、太線、スタイル、テキストの装飾など
 - o **煙**
 - 背景色
 - テキストの周りの枠
- 段落
- 改ページ
- 水平線
- ノイパーレンク
- ブッケマーク
- テーブル
 - 行の範囲、列の範囲
 - フォーマット済み//ッチコンテンツ
 - o ネストされたテーブル
 - ヘッダー、フッター
- リストナブリスト
 - 箇条書き 異なるスタイル
 - 列挙: 異なるスタイル
- イメージ

5.3 MS Excel コンテンツの挿入

Microsoft Excel 2007+ がマンイこインストールされている場合、コンテンツを Excel ドキュメトからデザインに静的コンテンツとして貼り付けることができます。 Excel コンテンツは、静的テーブルおよび他の適切に対応するデザインコンポーネントとして挿入されます。 書式 プロパティは保管されます、 アのスクリーンショナを照。 各 Excel シートは、個別の静的テーブルとして挿入されます。

1	Α	B				
1	09.03	Euclid's Elements				
2	11.09	English Phrasal Verbs in Use				
3	08.86	Code Book				
4	09.80	Foundations and Fundamental Concepts of Mathematics				
5	18.38	Style				
6	08.72	The English Language				
7	18.36	History of Mathematics				
8	06.09	QED				
9	07.89	Fowler's Modern English Usage				
10	05.30	Oxford Guide to Plain English				
11	21.88	Rediscover Grammar				
12	11.97	How to Solve It				
13	21.54	Advanced Learner's Grammar				
14	19.02	Macmillan English Grammar in Context				
15	22.57	Oxford Style Manual				
16	16.44	Proofreading				
17	20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren				
18	97.50	Oxford Companion to the Book				

Excel シナ



テキストの書式が保持された静的テーブルとしてデザインにインポートされた Excel コンテンツ

メモ Excel コンテンソコ加え、Excel ドキュメントとして貼り付けることのできるコンテンソは、Style Vision デザイン内に貼り付けることができます。これは、MS Word コンテンソ とHTML ページコンテンソを含みます。

メモ Excel ドキュメント全体が静的エンテンンを含む SPS を作成するコよ ファイル 新規作成 | Excel 2007+ から新規 作成 コマインを使用して新規 SPS を作成します。

サポートされる Word 機能

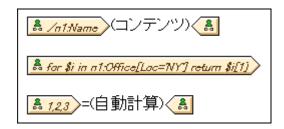
次のWord 構造と書式は、Word コンテンンかデザイノコピーして貼り付けられた場合サポートされます

- テーブル
 - 列の範囲、行の範囲
 - フォーマナ済み//ッチコンテンツ
 - o *ネストされたテーブル*
 - ヘッダー、フッター
- フォーマナされたテキスト
 - 異なるフォント、サイズ、太線、スタイル、テキストの装飾など
 - 色
 - 背景色
 - テキストの周りの枠

- 段落
- ・ノイパーレク
- リストサブスト
 - o *箇条書き 異なるスタイル*
 - 列挙: 異なるスタイル
- イメージ

5.4 ユーザー定義テンプレート

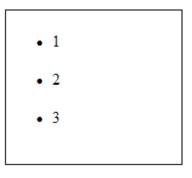
ユーザー定義テンプレートとは XPath 条件式により生成されるアイテムで対して適用されるテンプレートのことです。これらアイテムは原子型の値割させ、ドとなります。以下のスクリーシショナでは3つのユーザー定義テンプレートが示されており、タグの左側にユーザー定義テンプレートアイコンが示されます。ユーザー定義テンプレートが持つ柔軟性により、テンプレートを作成することができます。



上に示されるこれら3つのユーザー定義テンプレートに対して定義されたXPath 条件式により、以下の処理が行われます:

- ソーススキーマ内のノードを選択します。XPath 条件式を使用することで、全てのエンテキストノードから、任意のスキーマノース内にある任意のノードへアクセスすることができるようけないます。曖昧性が無いかたちで StyleVision が定義されたノードを識別することができる場合、テンプレートはユーザー定義テンプレートから通常のテンプレートへ自動的に変換されます。ユーザー定義テンプレートの場合、テンプレートタグの左側にユーザー定義テンプレートを表す緑色のアイゴンが表示されます。
- XPath 2.0 とXPath 3.0 のfor コンストラケトにより指定された条件を満たすノードを選択します。このようなテンプレートは Style Vision に通常のテンプレート(ユーザー定義テンプレート) を明確に解決することを許可しません。
- 原子型の値のシーケンス(1、2、3)を選択します。原子型の値に対してテンプレートを作成することは可能ですが、このようなテンプレート 内でコンテンソプレース市りがを使用することはできません。これは、(コンテンソプレース市りがこより生成される)xsl:apply-template 処理命令が、(原子型の値ではなく)ノードアイテム・1、が適用できない、ナーがです。そのかわり、箇条書きといす。デザイン要素とともに、自動計算を使用することは可能で、以下のスクリーシンョナの左側にあるユーザー定義関数により、右側のような出力が生成されます。





メモ SPS にてXSLT 1.0 が使用されている場合、XPath 条件式の返り値をノードセナにする必要があり、そうでは場合エラーとなります。

XPath を使ったテンプレートノードの選択を行う利点

スキーマノードをXPath 条件式(ユーザー定義テンプレート)により選択することで、XPath の スセレクターメカニズムを使*すこ*アイテムノード やシーケンスの選択を行うことができるようプロリ、ノードの選択に対してフィルタルグや条件をセットすることが可能でいます。そのため、特定のXMLドキュメトノードを、任意のテンプレートに対して使用することができるようでいます。例え

ば、//Office/Department[@Location="NY"] とう XPath 条件式により、NY とう値を持ったLocation 属性が含まれる Department ノード だけが選択されます。このセグション内にあるその他の例も参照ください。

メモ XPath 条件式に複数のロケーション スペロケーションステップか含まれる場合、複数のロケーション スペステップ内に括弧が使用されているかがアープにや並び替えを行う際に重要にないます。例えば、/Org/Office/Dept とう XPath 条件式は、(/Org/Office/Dept) とう条件式とは異なって方法で処理されます。括弧を伴わない条件式では、各ロケーションステップが繰り返されるのに対して、括弧付きの条件式の場合、全てのOffice 要素にある全 Dept 要素が、1つのノードセナにより返されます。

括弧	実際のXSLT メカニズム	効果
無し	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"></xsl:for-each></xsl:for-each></pre>	各 Office 要素は地自の Dept が含まれ、グループ化や 並び替えは、各 Office 内部で 行われる。
有り	<pre><xsl:for-each select="/Org/Office/Dept"> </xsl:for-each></pre>	Dept 要素が 全てのOffice なびいこOrg 要素に対して展 開される。

グループ化や並へ替えを行う際にXPath 条件式の評価方法が大きな違いをもたらします。

ユーザー定義テンプレートを挿入する

以下の操作によりユーザー定義テンプレートを挿入することができます:

- 1. デザイン要素の挿入ツール・インあるユーザー定義テンプレートを挿入アイコンをクックし、デザイン内のテンプレートを挿入する場所を クックします。 まけは、テンプレートを挿入する場所を右クックし、コンテキストメニューからユーザー定義テンプレートを挿入コマードを選 択します。
- 2. In the XPath 式を編集 ダイアログラーC XPath 条件式を入力し、OK をクトックします。XPath 条件式のエンテキストノードは、クトックされたノードしてお高に注意してください。空のノードテンプレートが作成され、場合によっては結合されたノードが作成されます。ノードが結合されると、ターゲトとなるインスタンスノードが同一のレベルコで選択されるのコップし、ノードが結合されていてい場合(つまり複数の階層レベルコで分割されている場合)、ループにより各インスタンスノードが選択されます。グループ化や並べ替えが行われない限り、(結合まどは分割により)返された両者のノードセットに違いながません。結合されたノードがブループ化や並べ替えメカニズムでどのように扱われるかの詳細については、・ノードランプレートオペレーションを参照してください。

テンプレートマッチの編集

任意の、ユーザー定義まけば通常のノードテンプレートに対する選択は、新たな XPath 条件式により変更することができます。ノードテンプレートのテンプレートマッチを編集するゴは、ノードテンプレートを右クトックし、テンプレートマッチの編集コマンドを選択します。 XPath 条件式を編集するごかのダイアログが表示されるので、新たなノードを選択する XPath 条件式を入力し、「OK」をクトックします。

ユーザー定義テンプレートにノードを追加する

SPS のスキーマノースコンテキスト内に配置することができないソードまけよシーケンスコ対してユーザー定義テンプレートが作成された場合、

スキーマソーから得られたノードがユーザー定義テンプレートへ追加される時に、新たむノードのエンテキストが決定できなくなります。そのため、新たむノードが(i)名前で(つまり相対/分で)参照されるのか、ませは(ii)スキーマノースのルートからのフル・公により参照されるのか決定するよう促されます(以下のスグノーンショントを参照)。



どのような動作を行うか、確認するのかデフォルトの振る舞いてなっています。 「ツール」 オプション ダイアログ」 のデザインタブにて、デフォルトの振る舞いを変更することができます。

□ 関連項目

- SPS ファイル コンテンツ
- ノードテンプレートオペレーション
- ユーザー定義要素

ユーザー定義要素、XML テキストブロック 5.5

ユーザ・定義要素 とユーザ・定義 XML テキストブロックにより、それぞれ(i) 任意の要素、(ii) 任意のXML テキストブロックをデザインへ 挿入することができます。これらの機能により、ソーススキーマならびに StyleVision デザイン要素の シナから得られた XML 要素やデザ イン要素以外の要素を追加することが可能になり、(i) 独自に定義した要素に対するテンプレート(ユーザー定義要素)や、(ii) 独自のオブ ジェケ(例えばActiveX オブジェケト)を作成する、独立したXML コード(ユーザー定義ブロック)を作成することができるようになります。

ユーザー定義要素とユーザー定義 XML テキストブロックコは重要な違いがあります。ユーザー定義要素は(属性を伴う)単一のXML 要素 に対するテンプレートノードとしてデザインにて作成され、このテンプレート内にある全てのエレテンンを明示的に作成する必要があります。 エレテ ンソは SPS で使用することができる様々なデザイン要素により構成されます。ユーザー定義 XML テキストブロック にデザイン要素を含める必要はなく、独立したブロックとして扱われます。ユーザー定義要素は空の状態で作成され、多くのコード行を必要とするオブジェクトを作成する必要はありません。

ユーザー定義要素ならびにユーザー定義テキストブロックは Enterprise エディションの Altova 製品に搭 妊 載された Authentic View でのみサポートされます。

- 関連項目
- ユーザー定義要素
- ユーザー定義 XML テキストブロック
- ユーザー定義デンプレート

ユーザー定義要素 5.5.1

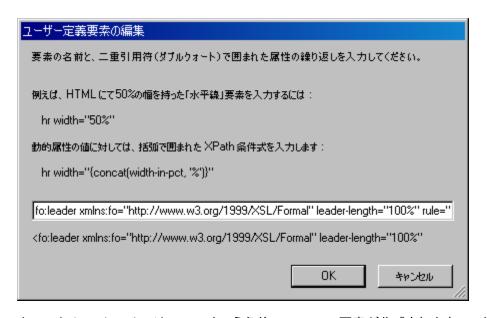
ユーザー定義要素とは、SPS の任意のスキーマノース内には無い要素を出力にて生成することができる要素のことです。この機能により、任 意の名前空間(例えばHTML やXSL+O)にある要素を、デザイン内の任意の場所に挿入することが可能になります。その後、挿入され た要素内部にSPS デザイン要素を挿入することができます。

圱 ユーザー定義要素は、Enterprise エディションの Altova 製品に搭載された Authentic View でのみサポー トされます。

ユーザー定義要素の挿入

以下の方法によりユーザー定義要素を使用することができます。

- 1. デザイン内にてユーザー定義要素を挿入する場所を右クリックします。 2. コンテキストメニューから「ユーザー定義アイテムを挿入 | ユーザー定義要素」を選択します。
- 2. コンテキストメニューから「ユーザー定義アイテムを挿人 | ユーサー定義安素」を選択しより。3. ダイアログが表示されるので、要素名や属性とその値、そして(ドキュメントに要素が含まれない場合は)名前 空間宣言を入力します。



上のスクリーンショットでは、leader という名前の XSL-FO 要素が作成されます。fo: というプレフィックスが 与えられ、そのプレフィックスが xmlns:fo="http://www.w3.org/1999/XSL/Format" という名前空間宣言に紐付 けられています。要素には leader-length や rule-style という属性が含まれており、それぞれ値が与えられています。要素やその属性、そして名前空間宣言は山かっこ("<" ならびに ">")無しで入力してください。
4. 「OK」をクリックして、デザインへ要素を挿入します。開始ならびに終了タグだけを伴う空のテンプレートとして

要素がデザイン内に表示されます(以下のスクリーンショットを参照)。



- 5. その他のテンプレートと同様に、テンプレートへコンテンツを追加することができます。ユーザー定義要素には 静的コンテンツや XML ドキュメントから得られた動的コンテンツだけではなく、その他のユーザー定義要素を 含めることができます。
- 圱 特定の出力でだけ表示を行うように追加されたユーザー定義要素を出力ベースの条件で囲むことにより、 その他の出力にて予期しない結果を回避することができます。
 - □ 関連項目
 - ユーザー定義 XML テキストブロック
 - ユーザー定義テンプレート

ユーザー定義 XML テキストブロック 5.5.2

ユーザー定義 XML テキストブロックは SPS から生成された XSLT コード内に挿入することができる XML フラグメトのことです。 デザイン 要素を追加することができない独立したブロックとして、SPS デザイン内部に配置されます。このような XML テキストブロックは、例えばスタイ ルシート内の目的の場所にXSLTコードを挿入するといか時に使用されます。

この機能により、XSLT フラグメントやカスタマイズされたコードを、デザイン内に挿入するといった操作を行うことができます。例えば、HTML の SCRIPT 要素内部にActiveX オブジェクトを挿入することができます。

火モ この機能は、Enterprise エディションの Altova 製品(StyleVision、Authentic Desktop、Authentic Browser、XMLSpy) に搭載された Authentic View でのみサポートされます。

ユーザー定義 XML テキストブロックの挿入

以下の操作によりXML テキストブロックを挿入することができます:

- 1. ユーザー定義ブロックを挿入するデザイン内の位置を右クリックします。 2. コンテキストメニューから、「ユーザー定義アイテムを挿入 | ユーザー定義ブロック」を選択します。
- 3. 表示されるダイアログにて、挿入する XML テキストブロックを入力します。 XML テキストブロックは整形式でな ければならない点に注意してください。



上のスクリーンショットでは、HTML のリストを生成する XML テキストブロックが追加されます。

- 4. 「OK」をクリックして要素をデザインに挿入します。XML テキストブロックがテキストボックスとして表示されま す。
- 火モ 特定の出力でだけ表示を行うように追加されたユーザー定義 XML テキストブロックを出力ベースの条件 で囲むことにより、その他の出力にて予期しない結果を回避することができます。
- 関連項目
- ユーザー定義要素
- ユーザー定義テンプレート

5.6 テーブル

SPS デザインでは、SPS テーブルとCALS/HTML テーブルとり、2種類のテーブルを使用することができます。このセケションでは、これらのテーブルを使用する方法について説明します。

SPS テーブル

SPS テーブルは デザインのエンポーネントで、デザイン内にて構成/フォーマットされます。SPS テーブルはデザイン内の任意の場所で作成することができ、作成できる SPS テーブルの数に制限はありません。

SPS テーブルはデータを表示するためがされて使用されるもので、出力 フォーマナトにおける文法に従って表示されます。SPS テーブルの構造がテーブルセルのエンテンツはXML ドキュメント内部のノードから得られる場合もありますが、構造そのものがXML ドキュメント内部のノードで表示されることはありません。

SPS テーブルコお以下の2種類が存在します:

- 静的テーブル出よ、SPS のデザイナーにより作成されるテーブルの工とです。テーブルの構造を作成した後に、各セルのエンテンタを個別に定義します。セルのエレテンソはスキーマソノーにある任意の場所から取得することができ、異なる型も扱うことができます。静的テーブルの行は繰り返される構造では無いことに注意してくたさい。繰り返しが行われないため、テーブルは静的と呼ばれており、XML コンテンからは影響を受けない固定された構造となります。
- 動的テーブル出よ、XMLドキュメトにて繰り返し出現するデータ構造に対応したテーブルのことです。テーブルはサブ構造・少なくとも1つの子属性まけは子要素・を持ったスキーマ要素に対して作成することができます。指定されたサブ構造のインスタンスが複数回出現する場合、その要素が繰り返し表示されます。要素の各インスタンスは動的テーブルの行として扱われ、全てまけは一部の子要素や属性がテーブルのカラムではります。そのため、動的テーブルの構造は XML ファイルのコンテンソニ依存し、コンテンソニより動的に変化します。

CALS/HTML テーブル

CALS テーブルおけまけてML テーブルのコンテンソモデルは、DTD やスキーマの拡張によりXML ドキュメントにて(CALS まけまけてML の仕様に従うかすっちで) 定義されます。SPS デザインでは、CALS/HTML テーブルがテーブルとして処理されるように指定することができます。この場合、CALS/HTML テーブルを表す XML データ構造から出力フォーマットにおけるテーブルマーケアップが生成されます。CALS/HTML テーブルのフォーマットはXML インスタンスドキュメント やSPS (まけばそれら両方)にて指定することができます。

HTML テーブルのHTML プレビューを以下に示します:

Name	Phone
John Merrimack	6517890
Joe Concord	6402387

上のテーブルコ対応するHTMLコードのフラグメントを以下に示します:

```
⟨テーブル border="1" width="40%">
⟨tbody⟩
⟨tr⟩
⟨td⟩ Name⟨/td⟩
⟨td⟩ Phone⟨/td⟩
⟨/tr⟩
⟨tr⟩
⟨td⟩ John Merrimack⟨/td⟩
⟨td⟩ 6517890⟨/td⟩
⟨/tr⟩
⟨td⟩ Joe Concord⟨/td⟩
⟨td⟩ 6402387⟨/td⟩
⟨/tr⟩
⟨/tr⟩
⟨/tr⟩
⟨/tr⟩
⟨/tr⟩
```

</テーブル>

オバナルのXMLドキュメノトは以下のようけなります:

XMLドキュメト内の要素名がテーブルを表すものでは無いことに注目してください。テーブルの構造は、HTML またはCALS テーブルモデルニ対応している必要があます。以下を参照くたさい

- XML 要素だけがHTML カラム要素の〈td/〉 「対応していることご注目してくたさい。
- CALS/HTML テーブルは、table 要素に相当する要素が許可されている場所であれば、XML ドキュメト内の任意の場所に挿入することができます。
- Authentic View では、データをテーブルセルニ直接入力することができます。データは対応する CALS/HTML テーブル要素に対応するコンテンソとして使用されます。
- CALS/HTML テーブルのフォーマナプロッティはXMLドキュメナから得られるまか、SPS デザイノこで指定することもできます。

デザイナー・「関する概要

ドキュメントデザイナーは、以下の点に留意してくたさい

- SPS テーブルの構造は SPS にて定義される一方で、CALS/HTML テーブルの構造はスキーマにて指定され、CALS/HTML テーブルモデルに従う必要があります。スキーマ内の要素名は CALS または HTML テーブルモデル内の名前と異なるものでも許されます。
- SPS テーブルの colspan と rowspan は SPS で指定されます。CALS/HTML テーブルの場合、colspan ならびに rowspan は XML インスタンスドキュメントにて指定されます。
- SPS テーブルのテーブルフォーマットは SPS にて指定されます。CALS/HTML テーブルのフォーマットは、XML インスタンスドキュメント、または SPS にて指定されます。

□ 関連項目

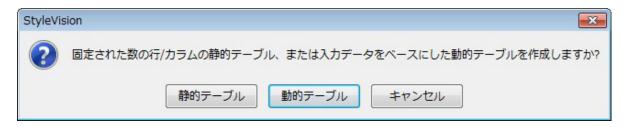
- <u>静的テーブルの作成</u>
- 動的テーブルの作成
- テーブル内の条件付き処理

5.6.1 静的テーブル

以下の操作により静的なテーブルを作成することができます:

1. メニューアイテムから「テーブル | テーブルを挿入」または「挿入 | テーブルを挿入」を選択するか、デザイン要素の挿入ツールバーにあるテーブルの挿入 アイコンをクリックします。

2. テーブルの作成ダイアログか表示されます(以下のスクリーンショントを参照)。

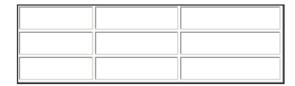


静的テーブルをクルクします。

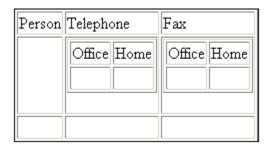
3. テーブルの挿入ダイアログが表示されるので、テーブルの大きさを入力し、使用することができる全ての幅を使用するかを指定します。



4. 「OK」をケリックします。下に示されるように、指定された次元を持つテーブルが作成されます。



5. StyleVision に備わっている通常の機能を使用することで、テーブルのセルコンテンツを入力することができます。直接入力され たテキストや、スキーマソノーから、ラッグされた要素、まけはイメージなどのオブジェケトや入れ子になって・デーブルをテーブルに含めることができます。入れ子構造になって・デーブルの例を以下に示します。



静的な SPS テーブルよ、スキーマ構造の、ヤーるところ:散りばかられた XML データを表示し、静的なコンテンソ(XML ソースから与えられてはいない ロンテンツ を表示するの に適したテーブル形式です。

カラム、行、テーブルを削除する

カラム、行、おけまデーブルを削除するけま、目的のカラム、行、テーブル内にカーノルを移動させ、メニューアイテムのディーブル | 行を削除」、「テーブル | 列を削除」、まけばテーブル | テーブルを削除」を選択します。

ツール デーブルの編集 アイコン

デフォルトでソールドの2段目に配置されているテーブル編集アイエノは、テーブルメニューコマイドコマオをショートカナとなっています。これらのコマイドを使用することで、静的テーブルの構造を編集し、フォーマナプロ、ティを割り当てることができます。これらのアイコノは動的 SPS テーブルコマナしても使用することができます。

■ 関連項目

- 動的テーブルの作成
- テーブル内の条件付き処理
- SPS デザインビュー内のテーブル
- 静的と動的テーブルのフォーマット

5.6.2 動的テーブル

以下の操作により動的テーブルを挿入することができます:

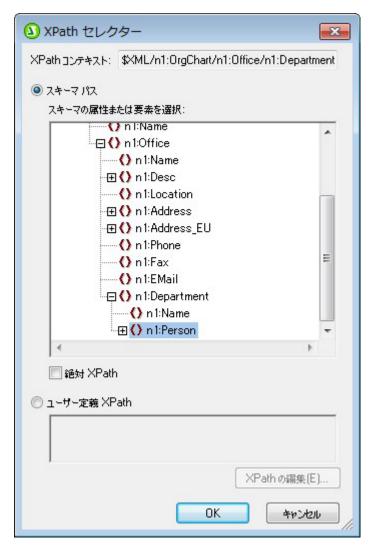
- 2. テーブルの作成ダイアログが表示されます(以下のスクリーンショナを参照)。ツールドーにあるテーブルの挿入アイコンをクトックした場合、テーブルを挿入するデザイン内の箇所をクトックした段階でダイアログが表示されます。



動的テーブルボタンをクリックします。

3. XPath セレクターダイアログが表示されます。XPath コンテキストが、テーブルが挿入される場所になっていることに注目してください。この値をダイアログで変更することはできません。動的テーブルとして作成されるノードを選択してください。以下のスクリーンショットでは、n1:Department がコンテキストノードとなり、n1:Person がテーブルとして作成されるノードとして選択されています。

テーブル 127 SPS コンテンツ



ユーザー定義 XPath オプションを選択した場合、動的テーブルとして作成されるノードを XPath 条件式により選択することができます。
4. 「OK」をクリックします。動的テーブルの作成ダイアログが表示されます(以下のスクリーンショットを参

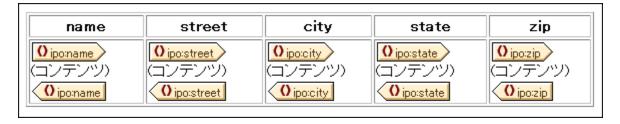
照)。



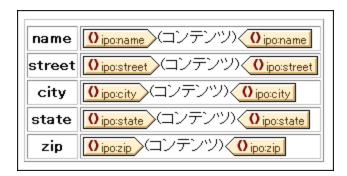
- 5. デザインウィドウンドラッグされた要素の子要素や属性が表示されます。「行/カラムを選択」では、テーブルのカラムとして使用することができるノストが表示されます。カラムとして使用する子要素や属性を選択してください、上の例では、Shares、LeaveTotal、LeaveUsed、LeaveLeft が選択解除されています)。その他のオプションについては以下で説明されます。設定が完了したら「OK」をクトックします。カラムは子要素や属性に対して行う作成され、それより下の階層に対しては作成されないとと言意してください。
- メモ ユーザー定義された XPath を指定して、動的テーブルとして作成するノードを選択すると、ターゲットとなっているノードを曖昧でないかたちで StyleVision により判断することができなくなります。このような場合、テーブルのフィールド(カラム)を選択するための子属性/要素が、動的テーブルの作成ダイアログに表示されず、作成されたテーブルには手動でノードコンテンツを入力する必要があります。ノードコンテンツは、テーブルとして作成される選択ノードの子属性/要素である必要があります。
- **メモ** テーブルとしてスキーマノードを作成する、その他の方法として、スキーマツリーからデザインへノードをドラッグし、ドロップした時にテーブルとして作成することもできます。

縦向きと横向きのテーブル

テーブルが上から下に向けて拡張される場合、テーブルおり下のように表示されます。



テーブルが生から右に拡張される場合、テーブルは以下のように表示されます:



~ダーとフター

カラムと行け、オカラムならび、ご行要素の名前となるヘッダーを与えることができます。カラムヘッダーは各カラムの上部に表示されます。行ヘッダーは、各行の左側に表示されます。ヘッダーを含めるけは、動的テーブルの作成ダイアログにある、ヘッダーを作成チェックボックスにチェックを入れます。テーブルの拡張方向に上から下が選択されている状態でヘッダーが作成されると、テーブルが示えての上にヘッダー行が作成されます。テーブルの拡張方向が生から右になっている場合、テーブルがディーの左側にカラムのヘッダーが作成されます。

フッターを含めるコよ、フッターを作成チェックボックスにチェックを入れます。ヘッダーと同様に、フッターはカラム(カラムの下部に表示)と行(行の右側に表示)に対して作成することができます。数値フィールドの要約チェックボックスにチェックがはいている場合、数値カラムや行の合計がフッターに表示されます。

テーブルメニューからは、ヘッダーやフッターのセルを統合ならびに分割し、行やカラムの挿入、追加、削除を行うことができます。この機能により、ヘッダーやフッターの構造をとても柔軟に構成することができるようによります。更に、ヘッダーとフッターは条件テンプレートや自動計算を含む任意の型の静的ならびに動的エンテンツを含めることができます。

メモ ヘッダーやフッターは、動的テーブルが定義された際に作成する必要があります。ヘッダーの作成を行うには、動的テーブルの作成ダイアログにある、ヘッダーを作成またはフッターを作成オプションにチェックを入れます。動的テーブル内部で行の追加を行なっても、新たな行が作成されるだけで、ヘッダーやフッターは作成されません。この違いは重要で、ヘッダー/フッターを作成コマンドにより、テーブルを含む各ページの上部または下部にヘッダーやフッターが表示されるようになります。行が挿入または追加された場合、動的テーブルとして作成された要素の各出現がそれぞれの行に表示されます。

入れ子になった動的テーブル

ある動的テーブルニ対応する要素の子要素を使用することで、動的テーブル内部に別の(入れ子(ゴンオン)動的テーブルを追加することができます。入れ子(ゴンオ・動的テーブルの作成は以下のように行うことができます:

- 1. 要素の子要素がテーブルのカラムとして作成されるような外側の動的テーブルを作成します。
- 2. デザインビューにある動的テーブルにて、子要素を右クリックします。
- コンテキストメニューから「変更 | テーブル | を選択し、動的テーブルの作成ダイアログを表示します。
- 4. ネストされる動的テーブルのプロパティを定義します。

動的テーブルを静的テーブル内にネストするコよ、動的テーブルとして作成される要素を、目的の静的テーブルのセル・ドラッグし、ドロップした際にエンテキストメニューカら「テーブルの作成」を選択してください。

テキストコンテンツを含む要素をテーブルコ収める

子要素に対してカラムや行を作成するコよ テーブルとて作成された要素内に子要素ノードや属性ノードが含まれて、なければらず、子テキストノードは処理されません。テキストノードの内容をカラムに表示するコよ、例えばTableElement とり要素がらテーブルを作成する場合、Text とり名前の子要素を新付に作成して、TableElement/Text 要素内に目的のテキストノードを挿入します。この操作によりTableElement を動的テーブルとて作成することができます。各行のText コよテキストノードが含まれ、テーブルの各行における内容は、TableElement 要素の出現内容に対応することができます。

テーブルボディーセルのコンテンツ

動的テーブルを作成すると、ノードコンテンツを任意の StyleVision コンポーネトとして作成することができます。上の例では、テーブルボディーのセルがコンテンツとして作成されました。動的テーブルの作成ダイアログでは、セルの表示方法にコンテンツが選択されていました。テーブルはデータ入力デッドスとして作成することもできます。以下の点に留意する必要があります:

- 表示方法の設定は、テーブル全体に対して適用されます。個々のセルの表示を変更するコよ、テーブルを作成した後に個々のセルを編集する必要が対けます。セルを右ケングして、コンテキストメニューの「変更」から、目的のセルコンテンツ型を選択してくたさい。
- セルを要素エンテンツとして作成し、その要素が派生型である場合、セルの文字列エンテンツと、その派生要素全てが連結された力をちでセルに表示されます。

カラム、行、テーブルを削除する

カラム、行、テーブルを削除するコよ、削除するカラム、行、テーブル内にカーソルを配置し、メニューアイテムからテーブル | カラムを削除」、「テーブル | 行を削除」、まコゴテーブル | テーブルを削除」をそれぞれ選択してくたさい。テーブルが入れ子になっている場合、「テーブル | テーブルを削除」コマイでを使用することで、カーソルが含まれているテーブルを削除することができます。

ツール ーテーブルの編集 アイコン

ツール・十二用意されているテーブル編集アイエノはテーブルメニューコマナドへのショートカナとなります。これらのコマナドを使用することで、動的テーブルの構造やフォーマナプロ・ティを挿入、削除、そして編集することができます。これらのアイコノは静的テーブルコ対しても使用することができます。CALS/HTML テーブルは、StyleVision からのみ有効にすることづいます。

グローバルテンプレート内の動的テーブルの作成

動的テーブルはグロー シーテンプレート内にある要素でも作成することができます。テンプレートの作成は、メインテンプレートの要素と同様の方法で行われます。グロー シーテンプレートでは、グロー シーテンプレートノードの派生型要素に対してしか動的なテーブルを作成することができず、グロー シーテンプレート内にある authors という要素に対して動的なテーブルを作成するコま、authors 要素の親要素内に動的テーブルを作成する必要があります。 authors 要素のガロー・シーテンプレート内部にテーブルを作成することなできません。

□ 関連項目

- 静的テーブルの作成
- テーブル内の条件付き処理
- SPS デザインビュー内のテーブル
- 静的と動的テーブルのフォーマト

5.6.3 テーブル内の条件付き処理

静的ならびに動的なテーブルにある個々のカラムや行、またはカラムや行のヘッダーに対して条件付きの処理をセナすることで、条件の評価結果に従うかたちでカラムや行、またはヘッダーを表示、隠すことができます。条件の結果がtrue となる場合、カラムや行、ヘッダーカ表示されます。

条件付き処理の追加と編集

カラムや行、ヘッダーに対して条件付き処理を追加するコよ、目的のデザインエンポーネントを右クトックして、条件処理の編集を選択します。(下のアクリーンショナでは、左上のカラムヘッダーデザインコンポーネントが イライトされて表示されています。 2番目のカラムデザインコンポーネントは青でアナラインされています。 行のエンポーネントはカラムヘッダーデザインコンポーネントの下に表示されています。)



条件処理の編集を選択することで、XPath 式の編集ダイアログ か表示され、条件のXPath 条件式を入力することができます。以下にあるような用途に対して条件付き処理を使用することができます:

- テーブルのカラムや行に対して false() という XPath 条件式を入力することで、それらを隠し、true() により表示させる。
- ◆ カラム内にある値の合計が一定の値を超えた場合にだけ、そのカラムが表示される。
- カラムや行にある全てのセルになんらかの値が与えられている(空でない)場合にだけ、そのカラムや行が表示される。
- 特定のセル値がカラムや行に存在する場合にだけ、そのカラムや行が表示される。

既に作成された条件を編集するけまそのデザインエンポーネントを右クリソして、条件処理の編集を選択してください。XPath 式の編集ダイアログが表示され、条件の評価を行うことができます。

条件処理の削除

カラムや行、ヘッダーの条件処理を削除するコよ 目的のデザインエンポーネントを右クトンして、条件処理のクリアを選択してください。

□ 関連項目

- 静的テーブルの作成
- 動的テーブルの作成
- <u>SPS デザインビュー内のテーブル</u>
- 静的と動的テーブルのフォーマト

5.6.4 デザインビュー内のテーブル

静的ならびご動的 SPS テーブルのエレポーネトを、テーブルマーケアップが有効になっている状態(「テーブル | テーブルマークアップの表示」)で以下に示します:



上のスクノーンショナに示されるテーブルは、上から下に展開され、ヘッダーとフッターか管まれているという単純なものです。

- 下向きの矢印を含む長方形により、カラムのサインが示されます。カラムのサインはカラムの上位に配置されます。(そのカラムに対してフォーマットプロパティを割り当てるなどのために)カラム全体を選択するには、そのカラムのサインをクリックします。
- 行は、右向きの矢印が含まれるサインにより示されます。行のサインをクリックして、行全体を選択することができます。

• 上から下に展開されるテーブルでは、ヘッダーとフッターがそれぞれ上向き下向きのアイコンにより表されます (上のスクリーンショットを参照)。左から右へ展開されるテーブルでは、ヘッダーとフッターは、それぞれ左と右方 向のアイコンアイコンにより表されます(下のスクリーンショットを参照)。

- テーブル全体を選択するには、テーブルの左上(上、そして下にあるスクリーンショットにてマウスポインターが配置されている場所)をクリックします。
- テーブル行またはカラムが選択されると、その背景が濃い青色でハイライトされます。上のスクリーンショットでは、フッターが選択されています。
- 上から下方向に展開されるテーブルでは、作成されたテーブルの要素がカラム-行グリッドの更に外側にある一番左側に表示されます(上のスクリーンショットを参照)。左から右方向に展開されるテーブルでは、作成されたテーブルの要素が、カラム-行グリッドの上に表示されます(以下のスクリーンショットを参照)。



カラムや行、またはテーブルが選択された後には、スタイルならびにプロパティを、スタイルまたはプロパティウィンドウにてそれぞれセットすることができます。

ドラグアイドロプ 機能

(静的または動的な)SPS テーブルのカラムと行をドラッグして、同一テーブル内にドロップすることで、位置を変更することができます。

行とカラムをテンプレートのタグで囲む/削除する

行まけまけたのサインを右クトックして、コンテキストメニューからすりで囲む | テンプレート」まけばすりで囲む | ユーザー定義テンプレート」を選択することで、行やカラムをテンプレートのタグで囲むことができます。その後スキーマソレーからノードを選択、まけはユーザー定義テンプレートに対して XPath 条件式を入力することで、行やカラムの周りにテンプレートが作成されます。



行まけまけテムの周りに配置されたテンプレートは、テーブル内にある行やカラム、影響を与えること無く削除することができます。テンプレートタグが選択されている状態で、Delete キーを押下してください。

行やカラムのエンテングを削除すること無くテンプレートの削除を行い、必要な場合行やカラムを別のテンプレートで囲むということを、テンプレートにより実現することができます。 ユーザー定義テンプレート を使用することで、行ませまカラム内から(例えば自動計算を使用することで)テンプレートマッチの結果を使用することができます。

□ 関連項目

- テーブルメニュー
- 静的テーブルの作成
- 動的テーブルの作成
- 静的と動的テーブルのフォーマット
 - ノードテンプレートオペレーション

• ユーザー定義テンプレート

5.6.5 テーブルのフォーマット

以下の機能を使用することで、静的ならびに動的テーブルのフォーマットを行うことができます:

- (プロパティサイドバーにある) HTML テーブルフォーマットプロパティにて
- (スタイルサイドバーにある) CSS(スタイル) プロパティにて

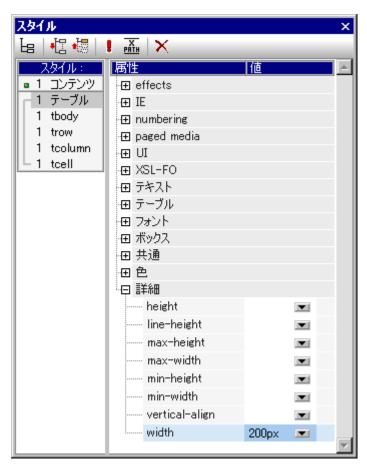
プロパティサイドバー

プロ 守ィサイドバーでは、HTML テーブルフォーマ・ナプロ 守・を利用することができます(以下のスクリーンショナを参照)。テーブルコンポーネント やサブコンポーネント (ボディー、行、カラム、セル)に適用されるこれらプロ ティヘは プロ ティの HTML グループからアクセスすることができます。



スタイルサイドバー

スタイルサイドバーでは、CSS テーブルフォーマ・トプロ・ケイを利用することができます(以下のスクリーンショナトを参照)。テーブルエンポーネントやサブエンポーネン・(ボディー、行、カラム、セル)に対して CSS プロ・ティを使用することができます。



メモ テーブル内にある全てのセルが空になっている場合、Internet Explorer により行が縮退されており、見えなくなっている場合があります。この場合、特定のセルに空白を挿入して対応することができます。

垂直のテキスト

テーブルセル内にあるテキストは、時計回り、まけは反時計回り、30度回転させることができ、上から下まけば下から上方向に、テキストを垂直に表示することができます。 デザインにてテーブルセル内のコンテンクを選択し、プログライナインで tcell を選択します(以下のスクリーンショナを参照)。 テーブルセルプログライグループにある方向プログライの値を選択してくたさい。



以下の点に注意してくたさい

- 出力では回転された状態で表示されますが、デザインでは水平のまま表示されます。
- このプロパティはテキストに対してだけ適用されるべきもので、その他のコンテンツに対して使用しないでください。

テーブルセル内のテキスト以外にも、テキストボックス内のテキストに同様のプログライを適用することができます。

プログライならびバンスタイルによるテーブルのフォーマット

フォーマナ・プロ・ティは、プロ・ティサイド・ドーとスタイルサイド・ドーの両方で利用することができます。 両方のサイド・ドーで利用することのできるテーブルプロ・ティを以下のテーブルニ示します。

テーブルコンポーネント	プロパティサイドバー	スタイルサイドバー	
テーブル	border、frame、rules; cellpadding、cellspacing、bgcolor、height、width (スタイルサイド・ドーこて height、width が指定された場合、ここでの値が上書きされます)	ボックススタイル内にあるborders と padding、詳細グループ以下のheight と width(height とwidth の値はプロ・ディサ イドバー内の値を上書きします)、color、 font。text styles	
ボディー	align, valign	height、vertical-align、color、font、text styles	
カラム	align, valign	width, vertical-align, color, font, text styles, box styles	
行	align, valign	height, vertical-align, color, font, text styles, box styles	
也し	align, valign	height, width, vertical-align, color, font, text styles, box styles	

副とも高

テーブルゲ行、カラム、セルの高さや幅はスタイルサイドバー(の詳細グループ)にてセットすることができます。マナスを使ってデザインに表示されているテーブルベカラム、ませば行の大きさか変更されると、スタイルサイドバーの該当するスタイルにて、変更された値が自動的に入力されます。余白が指定されたセルス対して height やwidth スタイルはサポートされません(row-spanned ませはcolumn-spanned)。

テーブルの中央寄せ

テーブルの中央寄せを行うコよ、プロ・ティのHTML グループにある align プロ・ティの値を center にセットします。テーブルが選択されている状態で、「テーブル | テーブルのプロパティ」を選択することでも、 align プロ・ティヘアクセスすることができます。 まけよ プロ・ティサイド・ドースある HTML グループからも align プロ・ティヘアクセスすることができます。

PDF 出力にてテーブルの中央寄せを行うコよ ご使用のFOP プロセッサース従って設定を行う必要があります。FO の仕様に従えば fo:table-and-caption 要素をとれるうfo:table によりテーブルを囲み、fo:table-and-caption 要素のtext-align 属性の値にcneter を指定することづいます。デザイステーブルが挿入されナジナでは、fo:table-and-caption 属性は作成されませんが、ユーザー定義要素 としてこの要素を追加することができます。Apache FOP プロセッサーを使用している場合、FOP のドージョンによってはではable-and-caption要素がサポーされていないこともあります。この場合、幅が固定されナテーブルを作成し、(「テーブルーテーブルのプロパティ」からアクセスすることができる)テーブルプログライのHTML グループにて width プログライの値を 4in まりは120mm に指定することで、簡単に中央寄せを行うことができます。

交互に表示される行に異なる背景色を与える

以下の方法により、動的テーブルで交互に表示される行に対して背景色をセナすることができます:

1. 背景色を指定する行のアイコンを選択します。テーブルは動的テーブルであり、各要素が行として表示され、テーブルとして作成されている要素に対応する行がデザインには表示されていることに留意してください。

- 2. 行のサインが選択された状態で、プロパティサイドバーの trow プロパティをクリックします。
- 3. bgcolor プロパティを選択します。
- 4. プロパティサイドバーにあるツールバー内にある XPath アイコンをクリックし、XPath 式 の編集ダイアログ にて、以下のような XPath 条件式を入力します:

```
if (position() mod 2 = 0 ) then "white" else "gray"
```

この XPath 条件式により、偶数行の bgcolor が white (白)になり、奇数行の bgcolor が gray (グレー)となります。

上の方法を拡張することで、さらて複雑なフォーマットを行うこともできます。

動的テーブルの行に番号を付ける

XPath 条件式のposition() 関数を使用することで、動的テーブルの行に番号をつけることができます。テーブル内に行番号を表示するカラムを挿入し、そのカラムのセル内にposition() 関数を含む XPath 条件式が記述された自動計算を挿入します。動的テーブルにおける行に対応した要素はエンテキストノードとなるため、行要素セナトにおける各行の位置がposition() 関数により返されます。

PDF 出力におけるテーブルへ・ダーとフッター

テーブルがページをおって表示される場合、テーブルを含むページ全てにテーブルのヘッダーとフッターが表示されます。以下の点に留意してく たさい

- フッターに自動計算が含まれる場合、各ページの最後に表示される自動計算には、(それぞれのページに表示されている行ではなく)テーブル全体に対する自動計算の結果が含まれます。
- 個々のページにおいてヘッダーやフッターの表示を無効にすることはできません。例えば、(各ページに表示されるテーブルの最後ではなく)テーブル全体の最後の行にだけフッターを表示することはできません。

改ページが行われるごとで表示されるヘッダーやアッターの表示を回避するコよ table 要素のtable-omit-header-at-break やtable-omit-footer-at-break プロ ディ(属性)を使用してくたさい。これらのプロ ディへよ スタイルサイド バーの XSL FO グループからアクセスすることができます。 改ページこお おへ ツダーやフッターの表示を回避するコよ それぞれの属性値を true にセットします。 デフォルトの値は false 「プンマイおり、値の指定を行わないと、 改ページが行われるかごパニヘッダーとフッター が挿入されます。

テーブルセルのコンテンツノン イフンを追加する

テーブルセル内のエンテンソン・イフンを追加するコよ、対応するブロックにて、イフンを追加してくたさい。

□ 関連項目

- 静的テーブルの作成
- 動的テーブルの作成
- SPS デザインビュー内のテーブル

5.6.6 行と列の表示

以下に示されるテーブルコ対して用意された行ならびにナラム表示オプションは、HTML 出力に対してのみ使用することができます。これらの機能は Authentic View にてサポートされておらず、SPS のXSLT バージョンに XSLT 2.0 ませは XSLT 3.0 を選択する必要があります。

- 空の行ならびにカラムを自動的に隠す。
- 各カラムに閉じるボタンを配置して、個々のカラムを閉じる。
- 派生したリレーションシップが含まれる行要素に展開/縮退ボタンを表示する。

空の行ならびこカラムをデフォルトで隠す

以下の操作により、HTML 出力にて空の行まりよりうとを隠すことができます:

- 1. デザインビューにて、テーブルまたはその一部(カラム、行、セル)を選択します
- 2. プロパティサイドバーにて、テーブルのプロパティ以下にあるテーブルグループを選択します(以下のスクリーンショットを参照)。



- 3. カラムを隠すまたは行を隠すプロパティに対する値を選択します。これら2つのプロパティで選択することができるオプションは、行わない、空であれば、ボディーが空であれば、と共通しています。空であればオプションにより(ヘッダーとフッターを含む)全てのカラムまたは行が空であれば、そのカラム/行が隠されます。ボディーが空であればオプションでは、ボディーが空である場合にカラム/行が非表示になります。
- **メモ** 非 XBRL テーブルに(行やカラムのセルが結合された)行またはカラム領域が含まれる場合、空の行やカラムを隠す機能が動作しないこともあります。

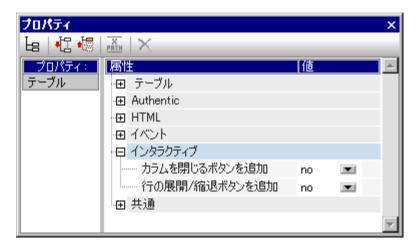
行を展開/縮退するためのカラムを隠す

HTML 出力において、テーブルの各カラムに閉じるボダンを含めるように指定することができます(以下のスクリーンショナを参照)。閉じるボダンをクトックすることで、ユーザーは個々のカラムを隠すことができます。ユーザーによりカラムが隠されると、最初のカラムにプラスシンボルは表示されます(以下のスクリーンショナを参照)。このシンボルをクトックすることで、隠されていた全てのカラムが再度表示されます。

Balance Sheet (in Millions)	2004-09-30	× 2004-07-01 - 2004-09-30	2003-12-31	2004-01-01
⊡ Assets, Total	€21.49	Ĭ	€24.02	
⊞ Current Assets, Total	€10.65		€12.32	
⊞ Non Current Assets, Total	€10.85		€11.7	
⊟ Liabilities and Equity, Total	€21.49		€24.02	
⊞ Liabilities, Total	€8.9		€10.79	
Minority Interests				
⊟ Equity, Total	€12.59		€13.23	
⊞ Issued Capital and Reserves	€12.59		€13.23	

その他にもHTML 出力では、要素名の隣に表示された展開/縮退(プラス/マイナス)シンボルにより、要素の子孫を表示することができます(上のスグノーシショナを参照)。HTML 出力にてこれらのシンボルをクトックすることにより、その行要素が展開ませば縮退されます。 デザインでは、CSS プロ ティから個々の行に対してインディトを指定することができます。

これら2つの機能の設定は、テーブルプロペティのインタラクティブグループにて行うことができます(以下のスクノーンショントを参照)。



プログラインセントすることができる値は、機能を有効にする)yes ませは、機能を無効にする)no となります。

□ 関連項目

- <u>静的テーブルの作成</u>
- 動的テーブルの作成
- SPS デザインビュー内のテーブル
- 静的と動的テーブルのフォーマナ

5.6.7 CALS/HTML テーブル

CALS/HTML テーブルではXML による階層構造が表示され、その要素により(i) CALS ませまHTML テーブルの構造を定義、(ii) そのテーブルのフォーマトを指定、そして(iii) そのテーブルのコレテングを含むことができます。XML の構造は、CALS ませまHTML テーブルのモデルに正確に対応する必要があります。

以下の方法によりCALS/HTML テーブルを作成することができます:

- 1. XML 構造をCALS/HTML テーブル構造として定義する
- 2. テーブルのためのスタイルのフォーマナを指定する
- 3. SPS デザイン内にCALS/HTML テーブルを挿入する

CALS/HTML テーブルが出力されるようにする

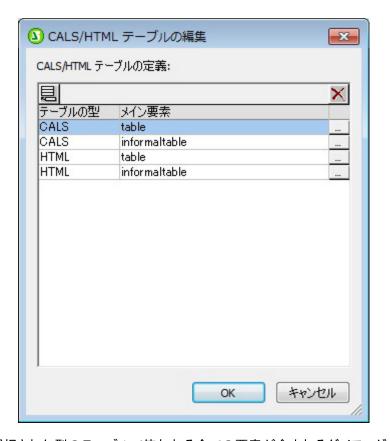
XMLドキュメトロはテーブルの構造とエンテンソを定義するデータ構造を含めることができます。例えば、以下のXML データ構造はHTML テーブルモデルコ対応しており、HTML テーブルモデルと同一の要素名が含まれています。

以下の例で示されるように、HTML テーブルモデルと要素名は異なるものの、HTML テーブルモデルの構造に対応したXML データ構造を使用することもできます。

XMLドキュメトにて定義されたこのテーブル構造は、異なる出力フォーマナトにてテーブルを直接生成するために使用することができます。テーブルを生成すること、このXMLデータをCALS おけまけてに表する必要があります。XMLデータ構造がCALS や(デフォル・の)HTMLテーブルとして定義されていない場合、データ構造内にある要素は一般的な非テーブル要素として扱われ、出力ドキュメナトにテーブルマークアップが使用されることはありません。

以下の方法によりCALS/HTML テーブルマーケアップを生成することができます:

- 1. メニューオプションから「テーブル | CALS/HTML テーブルの編集」を選択します。
- 2. ポップアップが表示されるので、CALS/HTML テーブルとして使用する XML データ構造を入力します。CALS テーブルモデルに関する情報については、CALS テーブルモデルat OASIS を参照(ださい。HTML テーブル構造を含むテーブル要素の例については、プロジェクトウィイヴにある) Example プロジェクトフォルダーの Basic フォルダーに収められている HTMLTable 1.sps を参照ください。スキーマ内にある要素を CALS または HTML テーブル要素として有効化するには、ダイアログの左上にある CALS/HTML テーブルボタンをクリックして、CALS テーブルの追加または HTML テーブルの追加コマンドを選択してください。以下のスクリーンショットでは、table ならびにinformaltable が CALS テーブルとして、その他の HTML テーブルとともに作成されています。「OK」をクリックして確定してください。



3. 選択された型のテーブルで使われる全ての要素が含まれるダイアログ(HTML テーブルの編集)が表示されます(以下のスクリーンショットを参照)。このダイアログにデフォルトで表示される要素名は、選択されたテーブルモデルの要素名となっています(CALS または HTML)。SPS ファイルスキーマ内に、CALS/HTML テーブルモデルの名前と同じ名前の要素がある場合、それらの名前が黒色で表示されます(以下のスクリーンショットを参照)。表示されている要素名が SPS スキーマ内に存在しない場合、要素名は赤色で表示されます。表示されている要素名をダブルクリックして名前を編集することで、スキーマ内の要素名にマッチさせることができます。

テーブル 141 SPS コンテンツ



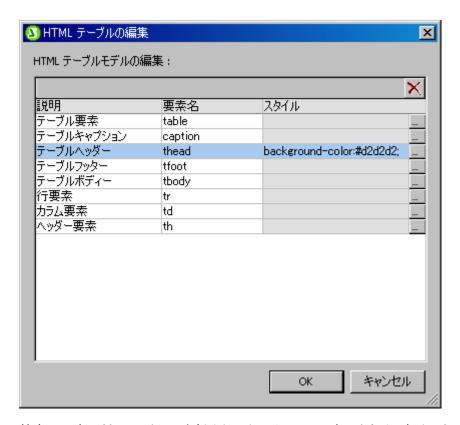
- 4. 「OK」をクリックして、XML データ構造を CALS または HTML テーブルとして定義します。
 5. 入力できる XML データ構造の数に限りはありません(ステップ 2. のスクリーンショットを参照)。同一のメイン要 素を CALS と HTML テーブル型の両方に使用することもできます。
- 6. CALS/HTML テーブルとして使用する XML データ構造の定義を完了したら、「OK」をクリックしてください。

CALS/HTML テーブルが定義され、XML データ構造がCALS/HTML テーブルとして正しく挿入 されたら、データ構造がテーブルとして出 カに送信されます。CALS/HTML テーブル定義を削除するコよ CALS/HTML テーブルの編集ダイアログにて削除する定義を選 択し、ぺんの上部にある削除がみをクリックしてくたさい。

テーブルのフォーマト

CALS/HTML テーブルは2種類の方法によりフォーマナを指定することができます:

- ソース XML ドキュメントにあるフォーマット属性から:CALS ならびに HTML テーブルモデルではフォーマット属性を受け取ることができます。ソース XML ドキュメント内にそのような属性が存在する場合、出力のテーブルマークアップにそれらの値が渡されます。
 CALS テーブルの編集または HTML アーブルの編集ダイアログに表示された個々の要素に対してスタイルを
- 指定(以下のスクリーンショットを参照)。



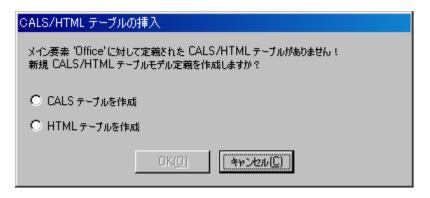
特定の要素に対してスタイルを割り当てるには、その要素のとなりにあるスタイルの追加ボタンをクリックして、表示される スタイルサイバーにて目的のスタイルを割り当てます。各スタイルが、独立した CSS 属性としてその要素へ追加されます。 style 属性により追加されたスタイルは、個々の CSS 属性により追加されたスタイルよりも優先される点に注意してください。 例えば、〈thead style="background-color: red" bgcolor="blue"/> という要素では、 style="background-color: red" 属性が bgcolor="blue" 属性より優先されます。

CALS/HTML テーブル定義にて割り当てられたスタイルを削除するには、その要素を選択(例えば、上のスクリーンショットにおける thead 要素)して、削除ボタンをクリックします。その要素のスタイルが削除されます。

CALS/HTML テーブルをデザインは挿入する

CALS/HTML テーブル構造をデザインへ挿入するコは2種類の方法がかます:

- 1. テーブル要素の親を(コンテンツ)としてデザインに挿入する:親のコンテンツが処理される際に、テーブル要素も処理されます。CALS/HTML テーブル出力が有効になっている場合、要素がテーブルとして出力されます。そうでない場合、テキスト形式の出力が行われます。
- 2. テーブル要素をスキーマツリーからドラッグする:目的の場所にドロップし、CALS/HTML テーブルの作成コマンドを選択することで、CALS/HTML テーブルを作成することができます。要素が<u>CALS/HTML テーブルとて定義されている。場合、CALS/HTML テーブルの挿入ダイアログが表示され、CALS まけまHTML テーブルとて要素を定義することができます(以下のアグナーシンコ外を参照)。</u>



CALS/HTML テーブルとして要素がデザインで作成されると、CALS/HTML テーブルデザイン要素のプレースホルダがその場所へ挿入されます(以下のスクリーンショットを参照)。



テーブル要素のグローバルテンプレート

以下のテーブル要素のグローバルテンプレート が作成されている場合、それらがCALS/HTML テーブル出力内部で使用されます。CALS テーブル title、entry。HTML テーブル caption、th、td。

サンプルファイル

サンプルファイルは(プロジェクトウィンドウ内部の)Examples プロジェクトフォルダー内に収められています。

□ 関連項目

- 静的テーブルの作成
- 動的テーブルの作成

144 SPS コンテンツ リスト

5.7 リスト

SPS にて作成することができるノストにコお以下の2種類があります:

• <u>静的なリスト</u>: コンテンツがSPS 内に直接入力されるリストです。リスト構造がXMLドキュメントの構造が動的に得られること はありません。

動的なリスト: XML ドキュメントから構造とコンテンツを動的に取得するリストです。

このセケションのサブセケションでは、これら2種類のノストを作成する方法について説明します。

- 関連項目
- テーブルとの作業

5.7.1 静的なリスト

静的なリストとは、リストアイテムのコンテンンをSPSに直接入力するリストのことです。以下の方法で静的なリストを作成することができます。

1. 静的なリストを作成する場所へカーソルを移動させ、メニューオプションから<u>挿入 | 箇条書きを挿入</u>を選択します(または<u>デザイン要素の挿入 ツール ド</u>内にある中黒、または番号アイコンをクリックします)。この操作により、作成するリストの種類を選択するためのダイアログが表示されます(以下のスクリーンショットを参照)。



2. **静的リストをク**トンクします。 箇条書き、ませる番号付きのリストを選択するダイアログが表示されます。 (*下のスクリーンション*イ).

SPS コンテンツ リスト 145



- 3. 目的のリストアイテムマーカーを選択し、「OK」をクリックします。空のリストアイテムが作成されます。4. 最初のリストアイテムに使用するテキストを入力します。5. Enter キーを押下することで、次のリストアイテムを作成することができます。

ネスト構造のノストを作成するコよ、リストアイテム内にカーノルを配置し、メニューオプションから<u>挿入 箇条書きを挿入</u>を選択します。そ の後上記の操作を再度行なってください。

火モ リストを挿入する箇所にカーソルが配置された状態で、書式ツール・にあるリスト(記号)まけはリスト(数字)アイ コンをクトックすることでも、リストを作成することができます。カーノルの挿入位置に最初のノストアイテムが作成されます。

静的テキストをリストに変換する

以下の操作により、静的テキストをノストに変換することができます。

リストへ変換するテキストをハイライトして、メニューオプションからタグで囲む リストを選択します。目的のマーカーを選択した後に「OK」 をクリックします。テキスト内に改行が含まれる場合、各行が個別のノストアイテムとして作成されます。テキストの一部だけがイライトされてい る場合、そのテキストがノストアイテムとして作成されます。作成することのできるノストの数に制限は無く、必要に応じてEnter キーを押下 することで新たなリストを作成することができます。タグで囲む リスト コマンドはコンテキストメニューからもアクセスすることができます。

- □ 関連項目
- 動的なリスト
- リスト

146 SPS コンテンツ リスト

5.7.2 動的なリスト

動的リストにより、同じ名前を持った兄弟ノードのコンテンソをリストアイテムとして表示することができます。インスタンスの要素がリスト内のアイテムとして表示されます。以下ではそのメカニズムと操作方法について説明します。

一般的なメカニズム

- 任意の要素をリストとして作成することができます。
- 要素がJストとして作成されると、その要素のインスタンスがJストのアイテムとして作成されます。例えば、department 要素内に複数の person 要素が含まれており、department 内にある全てのperson からなるJストを作成する場合、person 要素をJストとして作成する 必要があます。
- 要素に対してリストを作成した後に、テキスト、自動計算、動的コンテンツなどの静的まけば動的コンテンツを挿入することで、リストやリストアイテムの外観ませばコンテンツを修正することができます。

動的なリストを作成する

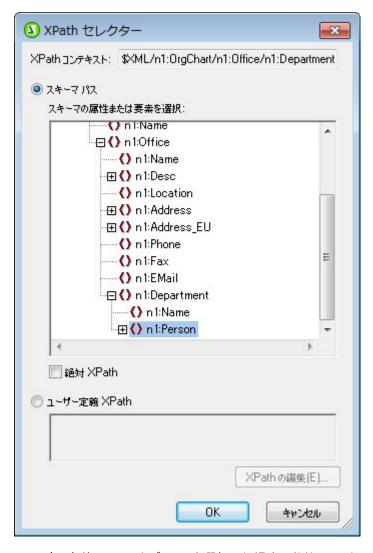
以下の方法により動的リストを作成することができます:

1. 動的リストを挿入する場所にカーソルを移動させ、メニューオプションから <u>挿入 | 箇条書きを挿入</u> を選択します。挿入するリストの種類(動的または静的リスト)を選択するダイアログが表示されます(以下のスクリーンショットを参照)。



- 2. **動的リストを**クトックします。XPath セレクターダイアログか表示されます(以下のスクリーンショントを参照)。
- 3. XPath セレクターダイアログでは、XPath コンテキストが挿入ロケーションのコンテキストになっており、ダイアログ内では変更できない点に注目してください。動的リストとして作成されるノードを選択します。以下のスクリーンショットでは、コンテキストノードが n1:Department となっており、リストのアイテムとして作成されるノードに n1:Person が選択されています。.

SPS コンテンツ リスト 147



ユーザー定義 XPath オプションを選択した場合、動的リストとして作成されるノードを選択する XPath 条件式を入力することができます。「OK」をクリックして、次のステップにて記述されている箇条書きダイアログを表示します。

4. 箇条書きダイアログにて、作成するリストの種類を選択します。箇条書きリストの種類(黒丸や白丸、黒四角)や、番号付き箇条書きを選択することができ、「OK」をクリックすることで、選択した種類のリストが作成されます。

148 SPS コンテンツ リスト



- □ 関連項目
- 静的なリスト
- <u>|小</u>

5.8 イメージ

デザインドキュメントへイメージを挿入する場合、(SPS デザイナーにより)SPS 内部でイメージの場所を直接指定することができるまか、
XMLドキュメント内にあるノードから取得することができます。イメージの挿入方法については、イメージURI のセグションを参照くたさい。異なる出力形式にてサポートされているイメージの種類については、イメージ型と出力セグションにて記述されます。

イメージのプロノティ

イメージはプログラインイドウでセナナることができます。デザイン内でイメージを選択し、プログライサイドグ・Hこて(i) プログライのカラムこてイメージを選択し、(ii) 目的のプログライグループを選択し、(iii) 選択されたプログライグループにある目的のプログライを選択します。例えば、イメージの高さと幅をセナナるコは、HTML グループにある height とwidth プログライコ値をセナします。

- □ 関連項目
- イメージの挿入
- 青写真イメージ

5.8.1 イメージ: URI とインラインデータ

イメージはデザインドキュメント内の任意の場所は挿入することができます。これらのイメージは出力ドキュメントに表示され、デザインビューではイメージの代わりにサムネイルまだはプレースホルダが表示されます。

イメージを挿入するコは<u>挿入 | イメージ</u> メニューコマンドを選択します。イメージの挿入ダイアログが表示されます(以下のスクノーシンョントを参照)。



イメージへは2種類の方法によりアクセスすることができます:

- イメージをファイルとして、イメージの挿入ダイアログにて URI を入力することでアクセスすることができます。
- Base-16 または Base-64 テキストとしてイメージをエンコードし、XML ファイル内に配置することができます。

イメージファイルを挿入する

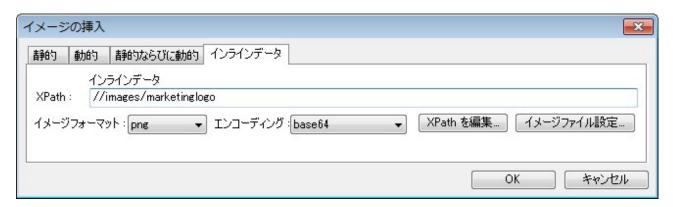
URI を指定することで、イメージファイルかデザイン内に挿入されます。このファイルはランタイムにアクセスされ、ドキュメト内部に配置されます。イメージの挿入ダイアログでは、3種類の方法でイメージのURI を入力することができます(上のスケノージンコナを参照)。

• 静的タブでは、絶対または相対 URI として URI が直接入力されます。例えば、相対 URI であれば nanonull.gif (以下のセグンとを参照ください)となり、絶対 URI であればC:/images/nanonull.gif となります。

- 動的タブでは、(i) 絶対または相対 URI、または (ii) <u>パースされたエンティティ名</u> を含むソードを選択する XPath 条件式を入力します。例えば image /@location というエントリーにより、コンテキストノードの子である image 要素の location 属性が選択されます。 XML ドキュメント内にある選択されたノードにはイメージ URI が含まれます。 非解析対象実体の使用方法については、パースされていないエンティティ URI のセグションを参照(たさい。
- 静的ならびに動的タブでは、XPath 条件式による動的な箇所を、静的なエントリー(テキスト)により囲むことができます。例えば、静的なプレフィックスを C:/XYZCompany/Personnel/Photos/ として、動的なパートを concat(First, Last)、静的なサフィックスを .png とすることができます。この組み合わせにより、例えば C:/XYZCompany/Personnel/Photos/JohnDoe.png といった絶対 URI が選択されるようになります。

エンコードされたテキストのイメージを挿入する

イメージはBase-16 まけはBase-64 でエンコードされたテキストとして XML ファイル内部に保管することができます。この方法により、個別のファイルヘアクセス(リンク)することなく、ソース XML ファイル内にあるイメージヘアクセスすることができるようけっぱます。 XML ソース内部 Iこてエンコードされたイメージを挿入するけま、イメージの挿入ダイアログにあるインラインデータタブを使用します(以下のスクリーンショナを参照)。



XPath 条件式を使用することで、エンコードされたイメージのテキストを含むXMLドキュメント内のソードを指定します。イメージフォーマトコンドドックスから、生成するイメージ形式を選択します(イメージファイルはエンコードされたテキストデータから生成され、このファイルが出力ドキュメントにて使用されます)。エンコーディングコンドドックスでは、ソース XML ファイルで使用されているエンコーディングの形式を指定します。この操作により、エンコードされたテキストを(指定されたエンコード形式により)StyleVision にて正しく読み取ることが可能しています。

(イメージファイル設定がないらアクセスすることができる)イメージファイル設定ダイアログでは、作成されるイメージファイルは対して名前を与えることができます。名前を与えないことも可能で、その場合 StyleVision によりデフォルトの名前が生成されます。

出力からイメージへのアクセス

出力の形式と作業を行なっている段階により、イメージへのアクセス方法が変わります。以下の点に留意してくたさい

- お使いのエディションでご利用になれる出力フォーマットを確認してください: (i) Basic エディションでは HTML が、(ii) Professional エディションでは HTML と RTF が利用可能で、(iii) Enterprise エディションでは HTML、RTF、PDF、そして Word 2007+ が利用できます。
- デザインビュープログライダイアログ にて、イメージス使用される相対が定 SPS ませま XML ファイルス対して相対とするか指定することができます。
- HTML 出力の場合、イメージの URI が HTML ファイルへ渡されることでブラウザーはイメージにアクセスします。 そのため、イメージへのパスが相対パスとなっている場合、HTML ファイルの場所から相対的になっている必要 があります。StyleVision 内にある HTML プレビューでは、一時的な HTML ファイルが SPS ファイルと同じフォル ダー内に作成されます。そのため HTML によるレンダリングを行うには、この場所から相対的となるパスを指定 する必要があります。
- URI が相対的か絶対的かにかかわらず、レンダリングを行うプロセスからイメージへ物理的にアクセスできる状態になっている必要があります。

イメージのプロパティの編集

イメージを変更するコは、デザインビューコでイメージを右クトックして、コンテキストメニューからURLの編集を選択します。実体はイメージの挿入ダイアログと等し、イメージの編集ダイアログが表示され、必要な修正を行うことができるようによります。プロ・ティサイド・バーに表示されているイメージグループのURLプロ・ティからもイメージの編集ダイアログへアクセスすることができます。プロ・ティのイメージグループコはaltプロ・ティも含まれており、イメージの代替テキストを指定することができます。

イメージを削除する

イメージを削除するコよ、イメージが選択された状態で、Delete キーを押下します。

□ 関連項目

- イメージ型 出力
- パースされていないエンティティURI

5.8.2 イメージ型 と出力

StyleVision にてサポトされているイメージと、StyleVision にてサポトされている出力形式を以下に示します。StyleVision のエディションによりサポートされる出力フォーマナのセナが異なる点に注意してください。Enterprise エディションではHTML、Authentic、RTF、PDF、Word 2007+ がサポーされるのは対して、Professional エディションではHTML とAuthentic、RTFが、Basic エディションではHTMLがサポーされます。

イメージ型	Authentic	HTML	RTF	PDF	Word 2007+
JPEG	はい	はい	はい	は、	はい
GIF	はい	はい	はい	はい	はい
PNG	はい	はい	はい	はい	はい
ВМР	はい	はい	はい	はい	はい
TIFF	は*	は*	はい	はい	はい
SVG	は*	は*	ાપત્ર	はい	いえ
JPEG XR	はい	はい	ાપત્ર	いえ	いえ

* 下の点に留意してください

以下の点に注意してください

- デザインビューでは、イメージの場所が静的 URL により与えられている時だけ、つまり SPS に直接入力されたときだけ、イメージが表示されます。
- TIFF ならびに SVN イメージを Authentic View ならびに HTML ビューで表示するには、Internet Explorer 9 以上が必要になります。
- RTF 出力では、TIFF イメージは埋め込みではなくリンクにより表示されるため、大きさの変更を行うことはできません。
- SVG ドキュメントは XML 形式で、SVG 名前空間内に収める必要があります。
- イメージファイルが見つからない場合、FOP によりエラーが返され、PDF は生成されません。

● PDF の生成に FOP が使用される場合、PNG イメージのレンダリングを行うのに JIMI イメージライブラリを FOP からアクセス可能な場所にインストールしておく必要があります。

● FOP による画像の処理に関する詳細については、FOP Web サイトを参照付さい。

サンプルファイル

サンプルファイルのImages.sps は以下の場所に収められています:

HTML はおける SVG

マウスイベトに対するコードが含まれる外部 SVG ファイルがイメージとして使用された場合、イメージ内部でSVG のレンダルングが行われ、インタラクティブ性が失われます。この制約は、オブジェケトとして外部 SVG イメージを扱うか、ユーザー定義された XML ブロックとして SVG コードフラグメトを追加することにい回避することができます。.

SVG をウェブページに含めるはは、以下に示される3種類の方法が別ます。

- 1. <u>イメージとして挿入された</u>外部 SVG:生成されたHTML ファイルにて〈img〉 タグを生成します。インタラクティブ性は失われます。
- 2. ユーザー定義要素機能によりオブジェケトとして挿入された外部 SVG(以下のスクリーンショナを参照): type 属性を正しく挿入するようにしてください(例: type="image/svg+xml")。このように挿入することで、SVG オブジェケトのインタラクティブ性が保たれ、マケスオーバー機能も有効になります。



3. ユーザー定義 XML ブロックにより挿入されたインライン SVG: SVG コードフラグメトの例を以下のスクリーシンコナに示します。 このように挿入を行うことでインタラクティブ性が保たれます。出力方法がHTML 4.0 または5.0 の場合、svg 要素を SVG 名前空間に含める必要はありませんが、XHTML を使用する場合、名前空間が必要になります。

```
ユーザー定義ブロックの編集
 生成された出力ドキュメント内に任意のXMLを配置するには、XMLをテキストとして入力してください。
 <ii>くli>リストアイテムく/li>
    <svg xmlns="http://www.w3.org/2000/svg")</pre>
      <style_type="css">circle:hover tfill-opacity:0.9; stroke-width:2mm;\</style>
      3
  4
  5
  б
  8
   L</svg>
                                                      Н
                                        OK.
                                                 キャンセル
```

■ 関連項目

イメージURI

5.8.3 サンプル: イメージのテンプレート

StyleVision パケードコはStyleVision 口搭載されているイメージ機能のデモするかのSPS ファイルは含まれて、ます。ファイルは(マハドキュメトフォルダー:C:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\Images\Images.sps です。
Images ドキュメト (Images.xml とImages.sps) は 3つの ー から構成されます:

- 2番目のパートには、StyleVision 出力フォーマットごとにサポートされているイメージフォーマットを表した テーブルが含まれています。 RTF、PDF おびW ord 2007+ 出力フォーマットは、 StyleVision のEnterprise Edition とProfessional Edition (RTF) でのみ使用することができます。 デザインビューでは静的 URI をともなう イメージだけが表示されます。このテーブルにあるイメージフォーマットは全て、3番目のパートにて表示されます。
- 3番目のパートでは、StyleVision にてサポートされる全てのイメージが表示されます。Images.sps ファイルを StyleVision にて開いた場合、StyleVision で利用できるプレビューを切り替えることで、どのイメージがそ れぞれのプレビューで表示されるのか確認することができます。イメージの場所は XML ノード内に含まれ ているので、独自のイメージの場所を Authentic View にて入力し、プレビューウィンドウでどのように表示 されるか確認することができます。

■ 関連項目

- <u>イメージURI</u>
- イメージ型 と出力

154 SPS コンテンツ フォームコントロール

5.9 フォームコントロール

XMLドキュメト内にあるノードは(入力フィールドヤコンボドックスといす)データ入力デバスとして作成することができます。 HTML 出力では、デザインビューにおける表示に類似したかたちで、データ入力デバスがサブジェケトとしてレンダレングされます。 HTML 出力では動作していにに注意してくたさい。

一般的なメカニズム

StyleVision で使用することのできるデータ入力デバイスのリストを、XMLドキュメントにてどのようにデータが入力されるかの説明とともに、以下に示します。.

データ入力デバイス	XML ファイル内のデータ	
入力フィール・(テキストボックス)	ユーザーにい入力されたテキスト	
複数行 入力フィールド	ユーザーにい入力されたテキスト	
ユンボボックス	ユーザーの選択を値へマパング	
チェックボックス	ユーザーの選択を値へマパング	
ラジオドダン	ユーザーの選択を値へマパング	
おり	ユーザーの選択を値へマピング	

入力フィールドに入力された値は、XML コンテンソとして XML ドキュメント内に直接入力されます。その他のデータ入力デッドスでは、Authentic View ユーザーの選択が値へマンピングされます。Style Vision では、ユーザーが選択することのできるオプションのリストと、各オプションコ対応した値を定義することができます。通常、ダイアロブニてオプションと、対応する値を定義することができます。

- 入力フィールド (テキストボックス)
- 複数行 入力フィールド
- コンポポックス
- チェックボックス
- ラジオギタン
- ボタン

一般的な使用方法

以下の操作により、データ入力デバイスを作成することができます:

- 1. スキーマソノーサイドバーからデザインビューにある目的の場所へノードをドラッグします。
- 2. コンテキストメニューから、ノードノマオして作成するデータ入力デバイスを選択します。
- 3.データ入力デバイスによってはダイアログが表示されます。ダイアログが表示される場合、必要な情報をダイアログに入力し、「OK」を グルグします。

データ入力デッイスのプロペティを再度開いて編集するコよ(それを含んでいるノードではなく)そのデータ入力デッイスをケックして、プロペティサイド・バーに表示されているプロペティを編集します。

- データは、HTML 出力のデータ入力デ・イスコは入力することができません。 データ入力デ・イスは、コンテンツを表示する代替の 方法としてのみ使用されます。
- 既にあるノードのエレポーネントをデータ入力デッドスへ変更することでも、データ入力デッドスを作成することができます。ノードを右クソクして、変更を選択してください。

HTML 出力では、ユーザーにより選択されたエトリーが表示されます。HTMLドキュメントに表示されたデータ入力デバスの値を変更しても、XMLドキュメントやHTMLドキュメント内にある値は変更されません。

- 関連項目

- 入力フィールド、複数行入力フィールド
- コンポギックス
- <u>チェックボックス</u>
- ラジオドダン、ドダン

5.9.1 入力フィールド、複数行の入力フィールド

スキーマノースウィイ・ウからデザインビューへノードをドロップすることで、SPS 内に入力フィールドや複数行の入力フィールドを挿入することができます。そのノードのエンテンンが入力フィールドまけは複数行の入力フィールドにて表示されます。

入力フィールドのプロ ティを編集する

入力フィールドを選択し、プログライサイドバーのHTML プログライを修正することで、入力フィールドのHTML プログライを修正することができます。

例えば、入力フィールドが選択された状態で、編集フィールドを選択し、プログライのHTML グループ以下にある maxlength プログライを選択します。 maxlength の値フィールドをダブルクトグして、値を入力してください。

メモ CDATA セクションを入力フィールド(テキストボックスと複数行のテキストボックス)に挿入することはできません。CDATA セクションは、テキストコンテンツコンポーネントとして Authentic View に表示される要素内部だけに入力することができます。

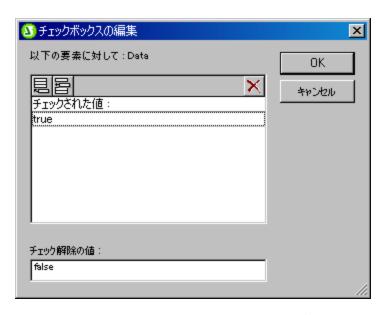
- 関連項目

- ユンボデックス
- チェックボックス
- ラジオイタン、イタン

5.9.2 チェックボックス

チェックボックスをデータ入力デッドイスとして作成することができます。チェックボックスをデータ入力デッドイスとして作成することができます。Basic エディションではチェックボックスの編集ダイアログを操作する必要はありません(Basic エディションではAuthentic View がサポートされないため、XML ファイルコ値を与えることはできません)。

156 SPS コンテンツ フォームコントロール

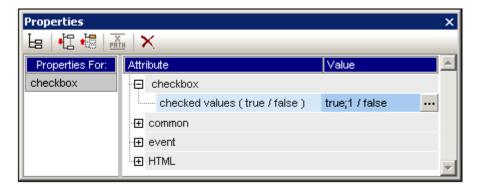


上のスクノーシショナでは、Name とう名前の要素がチェックボックスとして作成されました。 Authentic View ユーザーかチェックボックスこ チェンクを入れた場合、true という値がName 要素の値として入力されます。チェックボックスのチェックが外された場合、(ダイアログレスで定義 されている通りfalse とう値がName のXML 値とて入力されます。

チェックボックスの編集ダイアログへアクセスする

ノードをチェックボックスとして作成して、新たなチェックボックス作成する時にはチェックボックスの編集ダイアログが表示されます。以下の操作に より、チェックボックスの編集ダイアログヘアクセスすることができます:

- デザイン内にあるチェックボックスを選択します。
 プロパティサイドバーにてチェックボックスアイテムを選択し、プロパティのチェックボックスグループを選択します (以下のスクリーンショットを参照)。



- 3. チェックされた値プロパティにある編集ボタン をクリックします。チェックボックスの編集ダイアログが表 示されます。
- チェックボックスが選択された状態でプロパティサイドバーにある HTML プロパティを修正することにより、 旡 チェックボックスの HTML プロパティを編集することができます。
- □ 関連項目
- 入力フィールド、複数行入力フィールド

- ユンボボックス
- ラジオドダン、ドダン

5.9.3 コンボボックス

コンドド・ウンプダウンノスト形式のアイテムが表示されます。リスト内のアイテムは、以下のどれかの方法により選択することができます。リストは生成されがギュメントにて使用することができ、例えば生成された。HTML を更に処理することで、コンドド・クスを使った。HTML のフォームを作成することができます。

- 選択されたノードに対するスキーマの列挙。
- コンドドックスダイアログラこで表されたリストから。表示されるエトリーと、それに対応する XML の値を入力します。 XML の値は Enterprise ならびこ Professional エディションで利用することができ、 Authentic View ユーザーが選択した値から XML 値 へのマッピングが行われます。 Basic エディションでは、この操作は必要ではありません(Authentic View は Basic エディションでサポートされない ヤーが)。
- カレントノードに相対的な XPath 条件式の結果シーケンスから。結果シーケンス内にあるアイテムが「ロップダウンリストのエントリーとして表示され、

5.9.4 ラジオボタン、ボタン

ボダンコよ、通常のTダンとラジオTダンの2種類がかます。 ラジオTダンとTダンよ フォームこ入力する場合、または、 HTML 出力内でイベントをトリガーする場合役「さきます。

メモ ラジオボタンやボタンが選択された状態で、プロパティサイドバーにある HTML プロパティを修正することで、ラジオボタンやボタンの HTML プロパティを編集することができます。

□ 関連項目

- 入力フィールド、複数行入力フィールド
- ユンポポックス
- チェックボックス

158 SPS コンテンツ リンク

5.10 リンク

リンク(まけま) イパーレク)を、ドキュメト内のブックマークやケェブページという外部リソース されて作成することができます。リングは動的に生成されたアンカーで対しても作成することができます。 Style Vision では、ハイパーレクは使用される URI を非常に柔軟な方法で作成することができます。

ブックマークとハイパーレクのセクションでは、静的ならびに動的なブックマークをドキュメント内に作成し、ブックマークや外部ドキュメントへのリンクを作成する方法について記述されます。

□ 関連項目

• <u>ブッケマー</u>クとハイパーレク

5.11 バーコード

バーコード、バーコードデザイン要素を使用することで、出力ドキュメト内に、ドコードを生成することができます(以下のスクノーンショナを参照)。デザインドキュメント内の、ドコードを挿入する場所に<u>バーコードデザイン要素を挿入し、プロンティ</u>を指定します。 バーコードデザイン要素は XSLT 2.0 または XSLT 3.0 (XSLT 1.0 ではない) モードでのメサポートされるということで注意してくたさい。

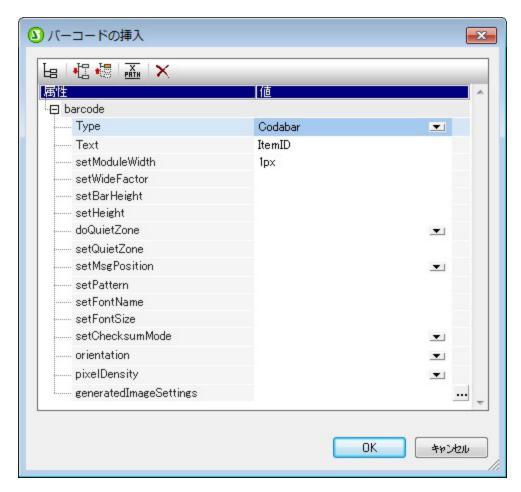


重要: バーコードを動作させるためコよ Java Runtime Environment (JRE) がインストールされている必要があります。JRE はバージョン 1.4 以降で、お使いのシステムにインストールされた Style Vision パッケージのビナバージョン (32-ビットまけばる4-ビッナ) と同一のものではければなりません。

バーコードを挿入する

以下の方法により、デザイン内部にントコードを挿入することができます:

1. バーコードを挿入する場所にて右クリックして、バーコードの挿入コマンドを選択します。その他にもメニューオプションから「挿入 | バーコードの挿入」を選択するか、ツールバーのバーコードアイコンをクリックして、デザインにてバーコードを挿入する場所をクリックします。スキーマツリーからデザインビューへ要素をドラッグアンドドロップして、バーコードの作成を選択することもできます。バーコードの挿入ダイアログが表示されます(以下のスクリーンショットを参照)。



- 2. Type ならびに Text プロパティは必須プロパティとなっている一方、それ以外のプロパティはオプション、または適切なデフォルト値が与えられています。Type プロパティの値はドロップダウンリストから選択することができ、ISBN バーコードを含む EAN-13 や UPC-A といったバーコードの種類を選択することができます。 バーコードプロパティについては 下で説明されます。 必須プロ ティの値と、目的口応じたオプションのプロ ティをセナしてください。 XML ファイル内にある値をプロ ティの値とて使用する場合、 XPath 式を入力 することで、アクセスする XML ノードを特定することができます。 XPath 条件式を使用するコよ、目的のプロ ティを選択し、プロ ティダイアログのソール・イニある XPath ボタンを有効にして、XPath 式 の編集ダイアログこて XPath 条件式を入力します。 XPath 条件式はカレントノード内で評価されます。
- 3. プロパティの設定が完了したら、「OK」をクリックします。バーコードイメージが挿入され、生成されたバーコードを出力プレビューにて確認することができるようになります(以下のスクリーンショットを参照)。



メモ バーコードのイメージは PNG ファイルとして生成されます。

バーコードのプロパティ

以下にある、デーコードのプロ・ティを指定することができます。Text ならびにType プロ・ティオ必須となっており、それ以外のプロ・ティオオプションとよります。 バーコードの種類により、利用することのできるプロ・ティか異なるということに留意してください。

• Type: 与えられたテキストを読み取るためのバーコードシステムです(例: EAN または UPC)。

- *Text:* バーコードパターンの生成に使用される値です。
- SetModuleWidth:コード内にあるバーの幅を指定します。
- SetBarHeight: バーの高さを指定します。
- SetHeight: バーコードのイメージの高さを指定します。
- DoQuietZone: Yes または No という値により、SetQuietZone と SetVerticalQuietZone プロパティにより 大きさが指定される余白を表示するかが決定されます。
- SetQuietZone: バーコードの周りに「クワイエットゾーン」(余白)が表示されるようにします。1次元のバーコードの場合、ここで指定された値が水平方向に対して適用されます。2次元のバーコードの場合、水平ならびに垂直方向に対して値が適用されます。垂直方向の値は、SetVerticalQuietZoneプロパティに値を指定することで、上書きすることもできます。
- SetVerticalQuietZone: 2次元のバーコードに対して、垂直方向の「クワイエットゾーン」(余白)を指定することができます。長さの単位であるミリメートル (mm) を指定する必要があります(例: 2mm)。
- SetMsgPosition: バーコードから見て、どこにメッセージテキストが表示されるか指定します。top、bottom、または none (メッセージが生成されません)を指定することができます。
- SetPattern: 数字だけの長い文字列といったメッセージテキストを、読みやすくするためのパターンをセットします。パターンの構文は以下に示されます。
- SetFontName: テキストを表示するためのフォントを指定します。
- SetFontSize: テキストに使用されるフォントサイズを指定します。
- SetChecksumMode: 以下の値を指定することができます: (i) Add: メッセージにチェックサムが追加されます、(ii) Check: バーコードのレンダリング中にチェックサムのチェックが行われます(チェックサムが存在する場合)、(iii) Ignore: チェックサムの処理が行われることはありません、(iv) Auto: 選択された種類のバーコードにて定義されたデフォルトの振る舞いが有効になります。
- Orientation: バーコードの回転を指定することができます。反時計回り方向に90度ごとの回転が行われます。
- PixelDensity: バーコードイメージにおけるピクセルの密度(解像度)が指定されます。密度が増えることにより、より鮮明なイメージが生成されます。
- GeneratedImageSettings: 生成されたバーコードのイメージファイル名が指定されます。名前が指定されなかった場合、StyleVision により自動的に名前が生成されます。

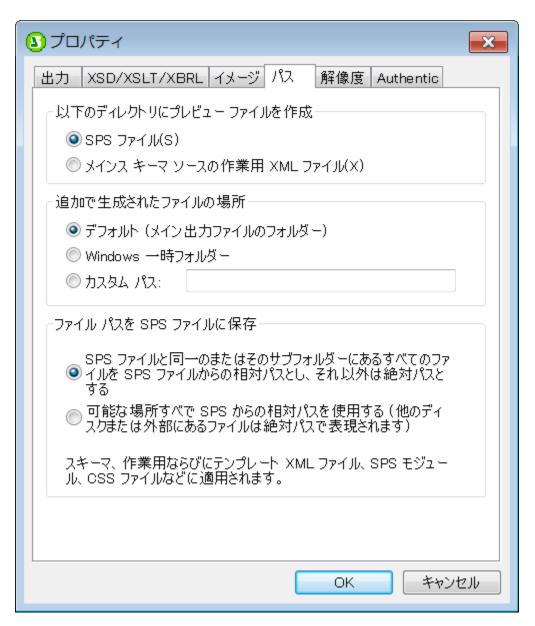
パターンの構文

/ やーノにより、バーコードに含まれる入力メッセージの文字列をお読みやすい かたちで出力することができます。/ やーンでは、入力メッセージのテキストがアンダースコアの"_"により示されます。/ やーン内にあるその他の文字は、出力メッセージテキストの対応する箇所に挿入されます。/ ドンクスラッシュ "¥" はエスケープンンボルとなります。例えば "¥?" により "?" とり文字が出力メッセージテキストに表示されます(ここでの"?" は全ての文字を表します)。 また、"#" 文字により、オルジナルのテキストに含まれていた文字を削除することができます。 以下にある やーンの構文例を参照くたさい。

入力メッセージテキスト	ノダーシ	出力メッセージテキスト
123456		12 34 56
15032011094655	_\\:_:_ UTC	15\03\2011 09:46:55 UTC
15-03-2011	#/#/	15/03/2011

出力ファイルを生成する

出力にて生成されるバーコードイメージのファイルは、プロイティダイアログのパタブにて指定された場所に保存されます。プロイティダイアログへは、メニューオプションからファイル | プロイトティ」を選択することでアクセスすることができます(以下のスグリーシショナを参照)。



プルビューで使用される、ドーコードのイメージファイルは、SPS ファイルまけば作業用 XML ファイルが収められているディレクトリに作成されます。これらのファイルは一時的なファイルとして作成され、SPS が閉じられる際に削除されます。メニューオプションの「ファイル | 生成されたファイルの保存」により出力が生成されが際には任意の場所に、ドーコードイメージファイルを作成することができます。パタダブニある「追加で生成されたファイルの場所」。ペイノにて場所を指定することができます(上のスクリーンショナを参照)。

- 関連項目

• SPS ファイル・コンテンツ

5.12 レイアウト モジュール

レイアナモジュール出ま、レイアナを含むオブジェケルのことです。モジュール全体を SPS デザイン内に挿入することにより、ドキュメ・トフロートにおけるブロックとして作成することができます。デザインの要件に従って、SPS デザイン要素を含む複数のレイアナボックスをレイアナモジュール内に配置することでできます。レイアナモジュールを使用することで、グラフィカルなデザインアプリケーションと同じような操作で、レイアナを作成することができます。

レイアウトモジュールを作成するナングのステップを以下に示します:

- 1. <u>レイア・ナコンテナー</u>を挿入するレイア・ナコンテナーはページ上の幅全体を占めることも、大きさを指定することもできます。デザイン の青写真を含めることでデザインガイドとして使用し、(サイド・ドーにある) スタイルを使用することでレイア・ナコンテナーのフォーマ・ナ を行うこともできます。
- 2. レイアウトコンテナー内に1つ以上の <u>レイアナボックス</u>を挿入するレイアナボックスコは複数のデザイン要素(スキーマテキスト、スキーマノード、自動計算、イメージ、リストなど)を含めることができ、(スタイルサイド、ドーコある) スタイルを使用することで、レイアナボックスのフォーマナを行うことができます。レイアナコンテナー内にあるレイアナボックスは、他のレイアナボックスコ対して相対的に位置を変更し、前面お出ま背面へ移動することができます。
- 3. <u>ライン</u> の描画やフォーマナ、また(レイアナンボックスやその他のラインという)レイアナンオブジェケトの前面や背面に移動することが できます。

フォームベースのデザイン

<u>新規 SPS の作成</u> する場合、フノー・ンドのデザインか、フォームを基にしたデザインかを選択することができます。フォームを基にしたデザインとは、レイアナーコンテナーには構成される SPS デザインのことです。

メモ レイアウトモジュールは、Enterprise エディションの Altova 製品における Authentic View でしかサポートされません。

- □ 関連項目
- デザインの作成
- SPS ファイル: コンテンツ

5.12.1 レイアウトコンテナー

レイアナーコンテナーコントのような特徴を備えています:

- ドキュメント内、つまりテンプレート内部に <u>挿入</u> することができます。また、ドキュメントデザインを含んでいるコンテナー内に 挿入することもできます。
- そのセクションに対して定義されたページの寸法に合わせる(レイアウトコンテナーの「用紙サイズに自動調整」プロパティ)ことができるほか、寸法を指定することができます。詳細については、レイアサーエテナーのサイズセグシン参照(たさい。
- <u>レイアナ・グトボ</u>と<u>ズーム機能</u>により、レイアナコンテナー内にあるオブジェナトの位置を簡単に指定することができます。
- ・ 境界線、背景色、フォントのプロパティといった スタイルプロ ディを、全てのエンテナーコ対して指定することができます。
- レイアナボックスやラインを含めることはできますが、その他のデザイン要素を含めることはできませんく全てのデザイン要素はレイアウトボックス内部(電配置する必要があります)。
- <u>青写真</u>を含めることで、テンプレートを作成するナータのガイドとして使用することができます。 青写真により表示されたイメージに、正確にマッチするようにデザインの配置を行うことができます。

164 SPS コンテンツ レイアウト モジュール

メモ レイアウトコンテナーは、Enterprise エディションの Altova 製品における Authentic View でしかサポート されません。

レイアナーを挿入する

レイアナーンテナーを挿入するける。 デザイン要素の挿入 ツール・ にあるレイアウトコンテナーの 挿入 アイエをクトックした後に、レイアナニンテナーを挿入する位置をクトックします。 レイアナニンテナーを用紙サイズに自動調整するかを決定するためのダイアログが表示されます。 はいをクトックすることで、そのドキュメトセグションのページレイアナープロ ティーて定義されたページの寸法と同じ大きさがレイアウトエンテナープロして与えられます。 いいえをクトックした場合、デフォルトサイズの3.5 in x 5.0 in のレイアナニンテナーが作成されます。

レイアウトコンテナーは SPS の作成時にも作成することができます。

レイアウトコンテナーのサイズ

レイアナンテナーの大きさは、以下の2つのプロ ティにより決定されます:

• 用紙サイズに自動調整プロパティ(プロパティサイドバー、以下のスクリーンショットを参照): yes を選択することで、ドキュメントセクション内にあるページに対して使用される寸法がレイアウトコンテナーでも使用されます。no を選択すると、大きさをカスタマイズすることができるレイアウトコンテナーが作成されます。



スタイルサイドバーのレイアウトから選択することができる詳細グループの height ならびに width プロパティにより、レイアウトコンテナーの大きさを指定することができます。レイアウトコンテナーの右ならびに下余白をドラッグすることでも、コンテナーの大きさを直接修正することができます。height ならびに width プロパティの値は、用紙サイズに自動調整プロパティの値が no になっている時にだけ有効になる点に注意してください。

レイアウトコンテナー グトド

レイアナコンテナー内にグランドを表示させることで、アイテム間の間隔指定に使用することができます。以下の設定により、グランドを使用することができます:

- グリッドを表示/隠す:デザイン要素の挿入ツールバーにて、グリッドの表示を有効/無効にすることができます。
- グリッドサイズ: オプションダイアログのデザインタブにて、水平ならびに垂直方向の長さを指定することができます。非常に大きな値が入力された場合、グリッドが正確に表示されない場合もあることに注意してください。
- グリッドへのスナップ:デザイン要素の挿入ツールバーにある切り替えコマンドにより、グリッドへのスナップ機能を有効または無効にすることができます。グリッドへのスナップ機能が有効になっている場合、レイアウトボックスの左上と、レイアウトボックスの終点がグリッドの線と点に対して配置されます。

ズーム機能

表示を拡大することで、オブジェクトの配置をより正確に行うことができます。(標準ツール・インある)ズームコンポポックスを使用するか、Ctrlキーを押下しながらマウスホイールを使用することで、拡大率を変更することができます。

レイアウトコンテナー スタイルプロノティ

レイアウトコンテナーに適用することができるスタイルコよ 以下の2種類があります:

- レイアウトコンテナーだけに適用され、承継されないもの(例: border や background-color プロパティ)。
- レイアウトコンテナー内にあるレイアウトボックスに継承されるもの(例: フォントプロパティ)。



レイアナコンテナーのスタイルプロ ティは スタイルサイドバー こあるレイアナ コンテナー こてセナ されます(上のスクナーシショナを参照)。

レイアウトコンテナーのコンテンツ

レイアナコンテナー内に含めることができるデザインアイテムは、レイアナ・ボックスとライノコ限られます。その他に、(デザイン要素ではない)青 写真をデザインのガイドとしてレイアナコンテナー内に配置することができます。その他のデザイン要素は全て、レイアナ・ボックス内に配置する 必要があります。

青写真

SPS の作成中に、デザインのガイドと、てレイアナコンテナー内に青写真を配置することができます。青写真には、レイアナコンテナーの大きさに等し、大きさのイメージファイルが使用されます。青写真イメージがレイアナコンテナーよりにから、デザイン内で指定された位置をオフセナとすることができます(以下のスクリーンショナにある青写真グループプロ、ティを参照ください)。青写真を使用することで、SPS デザインを青写真デザインで表現することできます。この機能により、青写真で表示されているとおりにデザイン要素を配置することが可能になります。青写真はデザインビュー・ゼガに表示され、出力ビューで表示されることはありません。

(プロ・ティサイド・ドーの)レイアナトコンテナープロ・ティーある青写真のプロ・ティグループから、青写真のプロ・ティヘアクセスすることができます (以下のスグノーシショナを参照)。

166 SPS コンテンツ レイアウト モジュール



レイアナーコンテナーコ表示される青写真の透明度を指定することで、デザインの邪魔ロコンジストレラにすることができます。 青写真のイメージは イメージを表示 プログライニて、必要口応じて表示/非表示を切り替えることができます。

- □ 関連項目
- レイアナトボックス
- ラシ

5.12.2 レイアウトボックス

レイアナト内にある全てのデザイン要素(静的テキストやスキーマノード、自動計算、イメージ、リストなど)は、レイアナトボックス内に配置する必要があます。デザイン要素を含んだシイアナ・ボックスは、必要に応じてレイアナトコンテナーに配置されます。デザイン要素をレイアナトコンテナーに直接配置することはできず、レイアナトボックス内部に配置する必要があります。

このセクションでは以下のサブセクションにより、レイアウトボックスの使用方法について説明します:

- レイアナボックスの挿入
- <u>レイアウトボックスの選</u>択と移動
- レイアウトボックスのサイズを修正
- レイアナ・ボックス スタイルプロ ティの定義
- レイアナボックスにコンテンツを挿入
- レイアナトボックスの表示順序

レイアウトボックスの挿入

レイアナボックスは<u>レイアナコンテナー</u>の中に対け挿入することができます。レイアナボックスを追加するコは<u>デザイン要素の挿入</u>ツールドースあるレイアナボックスの追加アイコンをケトックして、レイアナボックスを挿入するレイアナコンテナーの場所をケトックします。 ケトックされた

場所が主上端となるようなかたちでレイアナバックスが挿入されます。レイアナバックスは透明で境界線を持たず、デフォルトのテキストが表示されます。

レイアウトボックスの選択と移動

レイアナボックスの左まけまと側にある境界線までカーノルを移動させ、カーノルが移動状態になって土状態でクトックすることで、レイアナボックスを選択することができます。マウスが文を押下しながら、レイアナボックスをレイアナコンテナー内にある他の場所へ移動することもできます。レイアナボックスが選択された状態で矢印キーを押下することでも、レイアナボックスを移動することができます。レイアナボックスが選択されると、関連するプロ・ティやスタイルがそれぞれのサイド、ドーに表示されます。

レイアウトボックスのサイズを修正

各レイアナボックスコはサイズの自動調整とら名前のプロ、ティか含まれて、ます(以下のスクノーンショナを参照)。このプロ、ティの値を yes にセナすることで、挿入された任意の、マークアップを含む)コンテンソに合わせるかたちでレイアナボックスの大きさか自動的に調整されます。自動調整の値がno にセナされた場合、コンテンツが挿入されてもレイアナボックスの大きさか自動的に変更されることはありません。

右および下側の境界線をドラッグすることで、レイアナボックスの大きさを手動で変更することができます。矢印キーを使用することでもレイアウトボックスの大きさを変更することができます。まず<u>レイアナボックスを選択</u>し、Shift キーを押下しながら、矢印キーを押下することで、レイアナボックスを目的の大きさこ変更することができます。



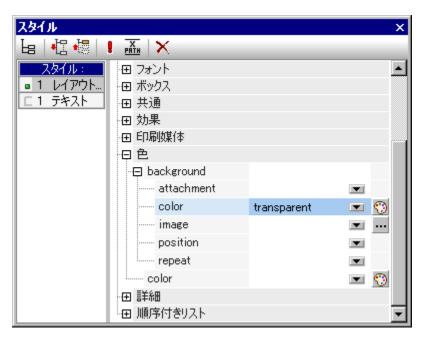
追加高さど追加幅プロ ティにより、自動調整にて決定される大きさを広げることができます。 追加幅と追加高さま、レイアナボックスの大きさが手動で変更された際に、新たて計算されます。 反対に、これらの値を変更することでも、レイアナボックスの大きさを変更することができます。

メモ レイアウトボックスでは Enter キーを押下することで改行が入力されます。改行を含まないコンテンツが追加されると、その行の長さがレイアウトボックスの幅とともに大きくなり、追加幅の値が影響されることになります。

レイアウトボックス スタイルプロバティ

レイアナボックスのスタイルプロ・ティは、スタイルサイド・ドーンあるレイアナボックスこでセナすることができます(以下のスクリーンショナを参照)。スタイルはレイアナボックスの選択時に表示され、編集することができます。

168 SPS コンテンツ レイアウト モジュール

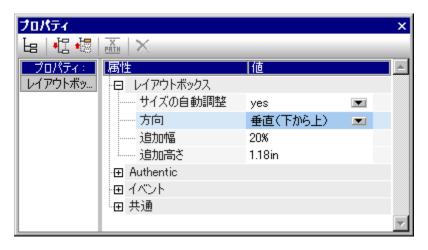


メモ (透明を意味する)transparent という値を、background-color のプロパティコンボボックスから選択することができます(カラーパレットから選択することはできません)。以下で説明されるように、レイアウトボックスが重なり合う際にこの値を使用することができます。

レイアウトボックスにコンテンツを挿入

レイアナボックスコは SPS と同様の方法で任意のデザイン要素を挿入することができます。 レイアナコンテナー おけは Layout Line はレイアナ・ボックス コ挿入できない ことづ注意してくたさい。 留意すべき点を以下に示します:

- コンテキストノードが必要なデザイン要素が挿入された場合、カレントノードがコンテキストノードとして使用されます。レイアウトモジュールの作成に使用されたノードがカレントノードになります。
- レイアウトボックス内のテキストコンテンツは、時計回りまたは反時計回りに90度回転させることができ、上から下、または下から上の方向ヘテキストを表示することができます。デザインにて回転させるテキストを選択し、プロパティサイドバーにあるレイアウトボックスグループを選択します。方向プロパティにて目的の値を選択してください。

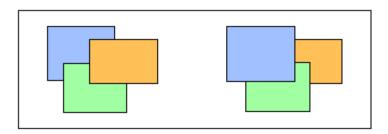


以下の点に注意してくたさい

- 出力にて文字は回転されますが、デザインではそのように表示されません。
- このプログライはテーブルセル内のテキストにも適用することができます。

レイアナボックスの表示順序

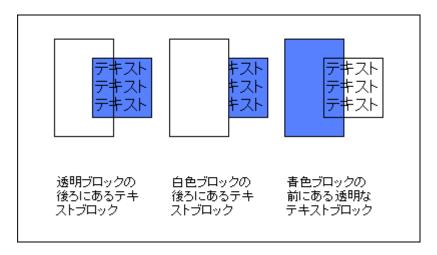
レイアナボックスは重ね合わせて表示することができます。あるレイアナボックスが別のレイアナボックスの上に配置されると、レイアナボック スが透明になっていない場合、覆われナシイアナボックスが別のシイアナボックスによい隠れてしまいます。 レイアナボックスを重ね合わせるこ とで最前面にあるレイアナボックスだけが完全に表示され、その他のレイアナボックスは部分的にしか表示されないか、全く表示されなくなり ます。



レイアナボックスのコンテキストメニューにある順序メニューコマンドを使用することで、選択されナシィアナバックスを前面まけば背面へ移動 することができます。これらのコマイを使用することで、レイアナボックスを(i) 最も近いボックスニ対して移動(前面へ移動ならびに背面 へ移動コマイ)、ませは(ii) 全てのドックスマ対して移動(最前面へ移動ならびに最背面へ移動コマイ)することができます。上の スクノーンショナでは以下の順序によりボックスが配置されています。

左側の図:オレンジ、緑、青 右側の図:青、緑、オレンジ

透明(transparent)な背景(レイアナボックスニデンオルで使用される背景)をともなうレイアナボックスよ、特に複数のボックスか透明と なっており、更に境界線を持たむ、際に、他のアイテムに対して順序が変更されているいように見えることもあります。以下のスクレーンショナ では、透明な背景が重ね合わせでどのように表示されるか示されます。



火モ レイアナラインを、レイアナボックスと一緒に追加することで、これらラインの順序を他のアイテムに対して移動することができます。

- 関連項目
- レイアナンテナー

170 SPS コンテンツ レイアウト モジュール

ラル

5.12.3 ライン

ラインは、レイアナボックスではなく、<u>レイアナコンテナー内に挿入</u>した後に、レイアナコンテナー内で<u>選択、移動や大きさの変更</u>を行い、 プロンティク割り当てや、レイアナボックスならびごラインからなるレイアナアイテムの表示順序を変更 することができます。

ラインを挿入する

以下の操作により、ラインをレイアウトコンテナーに追加することができます:

- 1. デザイン要素の挿入ツールバーにあるラインの挿入アイコンをクリックします。
- 2. 追加するラインの起点となるレイアウトコンテナー内の位置をクリックします
- 3. マウスボタンを離すこと無く、起点から終点までマウスをドラッグします。終点にてマウスボタンを離します。

黒色のラインが挿入されます。両端コは起点と終点を表すためのドットが表示されます。

ラインを選択、移動、大きさを変更する

メインウンドウでは、以下のドラッグアンドドロップ機能を使用することができます:

- ラインを選択するには、ラインの一部をクリックします(ラインの上部にカーソルがある場合、マウスポインターが 移動状態になります)。ラインを選択した後には、プロパティがプロパティサイドバーに表示され、編集を開始する ことができます(以下を参照)。
- ラインを移動するには、ラインが選択されている状態で、目的の場所へドラッグします。ラインが選択されている状態で矢印キーを押下することでも、上下左右へラインを移動することができます。
- ラインの大きさや方向をグランイカルに変更するコよ、起点まりは終点を選択し、大きさや方向を変更するように移動してください。Shift キーを押下しながら矢印キーを使用することでも、ラインの大きさや方向を変更することができます(矢印キーにより、終点が移動されます)。

ラインプロパティ

ラインが選択されている場合、そのプロ、ティがプロ、ティサイドバーに表示され、プロ、ティの編集を行うことができます(以下のスクリーシショナを参照)。ラインを右クトックして、ラインプロパティを編集を選択することでも、ラインのプロ、ティを編集することができます。

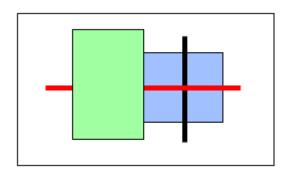


プロノティサイドノデーでは、以下のラインプロノティを編集することができます:

- color: ラインの色を指定します。デフォルトでは黒色となっています。
- 大きさと位置: x-軸(水平)ならびに y-軸(垂直)の位置によりラインの起点と終点の位置を指定することができます。レイアウトコンテナーの左上が原点(x=0, y=0)となります。
- width: ラインの太さを指定します。

ラインと表示順序

重ね合わせばっているレイアナボックスやその他のラインと、あるラインが同じ位置にある場合、そのラインのエレテキストメニューに表示される順序メニューコマイドを使用することで、前面や背面へ移動することができます。これらのコマイドを使用することで、ラインを(i) 最も近いアイテムコメして移動(前面へ移動ならびに背面へ移動コマイド)、まだは(ii) 全てのアイテムコメして移動(最前面へ移動ならびに最背面へ移動コマイド)することができます。



上のスクノーンショナでは、以下の順序で重ね合わせが表示されます:緑色のドックス、赤色のライン、黒色のライン、青色のドックス

□ 関連項目

- レイアナンテナー
- レイアナトボックス

172 SPS コンテンツ 変更機能

5.13 変更機能

テンプレートやテンプレートのエレテンツが選択されている状態で使用することのできる変更機能により、(i) そのテンプレートが適用されるノード、まけは(ii) そのソードがデザインでどのように作成されるのかを変更することができます。

変更機能により変更することができるもの

ノードやそのエンテングを変更することができます。以下に示されるスクリージンコナのうち、左側のイメージではノードが選択されており、右側のイメージでは、ノード内部にあるエンテンツが選択されています。

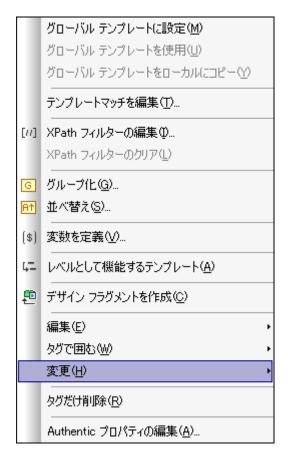




上にあるスクリーンショナでは、n1:Name 要素が(コンテンツ)として作成されており、ノードのコンテンツが(コンテンツ)プレースポルダにより示されます。他にも、別種類のコンテンツとしてノードを作成することもできます(例: 入力フィー・ルドヤコンボドックス)。その他の種類のコンテンツを選択することもできます。

変更コマイ

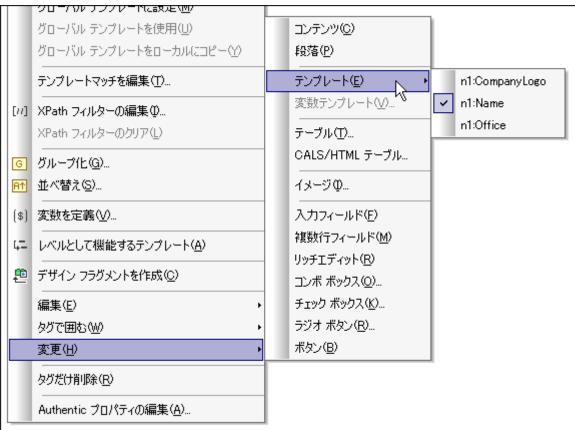
選択されたものを右クトグして、コンテキストメニューから変更コマンドヘアクセスすることができます(以下のスクリーンショナを参照)。



SPS コンテンツ 変更機能 173

テンプレートマッチを編集する

テンプレートが選択されている場合、そのテンプレートが適用されるノードを変更することができます。例えば、スキーマースで要素の名前が変更された際にこの機能を利用することができます。変更コマンドへマウスオーバーして、サブメニューからテンプレートを選択すると、選択されたノードの兄弟要素として挿入することができる全ノードのノストが表示されます。ノードをクトックすることで、そのノードに対してテンプレートが適用されます。



ーテンプレートの記述とマッチしてい、ロンテンツが選択されたノードに含まれている場合、構造的な矛盾が発生します。このような矛盾が発生すると、ノードタグに赤色の横線が表示され、エラーであることが示されます。

テンプレートマッチを変更することで、(ノードではなく)変数テンプレートをマッチさせることもできます。

ノードのコンテンツ型を変更する

テンプレートやそのエンテンソが選択されている場合、ノードが作成されたコンテンソの型を変更することができます。コンテキストメニューの変更コマンドにマウスオーバーすると、選択されたノードを変更することができるコンテンソの型がサブメニューにて表示されます(以下のスクリージンョナを参照)。

174 SPS コンテンツ 変更機能

コンテンツ(©)
---ブル(T)...
イメージ Φ...
入力フィールド(E)
複数行フィールド(M)
リッチエディット(R)
コンボ ボックス(W)...
ラジオ ボタン(R)...
ボタン(B)

上のスクノーンショナは、コンポドックスが選択されている状態で表示されたものです。

- □ 関連項目
- SPS ファイルゴンテンツ
- <u>テンプレートとデザインフラグメント</u>

SPS 構造 175

6 SPS 構造

SPS ドキュメントの構造は、入力ならびに出力をベースとしており、以下の項目により制御されます。

- スキーマノース
- モジュールSPS
- デンプレートとデザインフラグメント

入力ベースの構造: スキーマとモジュール SPS ファイル

入力ベースというのは、SPS のノーススキーマにより入力ドキュメトの構造が指定されることで、この構造がSPS ドキュメントの構造といます。例えば、ソーススキーマにて Office 要素のシーケンスが定義されている場合、Office 要素のテンプレートを SPS に含めることができます。 処理を行う際に、このテンプレートがソースデータドキュメント内にある各 Office 要素に対して適用されます。

ソースドキュメトの構造によりSPS ファイルのデザインが決定される例は、テーブルでも見ることができます。Office 要素に複数のPerson 子要素が含まれており、各 Person 要素にはName や Address、Telephone といた子要素のセナか含まれている場合、Person 要素に対してテーブルのテンプレートを作成することがで、各 Person 要素がテーブルにおける行として表示されます(以下のスクレーシンコケを参照)。テーブルのカラムは、Person の詳細(Person 要素の子要素)といます。

First	Last	Title (sorted by)
Loby	Matise	Accounting Manager
Frank	Further	Accounts Receivable
Vernon	Callaby	Office Manager

Person 要素の構造と複数のPerson 要素が兄弟ノードであるために、このようなテンプレートが可能しないます。テンプレートではPerson 要素に対してデザインされた行がいつ定義されており、この行によりソースドキュメント内にある各 Person 要素が処理され、Person 要素の数だけ新たな行が作成され、子要素がデーブルのカラムとして表示されます。

各種スキーマノースの使用方法についてはスキーマノースのセクションして記述されます。

更に、StyleVisionでは作成したSPSをモジュールとして別のSPS内部で再利用することができます。この機能により、モジュールを別の構造内部に含めることができるようになります。モジュール内部に含まれているスキーマ構造は、モジュールを含んでいるSPSのスキーマ構造に対して合致するものでなければなりません。モジュールSPSの使用方法については、モジュールSPSのセグションを参照ください。

出力べースの構造: テンプレートとデザインフラグメント

スキーマノースにより入力デーダンキュメトの構造が与えられる一方で、出力ドキュメトのデザインは SPS ドキュメトにより決定されます。このデザインは、メインテンプレートと呼ばれるドキュメントテンプレートに収められます。メインテンプレートには通常複数のエンポーネントテンプレートが含まれ、グローマリテンプレートに対する参照が行われます。テンプレートに関する説明は、デンプレートとデザインアグメントのセクションにて記述されます。

複数のテンプレートをモジュールとて使用する機能はデザインフラグメトと呼ばれる StyleVision の機能は拡張され、特定の処理を再利用可能なドキュメントフラグメントに対して割り当てることができます。グロー・ジルテンプレートと違い、デザインフラグメントは(i) 複数のテンプレートが構成することが可能で、(ii) 異なる処理が含まれる同一のコレテンンを別々のデザインフラグメントにて作成することができ、状況に応じて、必要な処理を行うことが可能はなります。例えば、状況に応じて Email ノードをレクとして表示し、太字の赤色で Email 要素を表示することが可能はなります。デザインフラグメントを2つ用意することで、それぞれ目的の処理を行うことができます。

デザインフラグメントに関する詳細は、デザインフラグメントのセグションにて記述されます。

- **関連項目**
- 使用方法の概要

SPS 構造 スキーマソース 177

スキーマソース 6.1

スキーマノースはデザインを行う際の出発点ともなるもので(i)スキーマの選択と(ii)スキーマにて選択したルト要素によりデザインの構造 は影響を受けることされます。

スキーマの選択

以下の操作により、新たな SPS ファイルコオするスキーマの選択を行うことができます:

- 1. メニューオプションから「ファイル | 新規作成」を選択し、サブメニューにある(新規作成(空)を除く)オプション
- を選択して、スキーマソースを直接選択します。 2. メニューオプションから「ファイル | 新規作成」を選択し、サブメニューから「新規作成(空)」を選択します。新規 SPS が作成され、GUI に表示されます。デザルで乗サルバーにて新しいソースの追加コマンドを選択します。 す。スキーマとして追加することのできるものが表示されます(以下のスクリーンショットを参照)。このメニューに表示されている各コマンドについては、このセクションのサブセクションにて記述されます。.

+🚰 XML スキーマ/DTD/XML を追加(S)...

+ 🚨 ユーザー定義スキーマを追加(U)

スキーマノースは、ファイル、まけまユーザー定義のスキーマから選択することができます。 グロー・ シレテンプレートを使用するか、まナーグロー・シレ テンプレートとして使用する要素がネキーマにグローハリ要素として定義されているかどうかがここでは重要になります。ファイルからDTD を 追加する場合、DTD 内で定義された要素は全てグローノい要素となることに注意してください、ファイルからXML スキーマを追加する場 合、どの要素がジロー・ い要素として定義されているか、そしてスキーマニ対して変更が必要な場合には、お使いのXML 環境にてそれがで きるかどうか調べるようはしてくたさい。

名前空間を SPS へ追加し、SPS から XSLT スタイルシートを生成するには、SPS がベースとしている XML 旡 スキーマのトップレベル schema 要素に対して名前空間を追加する必要があります。

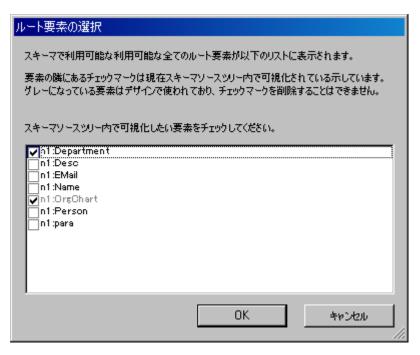
ルート要素

スキーマソノーニ複数のグロー・ゾル要素 か含まれている場合、複数のルート要素 (ドキュメント要素) をデザインで使用するために選択する ことができます。この機能により、複数のドキュメント要素にマッチするテンプレートをSPS デザイノに含めることができます。

UniversalSchema.xsd とら名前のスキーマをベースしたUniversalSPS.sps にてElement-A とElement-B とら2つのレト要素が与 えられている場合、Element-A をデキュメート要素とするインスタンスデキュメートと、Element-B をデキュメート要素とする別のインスタンスデ キュメトを、単一のSPS から扱うことが可能しています。各 XML インスタンスコ対して関係するテンプレートが使用され、そうでないテンプ レートは使用されません。これは各 XML インスタンスドキュメントにおいて、そのドキュメント要素にマッチするテンプレートがSPS 内につし カ無いためです。例えばドキュメト要素の/Element-A は /Element-A を選択するテンプレートにマッチしますが /Element-B を選択 するものコはマッチしません。複数のグロー・い要素がスキーマイこて定義されている場合、それらのたのどれかをドキュメント要素としている XMLドキュメトも妥当なものとして認識されます(勿論、そのサブ構造もスキーオマ対して妥当でなければむりません)。

複数のルト要素 (<u>ドキュメト要素</u>) を使用するようにSPS をセナアップするコよ スキーマのルト要素エナリーの右側にある … ボ タンをクリックしてくたさい。以下に示されるダイアログが表示されます。

178 SPS 構造 スキーマソース



ダイアログコはスキーマ内部にある全てのグロー・い 要素が表示されます。ルト要素(ドキュメト要素)として使用することができるグロードル要素を選択し、OK をクトグしてください。選択された要素がルト要素リストとして表示され、ルトドキュメト要素として利用できるようにはよす。これらのドキュメト要素に対してテンプレートを作成することができるようはいます。これらのデンプレートは、ルト要素テンプレートの代わりとして使用することができます。XMLドキュメトがSPSとともに処理された場合、代替ルト要素テンプレートの方ちの1つがナーXMLドキュメトのルト(ドキュメト)要素-が使用されます。

Element-A をデキュメト要素とする XML ドキュメトが SPS によ処理されると、Element-A にマッチするルトテンプレートが使用され、SPS 内にあるその他全てのルト要素テンプレートは無視されること ています。 Element-B をデキュメト要素とするデキュメトが処理された場合、 Element-B にマッチするルトテンプレートが使用され、 SPS 内にあるその他全てのルト要素テンプレートは無視されることになります。この機能により、異なるルート(ドキュメト)要素を伴う複数の XML ドキュメトに対して同じ SPS を使用することができるようになります。

- 関連項目

スキーマソノーサイドバー

6.1.1 DTD と XML スキーマ

SPS はXML スキーマやDTD をベースとすることができるまか、。以下の操作により、XML スキーマやDTD をスキーマノースとして作成することができます:

- SPS を作成(「ファイル | 新規作成 | XML スキーマ/DTD/XML から新規作成」)することで、XML スキーマや DTD がスキーマソースとして直接作成されます。
- XML スキーマや DTD を、空の SPS

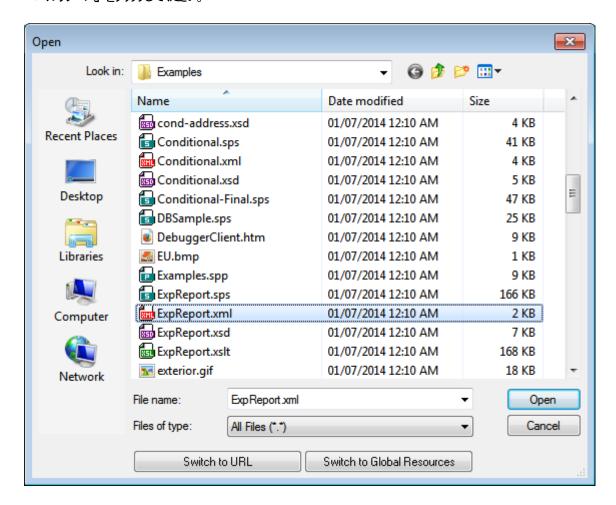
コマバを選択することにより、XML スキーマやDTD を参照するためのダイアログが表示されます。スキーマが妥当な場合、そのスキーマがスキーマノースとして作成され、スキーマソノーサイドバーにあるスキーマノースソノーに表示されます。 おこ XML ファイルを選択することができます。 XML スキーマ (.xsd) おさも DTD ファイルが XML ファイル 関連付けられている場合、 XML スキーマや DTD ファイルが ノーススキー

SPS 構造 スキーマソース 179

マとしてロードされ、XML ファイルが作業用 XML ファイルとしてロードされます。XML ファイルは関連付けられているスキーマが存在したい場合、XML ファイルの構造とコンテンから XML スキーマを生成するか、既存のスキーマを選択するように促されます。スキーマの生成を選択した場合、生成されたスキーマがソーススキーマとしてロードされ、XML ファイルが作業用 XML ファイルとしてロードされます。

▼ URL とグロー・ゾレリノースを使用してファイルを選択ませば保存する

「ファイルを開く」および「ファイルの保存」ダイアログでは、URL まけまグロー・ジリソースを使用して必要なファイルを選択まけまファイルを保存することができます(アのスグリーンショナ・参照)。選択プロセスに移動するかがに「URL に切り替える」まけま「グロー・ジリソノース」をクリックしてください。

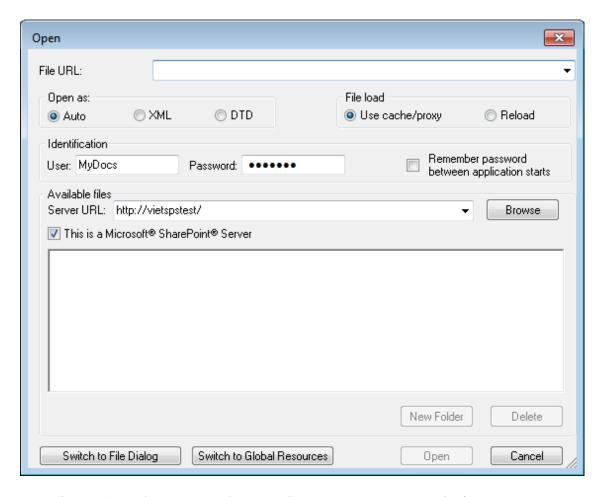


URL を使用してファイルを切り替える

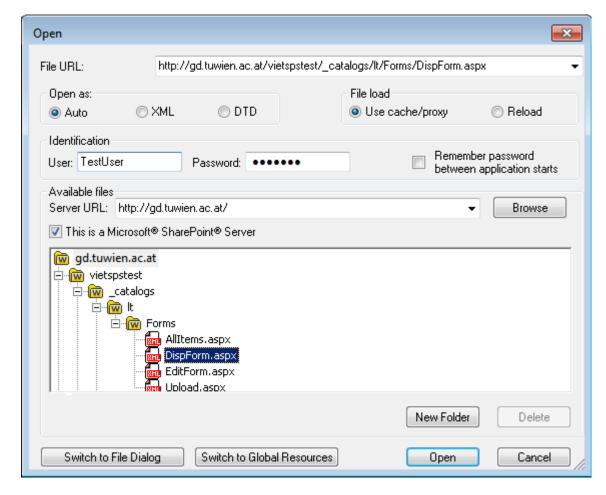
URL から閉くまけは保存するファイルを選択するコお以下を行ってくたさい

1. 「URL に切り替える」コマンドをクトックします。これにより、開くまは、保存ダイアログのURL モードにスイッチされます。(ア のスクリーンショットは開くダイアログを表示しています)。

180 SPS 構造 スキーマソース



- 2. アクセスするURL をサーバーURL フィールドに入力します(上のスクリーンショナ)。サードがMicrosoft® SharePoint® Server の場合 Microsoft® SharePoint® Server チェックボックスをチェックしてください。この種類のサーバー上にあるファイルと作業する場合、下のMicrosoft® SharePoint® Server メモを参照してくたさい。
- 3. サーバーが、ペワードにより保護されている場合、ユーザーとノスワードフィーリいに入力してくたさい。
- 4. 「参照」をクトックして、サーバーのディレクトリ構造をナビゲートします。
- 5. フォルダーソノー内でロードするファイルを参照してクトックします。



ファイルURL がファイルURL フィールドに表示されます(*上のスクリーンショナト参照*)。「開く」 まりは「保存」が文を使用できるようしなります。

6. 「開く」をクトックしてファイルをロードするか、「保存」をクトックして保存します。

以下の点に注意してくださ!

- WebDAV をサポートするサーバーとMicrosoft SharePoint Servers のみで参照機能を使用することができます。サポートされるプロトコールはFTP、HTTP およびHTTPS です。
- ファイルを開く際のロード処理を更に管理するコよ、ローカルのキャシシュ おコ スファイルが以前 コードされている場合スピードを向上する プロキシ のサーバーからファイルをロードします。(ファイルが事前 コロードされること ことが 便に処理スピードを向上することができます)。 おっ、電子出版 おコ まデータベースシステム などの作業中のファイルを再ロードして、「再ロード」 オ プションを選択します

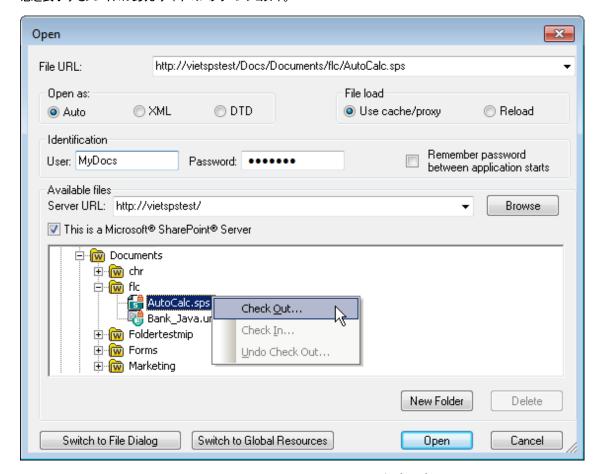
▼ Microsoft® SharePoint® Server メモ

Microsoft® SharePoint® Servers に関する以下の点に注意してくたさい

「利用可能なファイル」ペインに表示されるディレクト・構造では、ファイルアイコンコはファイルのチェックイン/チェックアナの状

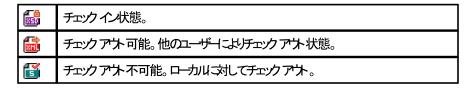
(C) 2015-2021 Altova GmbH

態を表示するシンボルがまます(下のスクリーンショント)。



ファイルを右クトックすると、ファイルで使用することのできるコンテキストメニューがポップアップします(上のスクリーンショント)。

異なるファイルアイコンが以下に表示されています:



- ファイルをチェッケアナトした後、Altova アプケーションを使用して編集することができ、「ファイル 保存 (Ctrl+S)」を使用して保存することができます。
- 編集されナンアイルを「URL を開く」ダイアログのエンテキストメニューからチェックインすることができます(上のスクリーンショット参照) まけま、アプリケーションのメインウィンドウのファイルタブを右クトックすることによりポップアップするエンテキストメニューを使用して(下のスクリーンショット)。



- ファイルが他のユーザーによりチェックアナトされている場合、チェックアナトすることができません。
- 他のユーザーによりファイルがチェックアナーされている場合、チェックアナーを行うことはできません。これによりファイルを変更す

ることなくサーバーに戻すことができます。

• 変更することない「ファイルがサードーへ戻されます。ある Altova 製品でファイルをチェックアナーした場合、他の Altova 製品で同じファイルをチェックアナーすることはできません。この時点で使用できる Microsoft® SharePoint® Server をサポートする Altova アプケーションのコマンドは、以下の通りです:「チェックイン」および「チェックアナトを元に戻す」です。

XML スキーマはおお anyType データ型

XML スキーマ内にある要素に、XML スキーマのanyType データ型が割り当てられている。まけまどのデータ型も割り当てられていない場合、そのスキーマンカる全ての要素が使用可能な子要素としてスキーマソナーに表示されます。例えば、email という要素に対してデータ型が割り当てられていない場合、person、address、city、tel という全てのグロージル要素が、使用可能な子要素としてスキーマソナーに表示されます。例えばxs:string というデータ型を email 要素に割り当てることで、このような振る舞いを回避することができます。

- □ 関連項目
- スキーマソノーサイドバー

6.1.2 ユーザー定義スキーマ

スキーマソノーサイドバーでは、ユーザー定義スキーマを素早く作成することができます。ベースとなるスキーマを持たないXMLドキュメントに対して SPS を作成する際にこの機能を利用することができます。

以下の操作により、スキーマツリーサイドバーにてユーザー定義スキーマを追加、作成することができます:

1. メニューオプションから「ファイル | 新規作成 | 新規作成 (空)」を選択します。<u>デザル概要サイドバー</u>にて、 (ソーズ行以下にある)新しいソースの追加コマンドをクリックし、ユーザー定義スキーマを選択します(以下 のスクリーンショットを参照)。

+□ XML スキーマ/DTD/XML を追加(S)...

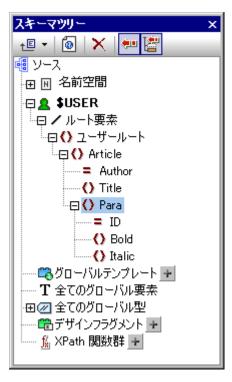
新たなスキーマが作成され、\$USER という名前のパラメーターにより表されます(以下のスクリーンショットを参照)。



- 2. ルート要素ツリーには、ユーザールートという名前の ルト要素(ドキュメト要素) が表示されます。
- 3. ユーザールートをダブルクリックすることで、スキーマがマッチするように XML ドキュメントの <u>ドキュメント要素</u> の名前を変更することができます。
- 4. 子要素や属性をドキュメント要素に割り当てるには、ドキュメント要素(ユーザールート)を選択し、 スキーマンリーサイドでのソール・インある新規要素の追加アインします。 またはアイコンの右側にある矢印をクリックして、属性を追加エントリーをクリックします。 右クリックして表示されるコンテキストメニューから目的のコマンドを選択することもできます。 要素が選択されている状態で要素の追加や挿入を行うと、選択された要素の前または後に、新たな要素が兄弟要素として追加されます。 子要素や子属性を追加することもでき、属性が選択されている状態から、その属性の前または後に他の属性を挿入/追加することができます。 新たな要素または属性をツリーに追加した後には、キーボードから名前を入力してください。 目的の場所へノードをドラッグすることもできます(次のステップにある記述を参照)。 以下のスクリーンショットでは、 Article 要素がドキュメント要素となります。 Title、 Para、 Bold、 そして Italic 要素と ID ならびに Author 属性が Article の子として追加されています。



5. Bold ならびに Italic 要素と ID 属性を Para の子として移動するには、個々の要素や属性を選択し、Para 要素へドラッグします。マウスポインターが矢印ポインター → に変わったところで、ノードをドロップします。ノードが Para の子として作成されます(以下のスクリーンショットを参照)。



6. ドキュメント要素以外の要素が選択されている場合、新たな要素や属性を追加することにより、選択されている要素と同じ階層に新たなノードが追加されます。ノード(要素や属性)を要素ノードへドラッグすることで、そのノードが要素ノードの「子」として作成されます。

ノード名の編集やノードを削除する

要素や属性の名前を編集するコよノードの名前をダブルクトックして名前の編集を行います。ノードを削除するコよ目的のノードが選択されている状態でソール・ドースある削除アイコンとをクトックするか、コンテキストメニューから削除を選択します。

□ 関連項目

- スキーマソノー サイドバー
- デザイン概要サイドバー

6.2 複数のソースから XML データのマージ

XSLT 2.0 ませま3.0 SPS で使用されるXSLT の デジョンとすることで、複数のノースXML ファイルから得られたXML データを組み合わせることができます。

通常、データのマージはID のような共通したデータをベースに行われます。例えば、従業員 ID により識別される社内の従業員の詳細データ(例: 従業員の詳細情報、給与、出社日)が、複数のXML ファイルにまたがって保管されている場合、これら複数のファイル内に保管されているデータを、従業員 ID をキーとしてマージし、出力ドキュメトにて表示することができます。

メモ Enterprise Edition では、複数のスキーマソースを含めることができ、パラメーター名を使うことで、対応するスキーマから得られた XML ノードを選択することができます(以下のサンプルを参照ください)。 Professional ならびに Basic Edition では、XPath 2.0 の doc() 関数を使用することで、目的の XML ファイルと、そのファイル内にある XML ノードを選択することができます。XPath 2.0 の doc() 関数により、外部 XML ドキュメントのドキュメントルートへアクセスすることが可能になり、外部 XML ドキュメント内にあるノードコンテンツを出力に挿入することが可能になります。そのため、doc() 関数を使用する 自動計算を使うことでも、XML データをマージすることができます(以下のサンプルを参照くがごい)。

サンプル

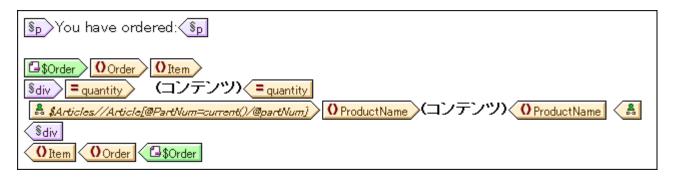
(マイ)ドキュメトフォルダーC:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2021\StyleVisionExamples は、サンプルSPS ファイル
(MergeData_2_Files.sps) が含まれており、異なるノース XML ファイルのマージの方法が説明されて、ます。SPS は、架空の主文(下にリストされる MergeOrder.xml) からデータを選択します。

このファイルの/Order/Item/@partNum 属性の値は他のファイルMergeArticles.xml に保管されているカタログから製品を主文するために使用されます(アのリストを参照してください)。

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:noNamespaceSchemaLocation="MergeArticles.xsd">
  <Article PartNum="833-AA">
     <ProductName>Lapis necklace
     <Price>99.95</Price>
  </Article>
  <Article PartNum="748-OT">
     <ProductName>Diamond heart
     <Price>248.90</Price>
  </Article>
  <Article PartNum="783-KL">
     <ProductName>Uncut diamond
     <Price>79.90</Price>
  </Article>
  <Article PartNum="238-KK">
     <ProductName>Amber ring
     <Price>89.90</Price>
```

```
</Article>
  <Article PartNum="229-OB">
     <ProductName>Pearl necklace
     <Price>4879.00</Price>
  </Article>
  <Article PartNum="128-UL">
     <ProductName>Jade earring
     <Price>179.90</Price>
</Article>
</Articles>
```

/Order/Item テンプレート内にユーザー定義テンプレート をセナアップすることで、注文されたパーツの番号によりMergeArticles.xml ファイ ル内の対応するArticle 要素が選択され、データのマージが行われます(以下のスクリーンショナを参照)。(/Order/Item コンテキスト内に ある)XPath 条件式は\$Articles//Article[@PartNum=current()/@partNum] といます。



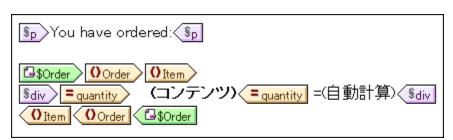
このテンプレートにより、以下のスクリーンショットに示されるような出力が表示されます。

You have ordered: 3 Amber ring 1 Diamond heart 1 Pearl necklace

2 Lapis necklace

各アイテムの注文数 (quantity) がMergeOrder.xml から得られているのつ対して、注文されたものの名前はMergeArticles.xml ファイルか ら得られていることで注目してください。ProductName ノードが/Articles/Article テンプレートのコンテキスト内にて選択されている点にも注 目してください。

自動計算を使用することには、上で得られたものと同じ結果を得ることができます(以下のスクリーンショナを参照)。スキーマソノーウィ・ドウ から quantity 属性をドラッグすることで、コンテンツとして作成します。その後、以下のスクノーンショナに示される様に自動計算を追加して、 以下にあるXPath 条件式をその自動計算に与えます。



自動計算の XPath 条件式により、他のスキーマソースのパラメーターや doc() 関数を使用することで、目的のノードを選択することができます:

\$Articles//Article[@PartNum=current()/@partNum]/製品Name

なな

doc('MergeArticles.xml')//Article[@PartNum=current()/@partNum]/製品Name

上にある最初のXPath 条件式がEnterprise Edition でのみ使用できる別のXML スキーで参照しているのご対して、2番目の条件式では(Professional ならびにBasic Edition でも使用することのできる) XPath 2.0 のdoc() 関数を使用しています。

□ 関連項目

• 自動計算

6.3 モジュール SPS

SPS のグロー・ジレテンプレートやデザインフラグメント、JavaScript 関数、そしてページレイアナトアイテムを、他のSPS デザインこて使用することができます。この機能により、以下のようなことが可能になります:

- 1. 複数の SPS にあるグローバルテンプレートやその他のコンポーネントを再利用することができ、単一のソース編集や出力の一貫性を確保するために使用することができます。
- 2. SPS をモジュール化して、更に柔軟な構造を作成することが可能になります。

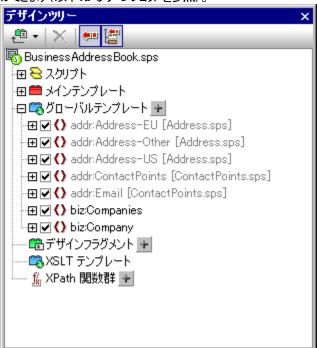
任意のSPS にて、複数のSPS をモジュールとして追加することができます。これらモジュール内にあるコンポーネナ(ませはオブジェケ)の
しくつかは SPS のインポート(や参照)を行う際に利用することができます。

利用が可能なモジュールオブジェクト

利用が可能なモジュールオブジェケ のセグションコよ SPS の様々なコンポーネントを SPS にインポートする際の制約と条件が記述されるまか、 SPS へのインポートに使用することができない コンポーネントが記されています。 追加されたモジューリスのたのにモジューリが含まれている場合、 これらモジューリは再帰的に SPS へ追加されます。 これにより、モジューリを広範囲なデザイン構造や複数のレベリハ展開することが可能になります。

モジュールSPS を作成する

モンュール化された SPS を作成するコよ、メイン SPS へ<u>目的の SPS をモンュールとて追加</u> します。 追加されたモンュール内部にある全ての Java Script 関数、グロー・ リテンプレート、デザインプラグメト、そして XPath 関数が、参照元の SPS にて利用可能とかます。 これらオブンエケトはデザインソノーこで表示され、チェックボックスへチェックを入れる/外すことで、個々のオブンエケトを有効/無効にすることができます(以下のアクノーンショナを参照)。



これらのオブジェケトはそれぞれのメカニズム:従うかかちで、参照元のSPS にて再利用することができます。グロー・バテンプレートは通常有効化するだけで参照元のSPS に適用することができます。デザインフラグメイは、デザインソノーから目的の場所へドラッグする必要があます。JavaScript 関数は、選択されたデザインエンポーネントに対してイベント・バンドラーとしてプロ・ティウン・ドウを介して割り当てることができます。(有効化された)XPath 関数は XPath 条件式にて使用することができます。

SPS モジュールの作成と使用方法については、モジュールSPS の作成のセグションを参照ください。

用語解説

SPS が他のモジュール内部で使用されると、後者のモジュールへの追加が行われます。2つのSPS は、それぞれ追加された SPS モジュールと、参照元 SPS モジュールと呼ばれます。SPS モジュールは追加されると、そのオブジェクトが参照元のSPS モジュールへ追加されます。これらのオブジェケトはモジュールオブジェクトと呼ばれ、以下の種類が存在します:グロー・シーテンプレート、デザインプラグメント、JavaScript 関数、ページレイアナアイテム。

□ 関連項目

- デザイン概要
- スキーマノース
- テンプレートとデザインフラグメント

6.3.1 利用可能なモジュールオブジェクト

このセグ・コンでは、追加されたSPS モジュールのうち参照元のSPS モジュールから利用することができるオブジェケトを紹介します。 SPS モジュールこれで、各オブジェケトをどのように利用することができるのか、おっその使用方法について説明します。モジュールSPS の作成方法については、次セグ・コンのモジュールSPS の作成を参照くたさい。セグ・コンの終わりには、参照元のSPS モジュールこで利用することのできない、オブジェケトが示されており、モジュールSPS かどのように動作するのか更に理解を深めることができます。

- 名前空間の宣言
- グローゾルテンプレート
- デザインフラグメント
- 追加されたモジュール
- スケルト
- CSS スタイル
- ページレイアサ
- 利用が不可能なモジュールオブジェクト

名前空間の宣言

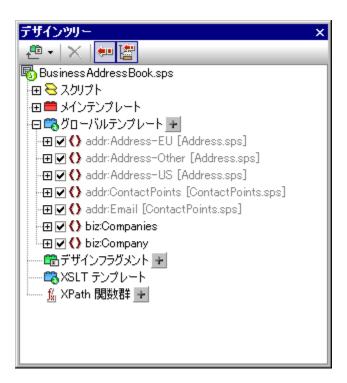
各 SPS には、名前空間 URI とそのプレフィックスのリストが保管されます。SPS がモジュールに追加されると、その中にある名前空間が参照元のSPS にあるスキーマノース内の名前空間と比較されます。追加された SPS 内にある名前空間 URI が、参照元 SPS のスキーマノース内にある名前空間 URI にマッチする場合、参照元 SPS のスキーマノースコ使用されているプレフィックスが、追加された SPS の名前空間プレフィックスとして使用されます。追加された SPS 内にある名前空間 URI が、参照元 SPS のスキーマノース内にある名前空間 URI のどれともマッチしない場合、エラーグッセージが表示されます。



上のスクレーンショナでは、あるSPS 内にある様々な名前空間がプレフィックスとともに、スキーマソレーサイドバーに表示されます。これら名前空間の値はメーススキーマから得られるため、編集することはできません。

グローバルテンプレート

追加されたSPS モジュールのグロー・ショナを参照)。デフォルトでは、追加されたモジュールに使用することができ、デザインソー・サイド・一に表示されます(以下のスクノー・ショナを参照)。デフォルトでは、追加されたモジュールに表示されているチェックボックスの状態を切り替えることにより、これらのモジュールを有効化/無効化することができます。グロー・ショテンプレートを作成して、追加されたモジュールから得られたグロー・シリテンプレートをオー・ドーライドする場合、グロー・ショテンプレートエ・トリーの隣にある・・・・アイエをクリックしてくたさい。新し、グロー・ショテンプレートを追加ダイアログが表示されるので、グロー・ショテンプレートを作成する要素ませは属性を選択してくたさい。スキーマ内にある目的のノードを選択する XPath 条件式を入力することできます。「OK」をクリックすることで、新たなグロー・ショテンプレートと追加されたモジュール内部のグロー・ショテンプレートのどうを有効化するのかを選択するように促されます。選択結果により新たに作成されたグロー・ショテンプレート、ませは追加されたモジュール内にあるグロー・ショテンプレートが有効化されます。2つあるグロー・ショテンプレートのチェックボックスを切り替えることにより、選択の切り替えを行うことができます。



追加されたモジュールのメインテンプレートは使用できない点に注意してくたさい。つまりモジュール化によりテンプレートを再利用する場合、グロー・バリテンプレートとしてテンプレートを作成する必要があります。特定の要素に対してグロー・バリテンプレートが定義されていない状況でその要素に対する処理が行われた場合、その要素に対するデフォルトの処理(XSLT の内蔵テンプレート)が使用されます。

デザインフラグメント

追加されたSPS 内にあるデザインフラグメント は、参照元のSPS にて利用することができ、SPS は、デザインソノーサイドバーに表示されます(上のスクノーシショナを参照)。デザイン内部にデザインフラグメントを挿入する際には、デザイン内の適切なコンテキストノード内へデザインフラグメントが配置されるように注意する必要があります。

追加されたモジュール

追加されたSPS モジュール内部に別のモジュールが追加されている場合、それらのモジュールも参照元のSPS から利用することができます。これにより、あるモジュールを追加することで、そのモジュールに追加された全てのモジュールを再帰的に利用することが可能しています。 勿論これらのモジュールには構成されるコンテンソモデルは、参照元 SPS モジュールのノーススキーマニ対して妥当なものでおければいません。モジュールの確認と管理は、デザン概要サイドバーで行うことができます。

スクノプト

追加された全てのSPS モジュールにあるスクノナトは参照元のSPS にて利用することができ、これらのスクノナトはデザインソト・サイドバー に表示されます。モジュールに追加されたスクノナトはライブラルに集められ、参照元のSPS からプロ・ティダイアログを介して利用することができるようにないます。

CSS スタイル

SPS モジュールに追加されたグロー・シレスタイルは、グロー・シレスタイルとして参照元のSPS へ渡され、スタイル・ポジャリサイド・デー内にレールならびにスタイルレールが表示されます。CSS ファイルはデザイン概要サイド・デーにも表示されます。同様に、SPS モジュールへ追加された。外部 CSS ファイルは、参照元のSPS にて利用することができます。

ページレイアウト

追加されたモジュールのページレイアナトは参照元のSPS から使用することができ、デザインソノーサイドバーにて表示されます。

参照元のSPS から利用することのできないモジュールオブジェクトオブジェクト

追加されたモジュール内にある以下のオブジェケトを参照元の SPS から利用することはできません。

- パラメーター定義:無視されます。
- スキーマソース: 追加された SPS がベースとているスキーマノースは無視されます。追加された SPS 内にあるドキュメト 要素のコンテンソモデルは、参照元 SPS のエンテンソモデル内に含める必要があり、そうでは、場合、追加された SPS をモジュールとして正しく使用できないと ら点に注意してください。必要な場合には、いつでもユーザー定義スキーマを参照元の SPS へ加えることができます。新たなスキーマコよ、追加されたグローバルテンプレートのコンテンソモデルを含めることができます。
- 作業用 XML ファイルならびにテンプレート XML ファイル: これらのファイルは対する参照が無視されます。参照元の SPS では、それ自身に与えられた作業用 XML ならびにテンプレート XML ファイルが使用されます。
- XPath デフォルト名前空間: これらの値がインポートされたモジュールこでセナされている場合、参照元のSPS にこれらの値が渡されることは対けません。

- 関連項目

- デザイン概要
- <u>モジュールSPS の作成</u>
- スキーマノース
- テンプレートとデザインフラグメント
- スクノプトの使用

6.3.2 SPS の作成

モジュールSPS の作成は、大きく分けて以下のステップにより行われます:

- 1. 追加するSPS モジュールのデザインと保存。
- 2. 使用される(つまり参照元の)SPS に対してモジュールを追加。
- 3. 必要に応じて<u>追加されたオブジェケトを有効/無効に</u>する。
- 4. 必要に応じて適切なオブジェクトを適用。

追加される SPS モジュール

他のモジュールへ追加される SPS を作成する際には、以下の2点に留意する必要がおます:

- 1. <u>参照元のSPS モジュール</u>にて使用することのできるテンプレートは、<u>グロー・ゾレテンプレート</u>だけです。つまり<u>追加される SPS モジュー</u>ルこで再利用した、アンプレートは、グロー・ゾレテンプレートとして作成する必要があります。
- 2. 追加される SPS モジュール内で定義されたドキュメント構造は、<u>参照元のSPS のノーススキーマ</u>. に 以定義される エンテンソモデル内部で 妥当で おければ おりません。 追加された テンプレートが SPS のメインスキーマニで 定義された エンテンソ モデル こ 含まれて ない 場合でも、 ユーザー 定義 スキーマ内で 定義することができます。

追加される SPS モジュールを作成する際には SPS がベースとしているスキーマは 以下のどれかどむます:

● 参照元 SPS のメインソーススキーマ。この場合、SPS が追加された際に追加されるグローバルテンプレートが、 参照元 SPS のメインスキーマにおけるコンテンツモデルの一部となります。そのため、Authentic View における これらグローバルテンプレートの出力は編集可能なものとなります。

- 参照元 SPS のメインスキーマにより定義されたコンテンツモデルの一部となるコンテンツモデルを定義したスキーマ。この場合、グローバルテンプレートが追加されると、参照元 SPS のメインスキーマのコンテンツモデルに対してテンプレートが追加されます。Authentic View におけるこれらグローバルテンプレートの出力は編集可能なものとなります。
- 参照元 SPS のメインスキーマにより定義されるコンテンツモデルの一部とはならないコンテンツモデルを定義したスキーマ。この SPS モジュールが追加されても、参照元 SPS のスキーマにおけるコンテンツモデルの一部となることはありません。それでもユーザー定義スキーマにより、グローバルテンプレートのコンテンツモデルを含むコンテンツモデルが定義される場合、出力の生成に使用することができます。Authentic View におけるこれらグローバルテンプレートの出力は編集可能なものとはなりません。

使用しているスキーマースニンテンソモデルを定義する場合、使用される名前空間 によりノードの展開名が決定されるため、慎重に操作を行う必要があます。

<u>作業用 XML ファイル</u>を使用することで、追加される SPS モジュールの出力をテストすることができます。この作業用 XML ファイルへの参照は、参照元の SPS にて無視されます。

SPS モジュールを追加する

SPS ヘモジュールを追加するコはデザク機要(こて新たなモジュールを追加コマンドをクリックします(以下のスクノーンショナを参照)。表示されるダイアログにて目的のSPS ファイルを選択し、開くをクリックします。



モジュールがSPS に追加され、デザイン概要のモジュール以下に表示されます。追加された全てのモジュールオブジェケトがデザインソノーサイドバーに表示されるとともに、追加された。CSS ファイルはデザイン概要サイドバーに表示されます。追加されたモジュールそのものが別のモジュールを参照している場合、それら参照されたモジュールは間接的にインポートされ、モジュール以下に灰色で表示されます。間接的にイン

ポーされたモジュールが、どのモジュールによりインポーされたものなのかは、そのく間接的にインポーされた)モジュールへマケスオーバーすることで確認することができます。

追加されたモジュールが間接的に个パートされたモジュールを StyleVision で開く口よ そのモジュールを右クリゾんて、コンテキストメニューからモジュールを開くを選択してください。

追加されたモジュールの順序

モジュールが追加される順序と表示される順序は、CSS スタイルの優先順位にとって重要な意味を持ちます。CSS のカスケード順序は、後の方(リストの下部)に定義されたモジュールが、それ以前(リストの上部)のモジュールに対して優先的に決定されます。参照元 SPS モジュール内にある CSS スタイルは、追加されたモジュール内にある全てのスタイルは、優先的に使用されます。追加されたモジュールの位置を変更するには、そのモジュールをデザイン概要にて右クアクレ、コンテキストメニューから上へまけば下へを選択します。

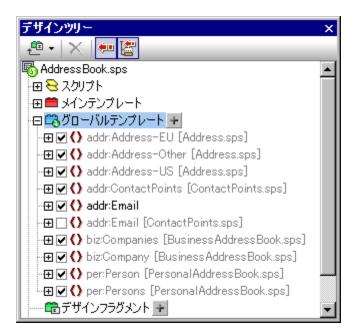
スクノプトやグロー・ジレテンプレート、デザインフラグメント、そしてページレイアウトアイテム間における衝突の解決に、モジュールの順序は意味を持ちません。

ファイルの変更警告

参照元のSPS モジュールが開かれている状態で、追加されたファイル(SPS モジュール、スキーマや作業用 XML ファイル)が変更されると、ファイルの変更ポップアップが表示され、参照元のSPS モジュールに変更を反映するか尋ねられます。

追加されたオブジェクトの有効化と無効化

追加された全てのモジュール内にある全てのモジュールオブジェクトは、(直接的、まけま間接的に追加されたかけっす)参照元のSPS へ追加され、デザインソー内の対応する箇所(スクレクト、グロー・ドレテンプレート、デザインフラグメント、XSLT テンプレート、XPath 関数)以下に表示されます。これらオブジェクトの隣にはチェックボックスが表示されており、チェックボックスを有効/無効にすることで、そのオブジェクトを有効/無効にすることができます。オブジェクトを無効にすることで、そのオブジェクトをSPS から実質的に削除することができます。



上のスクーンショナでは、AddressBook.sps モジュールコで使用されている全てのグロー・シリテンプレートがヴロー・シリテンプレート以下に表示されています。他のモジュールを介して追加されたモジュールは(それが直接的であれ間接的であれ)グレー色で表示されます。 AddressBook.sps 内で直接作成されたオジンエケトは黒色で表示されます。スクレーンショナでは、addr:Email と・ラグロー・シリテンプレートだけが、AddressBook.sps 内に作成され、その他のグロー・シリテンプレートは全て別のモジュールを介して追加され、オブジェケト名の隣に定義されたファイルの名前が表示されます。

addr:Email といろ名前のグロー・バーテンプレートが2つあることに注目してくたさい。1つは参照元のSPS(AddressBook.sps)にて作成されたもので、もう1つは追加されたモジュールのContactPoints.sps にて作成されたものです。複数のグロー・バーテンプレートが、同じ名前のく名前空間)展開名を持つ場合、そのうちの1つが1を同時に有効にすることができます。チェックドックスにより、有効にするものを選択することができます(その他にも、デザインビューのコンテキストメニューから、グロー・バーテンプレートを有効にすることもできます)。このメカニズムにより、(i)参照元 SPS モジュールこで作成したグロー・バーテンプレートのオー・バーライドや、(ii) ある要素に対して適用されるグロー・バーテンプレートが複数の追加モジュールにて定義されている状況での解決を行う際に便利な機能となります。

現在のSPS にて定義されたグロー・シリテンプレートは、選択した後に削除がなどをケックする事で削除することができますが、追加されたモジュールこで定義されたグロー・シリテンプレートを参照元のSPS にて削除することはできません。削除を行うには、追加された SPS を開き、そこでグロー・シリテンプレートを削除する必要がおります。

個々のスクノフトならびにデザインフラグメントも、同様の方法で有効化/無効化することができます。

モジュールオブジェクトの適用と使用

参照元のSPS モジュールにて、通常の方法によりテンプレートのデザインを行います。追加されたオブジェケトは、その種類により異なる方法で適用されます。 勿論、 適用する各モジューリオブジェケトを 有効にする必要があります。

グローノジレテンプレート

追加されたSPS モジュールからグロー・ドルテンプレートを使用するコよ、そのグロー・ドルテンプレートが本当に適用されたかどうか確認する必要があます。作成中のデザインにより、以下にある方法のどれかで確認を行うことができます:

- メインテンプレートにて、要素テンプレートがグローバルテンプレートを使用するように指定するか、グローバルテンプレートをローカルにコピーします。デザインにて要素タグを右クリックすることで表示されるコンテキストメニューにて、これらのコマンドを利用することができます。
- メインテンプレートでは、コンテンツやその他のコンテンツプレースホルダによりテンプレートが適用され、関連するグローバルテンプレートが処理されます。

デザインフラグメント

デザインフラグメントを使用するコは、デザインソノーからメインテンプレートやグロー・ジレデンプレートテンプレート内にある目的の場所へ、デザインフラグメントをドラッグします。デザインフラグメントが、適切なコンテキストノードへドロップされるようにしてください。詳細にないには、次を参照してください、デザインフラグメント。

スクリプト

デザインフラグメトを使用するコよ デザインソーからメインテンプレートやグロー ジレテンプレートテンプレート内にある目的の場所へ デザインフラグメトをドラッグします。 デザインフラグメトが、 適切なコレテキストノード・ヘドロップされるようししてどとい。 詳細 こしては プロ・ティサーバ・バーから 特定のイベント こうしてセント することができます.

- 関連項目

- デザイン概要
- 利用が可能なモジュールオブジェクト
- サンプル: Address Book
- スキーマノース
- テンプレートとデザインフラグメント
- スクノプトの使用

6.3.3 サンプル: Address Book

(マイ)ドキュメトフォルダー以下にあるC:\Documents and Settings\<username>\My

Documents Altova Style Vision 2021 Style Vision Examples Tutorial Modular SPS **におモジュールSPS のサンプルが収められています。このフォルダー内にあるサンプルファイルにより、ビジネス用と個人用の連絡先がモジュール化されているアドレス帳のプロジェケトファイルが構成されます。サンプルにより、モジュール化がどのように実装されるが確認することができるとともに、モジュール化を行う利点を知ることもできます。**

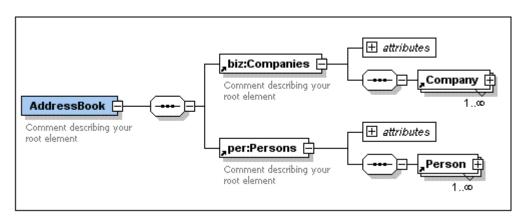
- アドレス帳は、(i) ビジネス用の連絡先と、(ii) 個人用の連絡先という2つのモジュールから構成され、両方とも独立した SPS によりデザインされます。2つのモジュールを合わせることで、合成されたアドレス帳が作られます。ここでは複数のモジュールにより、より大きなデータが構成されます。
- 各モジュール(ビジネス用と個人用の連絡先)のコンテンツモデルはそれぞれ多少違っていますが、住所や電話番号、FAX、電子メールといった情報から構成される ContactPoints モジュールという共通のモジュールを含んでいます。従って、ContactPoints モジュールは2つの連絡先(ビジネス用と個人用)にて共有することができます。モジュール化により、1つのモジュールをその他複数のユニットにおける共通部分として使用することができます。
- 更に、ContactPoints モジュールにより、柔軟性を向上させることができます。サンプルプロジェクトでは、EU(欧州)の住所、US(米国)の住所、またはその他地域の住所に対応した、3つのコンテンツモデルを含む独立した Address モジュールが作成されています。これら3つのコンテンツモデルに対する出力は単一の SPS にて決定 されていますが、それぞれの SPS にて細かな定義を行うことも可能です。この場合、モジュールの再利用を簡単に行うことができるため、モジュールを柔軟に使用することが可能になります。

このプロジェクトに関する説明は、以下のようにまとかられています:

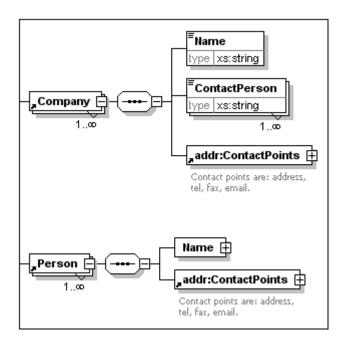
- スキーマファイル
- XML データノース
- SPS ファイル

スキーマファイル

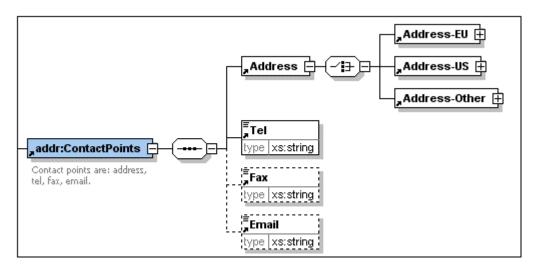
モジュールSPS に対するスキーマを作成する場合、再利用する要素をグロー い要素として作成しなければいられいに留意してください。アドレス帳のスキーマは AddressBook.xsd で、ビジネス用のアドレス帳 (BusinessAddressBook.xsd) と個人用のアドレス帳 (PersonalAddressBook.xsd) をインポーナすることで構成されています。BusinessAddressBook.xsd スキーマにより会社用のコンテンソモデルが与えられ、PersonalAddressBook.xsd スキーマにより個人用のコンテンソモデルが与えられます(以下のスクレーシンョナを参照)。



両方のスキーマーで連絡先詳細のエレテンツモデルが定義されている ContactPoints.xsd スキーマがインポートされます。



ContactPoints.xsd スキーマコはEU、US、そしてその他(Other)の地域に対応したアドレス帳のコンテンソモデルが定義された Address.xsd スキーマか含まれています(以下のスグリーンショントを参照)。



対象となるスキーマが別の名前空間に属している場合、スキーマが「インポート」されます。対象となるスキーマが同じ名前空間に属している場合、スキーマオゴインクルード」されます。

メモ 上に示されるスキーマのスクリーンショットは、Altova XMLSpy のスキーマビューです。

XML データノース

XML データはAddressBook.xml ファイル内に含まれています。このファイルでは、AddressBook 要素内にcompanies とpersons 要素が子要素として含まれています。これら2つの要素のコンテングモデルは、それぞれBusinessAddressBook.xsd とPersonalAddressBook.xsd スキーマファイルこで定義されています。

BusinessAddressBook.xsd とPersonalAddressBook.xsd スキーマンぞれぞれ対応する XML データファイルも存在します。これら2つの XML ファイル(BusinessAddressBook.xml ならびこPersonalAddressBook.xml) は対応する SPS ファイルの作業用 XML ファイルとて使用されます。

SPS モジュールと作業用 XML ファイルの対応を以下に示します:

- AddressBook.xml => AddressBook.sps, ContactPoints.sps, Address.sps
- BusinessAddressBook.xml => BusinessAddressBook.sps
- PersonalAddressBook.xml => PersonalAddressBook.sps

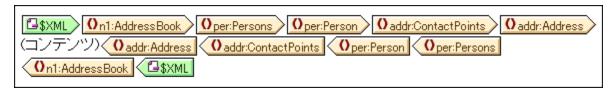
SPS モジュール

最も基本的なモジュール (Address.sps) を最初に説明し、アドレス帳 (AddressBook.sps) を完成させるまでのステップを記述します。全てのSPS モジュールでは AddressBook.ssd をスキーマとして使用しています。

Address.sps

スキーマと作業用 XML ファイルの使用に関するポイントを以下に記します。

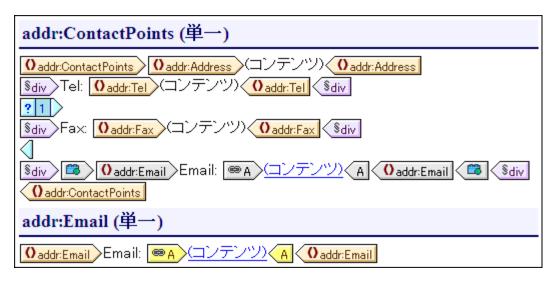
- Address.sps では AddressBook.xsd がスキーマとして使用されます。しかし、Address.xsd、ContactPoints.xsd、BusinessAddressBook.xsd、PersonalAddressBook.xsd の全てに Address 要素がグローバル要素として含まれているため、これらスキーマのどれかを使用することもできます。SPS モジュールが別の SPS モジュール内に追加されると、インポートされたモジュールのスキーマが無視されるため、SPS がモジュールとして追加される際にどのスキーマが使用されたかは重要ではなくなります。
- AddressBook.xml が作業用 XML ファイルになります。Address.sps 内にあるメインテンプレートでは、Address 要素だけが処理されるように指定されているということと、Address-EU、Address-US、そして Address-Other のグローバルテンプレートが定義されているだけである点に注目してください。



Address 要素だけが処理されるため、出力プレビューにて Address の出力だけが表示されます。Address.sps がモジュールとして使用される場合、グローバルテンプレートが追加され、メインテンプレートが無視されます。

ContactPoints.sps

- ContactPoints.sps では AddressBook.xsd がスキーマとして使用され、AddressBook.xml が作業用 XML ファイルとして 使用されます。
- Address.sps がモジュールとして追加されるため、Address-EU、Address-US、Address-Other 要素がグローバルテンプレートとして利用可能になります。
- ContactPoints と Email 要素のグローバルテンプレートが定義されます。ContactPoints の定義に Email のグローバルテンプレートが使用されている点に注目してください(以下のスクリーンショットを参照)。



• プレビューに必要なメインテンプレートでは ContactPoints 要素のグローバルテンプレートが使用されているため、ContactPoints 出力のプレビューが有効になっています。

BusinessAddressBook.sps & PersonalAddressBook.sps

プレビューに必要なメインテンプレートでは ContactPoints 要素のグローバルテンプレートが使用されているため、 ContactPoints 出力のプレビューが有効になっています。

- これら2つのモジュールでは AddressBook.xsd がスキーマとして使用されます。作業用 XML ファイルには、BusinessAddressBook.xml と PersonalAddressBook.xml がそれぞれ使用されます。
- ContactPoints.sps がモジュールとして使用されます。これにより Address.sps が間接的にインポートされます。これ ら2つのモジュールにある全てのグローバルテンプレートが参照元の SPS モジュールにて利用可能になります。
- BusinessAddressBook.sps グローバルテンプレートが Companies ならびに Company 要素に対して定義されます。 Company の定義ではグローバルテンプレートの ContactPoints が使用されている点に注目してください。
- PersonalAddressBook.sps では、グローバルテンプレートが Person ならびに Persons 要素に対して定義されます。 Person の定義ではグローバルテンプレートの ContactPoints が使用されている点に注目してください。

AddressBook.sps

Email 要素に対して2つのグロー・ジレテンプレートが表示されており、そのどちらかを有効にすることができます。

- AddressBook.sps では AddressBook.xsd がスキーマとして使用されます。作業用 XML ファイルには AddressBook.xml が使用されます。
- BusinessAddressBook.sps と PersonalAddressBook.sps がモジュールとして追加され、その結果 ContactPoints.sps と Address.sps が間接的にインポートされます。
- グローバルテンプレートが Email 要素に対して定義されます。つまり Email に対して、ContactPoints.sps と AddressBook.sps から得られた2つのグローバルテンプレートが存在します(以下のスクリーンショットを参照)。



デザインツリーのグローバルテンプレートでは、どのグローバルテンプレートをアクティブにするか選択することができます。StyleVisionで同時に使用することのできる同名のグローバルテンプレートは1つだけで、どちらがアクティブになろうとも、ContactPoints グローバルテンプレートにて使用されます。

• メインテンプレートには、出力のヘッダーに使用される静的コンテンツが含まれています。

□ 関連項目

- デザイン概要
- 利用が可能なモジュールオブジェクト
- モジュールSPS の作成
- 7+-7/-7
- <u>テンプレートとデザインフラグメント</u>
- スクノプトの使用

6.4 テンプレートと デザインフラグメント

デザイドキュメトはテンプレートによく構成されるため、使用することができるテンプレートの種類を理解することが重要になります。

- メインテンプレートとグローバルテンプレート: デザインドキュメントは1つの メインのデンプレート と オプションとなっている1つ以上 グロー バリテンプレート によい構成されます。グローバリテンプレートはメインテンプレートから参照することができます。
- ノードテンプレート:メインテンプレートとグローバルテンプレートを構成するテンプレートのことで、ノードテンプレーはスキーマソース内にある ノードテンプレート にマッチします。
- デザインフラグメント:個別に作成され、デザイン内にあるメインテンプレートまたはグローバルテンプレートので再利用ことができるテンプレートです。

このセグションでは、デザインの構造内部にてテンプレートやデザインフラグメトが果たす役割について説明します。ここで重要になるのはデザイン内の表示プログライではなく、その構造です。

メモ デザインビューでは、メインテンプレート、グローバルテンプレート、とデザインプラグメトと、複数のテンプレートを、SPS に含めることができます。 <u>ツール ドアイコン</u>として用意されているデンプレート デザインフィルター機能を使用することで、デザインビューに表示されているデンプレートを管理することができます。

□ 関連項目

- デザインの作成
- <u>スキーマノース</u>
- デンプレートフィルター

6.4.1 メインのテンプレート

メインのテンプレートにより、出力の構造が決定されます。つまり、デザインあるメインテンプレート内の順序が、そのまま、出力における表示順序となります。プログラング用語を使えば手続き型の処理となり、テンプレートの開始点から処理が開始され、最後まで順序とおり処理が行われます。その中で XML ドキュメントから得られたノードが処理されます。これらノードが処理されるテンプレートはコーカルテンプレートと呼ばれます。あるローカルテンプレートにおける処理が終わると、メインテンプレート内にある次のエンポーネントへ処理が移ります。ノードの処理を行うために、ノードから グロー・ドルテンプレート への参照が行われることもあります。この場合、そのノードに対してグロー・ドルテンプレートが実行された後に呼び出し元のメインテンプレートへ処理が戻り、次のエンポーネントへ処理が移ります。

メインテンプレートのエトリーポイトはスキーマの<u>ドキュメトノード</u> どいます。StyleVision ではオプションとして複数のレート要素 (<u>ドキュメート要素</u>) を選択することもでき、メインテンプレート内部で、アクティブぶ各ドキュメート要素に対して ローカルデンプレートをセナすることができます。処理中に実行されるテンプレートは、処理されている XML インスタンスドキュメートのドキュメート要素に対して定義されたテンプレートといます。

□ 関連項目

- デザインの作成
- テキストとして XML コンテンツを挿入
- グローバルデンプレート
- <u>デザインフラグメント</u>
- デンプレートフィルター

6.4.2 グローバルテンプレート

グロー ジョテンプレート は、任意のソードやスキーマ内の型、おけはXPath 条件式により指定されナノードに対して定義することができます。

グロー、ジリテンプレートは選択されたノードや型の処理命令に対して適用され、メインのテンプレート、デザインフラグメント、まけはその他のグ ロー・ゾリテンプレートから呼び出されます。 あるテンプレートが定義され、 複数回呼び出されるというのは宣言型の動作ですが、 手続き型の処 理でも使用されます。このようはして、ある定義を複数回再利用することが可能しています。グロー・ゾリテンプレートの呼び出しは、以下のよう な状況で発生します:

- メインのテンプレート 内にあるノードや型が、(デザイン内のエンポーネントを右クシクし、グロー シリテンプレートを選択して)グ ローバルテンプレートを参照するようにセットされている。
- (コンテンツ) おけよ(その他のコンテンツ) がローカルテンプレート 内にある要素や型内に挿入され、その要素に対するその他のコン テンパ型にはグロージテンプレーのノードヤ型が含まれている。

様々な要素や複数の場所でノード(おけは型)が出現する、おけば特定の処理命令が全ての出現に対して必要となるような状況で、グロー バルテンプレートは更利な機能となります。例えば、chapter、section、appendix、まけまblockquote 要素内にあるかどうかにかかわらず、 para 要素の書式が常に同じものになるようにすることを考えてみましょう。para 要素に対してグローバルデンプレートを定義して、メインのデ プレート 「こて para 要素に対するグロー・ シレテンプレートが適用されるように指定することができます。 例えば //chapter/para をメインテンプ レート内に追加して、para がブロー ジリテンプレートを参照するようし、//chapter/title を追加した後に(コンテンツ) または(その他のコン テンツ を追加することで、chapter 要素内にあるその他のエレテンツが、利用可能なグロー・バリテンプレートやデフォルトテンプレートにより処 理されるようごなります。更に、グロー・バリテンプレートは複合型(例: アドレスモデルの定義)や単純型(例: xs:decimal)に対しても定義す ることができます。この場合、その型(複合型まけは単純型)により定義されて出現のうち、グロー・ジリテンプレートを呼び出す全ての出現にお いて、グロー・ゾレテンプレートのルールに従って処理が行われます。

グローバルテンプレートの作成

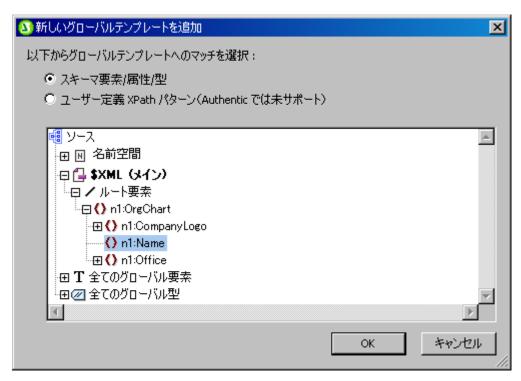
グローバルテンプレートは、スキーマ内にある任意のノードや型、または XPath 条件式にて指定されたノードに対 して作成することができ、スキーマツリーサイドバーにて作成されます(以下のスクリーンショットを参照)。



グローバルテンプレートは以下にあるいずれかの方法により作成することができます:

スキーマツリーに表示されているグローバルテンプレートアイテムの右側にある新しいグローバルテンプ レートを追加アイコンをクリックします(上のスクリーンショットを参照)。新しいグローバルテンプレートを追 加ダイアログが表示されます(以下のスクリーンショットを参照)。ダイアログに表示されているスキーマツ

リーから要素や属性、または型を選択することができるほか、XPath 条件式を入力することもできます。この操作により、グローバルテンプレートとして作成されなければならないノードが決定されます。「OK」をクリックするとテンプレートが作成され、デザインビューにすでに存在するテンプレートに追加され、編集することができるようになります。スキーマツリーでは、スキーマノードや型アイコンにプラスサインが表示されます。



- スキーマツリーのルート要素以下にあるスキーマノードや型コンポーネントを右クリックして、グローバルテンプレートの作成/削除コマンドを選択します。テンプレートが作成され、デザインビューに既に表示されている既存のテンプレート以下に追加されます。スキーマツリーでは、ノードや型のアイコンにプラスサインが表示されます。
- デザインビューのメインテンプレートにあるテンプレートからグローバルテンプレートを作成することもできます。デザインビューやスキーマツリーサイドバーにあるテンプレートを右クリックして、グローバルテンプレートに設定を選択します。選択されたテンプレートからグローバルテンプレートが作成され、メインテンプレート内にあるテンプレートが、このグローバルテンプレートを使用するように自動的に定義されます(グローバルテンプレートの使用方法については、下記を参照ください)。

グロー シレテンプレートは、メインのデンプレートの下のデザインビュー内に存在します。グロー シレテンプレートのオー幼に作成されたノード名を含むゞ~を操作して表示されます。: (単一) まけは(複合型)。 グロー シレテンプレートは、下のスクリーンショナに表示されています。



グロー・シレテンプレートの処理は、ユーザー定義で、静的と動的であることができます。 おた、メインのテンプレートの処理のために使用することができます。

グローバルテンプレートの使用

作成したグローバルテンプレートは、同じ修飾名を持ったノードがドキュメントへ挿入された際に使用することができます。ノードがデザイン上にドロップされると表示されるメニューにて、グローバルテンプレートを使用を選択してください。その他にも、ローカルテンプレートがデザイン内にあり、同じ修飾名を持ったノードに対してグローバルテ

ンプレートが存在する場合、ローカルテンプレートの代わりにグローバルテンプレートを使用することができます。ロ 一カルテンプレートに対してグローバルテンプレートを使用するには、デザインビューにあるローカルテンプレートを 右クリックして、グローバルテンプレートを使用コマンドを選択します。グローバルテンプレートが使用される場合、 内部に含まれる処理命令が呼ばれ、ランタイムにてローカルテンプレートにより使用されます。

グローバルテンプレートがデザインにて使用される場合、グローバルテンプレートにて XPath のパターンを作成し、 ノードセットのフィルタリングを行うことができます。フィルターを作成するには、デザインビューにてグローバルテン プレートのタグを右クリックして、コンテキストメニューから XPath フィルターの編集 を選択します。 XPath 式の編集ダイ アログが表示され、目的の条件式を入力することができるようになります。

再帰的なグローバルテンプレート

グロー・ゾリテンプレートは再帰的に作成することも可能で、グロー・ゾリテンプレートが自分自身を呼び出すように定義することができます。 Authentic View における無限ループを回避するために、呼び出しの深さを制限するためのプログライコ値をセナすることができます。SPS のプロ ティダイアログ(「ファイル」プログティ」) にある Authentic ダブ内のテンプレート呼出し階層上限プロティニで値をセナす ることができ、この値により、Authentic View 出力を処理する際に再帰的なテンプレート呼び出しの最大値が指定されます。テンプレート 呼出し階層上限プロ・ティで指定された値を超えたテンプレートの呼び出しか行われた場合、エラーが返されます。

グロー/ シレテンプレーをローカル(コピーする

グローバルテンプレートが作成されると、その処理命令をメインテンプレート内にある同名のテンプレートへ直接コピーすることができます。ローカルテンプレートを右クリックして、グローバルテンプレートをローカルにコピーを選択します。グローバルテンプレートをローカルにコピーするという処理は、処理命令が一度だけコピーされるという 点で(ランタイムにおける)グローバルテンプレートの使用とは異なります。一旦ローカルにコピーされたら、グロー バルテンプレートがローカルテンプレートに対して影響を及ぼすことはありません。グローバルテンプレートとローカ ルテンプレートのどちらか(または両方)をその後修正しても、相手に対して影響を与えることはありません。グローバルテンプレートがローカルテンプレートにより使用される場合、グローバルテンプレート側の変更が、ランタイムに てローカルテンプレートに反映されます。

グローバルテンプレートの有効化と無効化

スキーマツリーサイドバーに表示されているグローバルテンプレートのエントリーをチェックすることで、そのグロー バルテンプレートが有効化されます。エントリーのチェックを外すことでテンプレートが無効化されます。グローバル テンプレートが有効化された場合、テンプレートが XSLT スタイルシートにて生成されます(グローバルテンプレート が作成されるとデフォルトで有効になります)。無効になっている場合、XSLT スタイルシートにて生成されることは ありませんが、SPS デザインには保存されます。

無効になっているグローバルテンプレートを使用するローカルテンプレートでは、(存在しないグローバルテンプレー トを参照することはできないため)XSLT のデフォルトテンプレートが使用され、コンテンツやテキストノードの子孫に もその影響は現れます。

有効化/無効化の機能を使用することで、(i) グローバルテンプレートが一時的に必要ない場合でも削除する必要 は無く、必要になった時に再度有効化することができ、(ii) インポートされたスタイルシートにあるテンプレートにて名 前の衝突が起こった際に、必要ないグローバルテンプレートを一時的に無効化することが可能になります。

グローバルテンプレートの削除

グローバルテンプレートを削除するには、削除するグローバルテンプレートをデザインビューまたはスキーマツリー サイドバーにて右クリックし、グローバルテンプレートの作成/削除コマンドを選択します。

単純型のグローバルテンプレートと複合型のグローバルテンプレート

グロー バテンプレート は、単純型と複合型の2種類がかます。複合型グロー バテンプレートは下位互換性を保証するためのもので、バ ージョン 2006 より前のStyleVision にて作成された SPS のグロー・ジリテンプレートにテーブルやリストか含まれている場合、 StyleVision 2006 やそれ以降の デジュンでは 通常複合型として、そのブロー シテンプレートが開かれます。

複合型のグロー ジリテンプレートと単純型のグロー ジリテンプレートとの違いよ、グロー・ジリテンプレートに対して作成されたノードの処理方法におります。ノードの最初のインスタンスがドキュメントにて見つかった場合、そのノードの直後にある全てのインスタンスが複合型のグロー・ジリテンプレートにより処理されます。単純型のグロー・ジリテンプレートの場合、個々のノードインスタンスが発見されるごとに処理が行われます。

グロー ジレテンプレートに対して作成された要素ノードの周りに、定義済みのフォーマル おけお改行が作成されている場合、単純型のグロー ジレテンプレートは、自動的に複合型のグロー ジレテンプレートへ変換されると ら点に留意してください。これにより、複合型のグロー ジレテンプレートと同様の処理が行われます。単純型のグロー ジレテンプレートへ戻したい場合は、(定義済みのフォーマルの外側へノードをドラッグし、定義済みフォーマルを削除することで)定義済みのフォーマルを削除するか、デザインソレーサイドバー内にあるアイテムを削除することで)改行を削除する必要がおります。単純型のグロー ジレテンプレートが複合型のグロー ジレテンプレートへの自動変換を回避するには、単純型のグロー ジレテンプレートに対して作成された要素のノードタグ内部に定義済みのフォーマル や改行を追加するようにしてください。

モジュールSPS におけるグローバルテンプレート

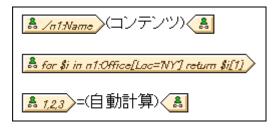
SPS モジュール別のSPS モジュール・追加された場合、追加されたモジュール内にあるグロー・ジリテンプレートを、参照元のSPS にて利用することができます。モジュールSPS の使用に関する詳細については、モジュールSPS のセグションを参照ください。

■ 関連項目

- テキストとして XML コンテンツを挿入
- その他のコンテンツ
- メインのテンプレート
- デザインフラグメント
- デンプレートフィルター
- モジュールSPS

6.4.3 ユーザー定義テンプレート

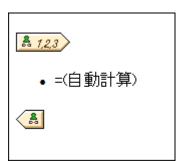
ユーザー定義テンプレートとは、XPath 条件式により生成されるアイテムで対して適用されるテンプレートのことです。これらアイテムは原子型の値ませるレードといります。以下のスケリーシショナでは3つのユーザー定義テンプレートが示されており、タグの左側にユーザー定義テンプレートアイコンが示されます。ユーザー定義テンプレートが持つ柔軟性により、テンプレートを作成することができます。



上に示されるこれら3つのユーザー定義テンプレートに対して定義されたXPath 条件式により、以下の処理が行われます:

- ソーススキーマ内のノードを選択します。XPath 条件式を使用することで、全てのエンテキストノードから、任意のスキーマノース内にある任意のノードへアクセスすることができるようけます。曖昧性が無い対すった。StyleVisionが定義されたノードを識別することができる場合、テンプレートはユーザー定義テンプレートから通常のテンプレートへ自動的に変換されます。ユーザー定義テンプレートの場合、テンプレートタグの左側にユーザー定義テンプレートを表す緑色のアイゴンが表示されます。
- XPath 2.0 とXPath 3.0 のfor コンストラケトにより指定された条件を満たすノードを選択します。このようなテンプレートは StyleVision に通常のテンプレート(ユーザー定義テンプレート) を明確に解決することを許可しません。
- 原子型の値のシーケンス(1、2、3)を選択します。原子型の値に対してテンプレートを作成することは可能ですが、このようなテンプレート

内でコンテンソプレース市」がを使用することはできません。これは、(コンテンソプレース市」がことと生成される)xsl:apply-template 処理命 令が、(原子型の値ではなく)ノードアイテムさしか適用できない ヤーめです。 そのかわり、 箇条書きといオーデザイン要素とともに、 自動計算を 使用することは可能で、以下のスクレーシンョナの左側にあるユーザー定義関数により、右側のような出力が生成されます。



- 1
- 火モ SPS にてXSLT 1.0 が使用されている場合、XPath 条件式の返り値をノードセナにする必要があり、そうでは、場合エラーと なます。

XPath を使ったテンプレートノードの選択を行う利点

スキーマノードをXPath 条件式(ユーザー定義テンプレート)にお、選択することで、XPath のパスセノクターメカニズムを使*オニアイテムノー*ド やシーケンスの選択を行うことができるようコなり、ノードの選択に対してフィルダレグや条件をセナすることが可能についます。そのため、特定 のXML ドキュメントノードを、任意のテンプレートに対して使用することができるようにさいます。例え

ば、//Office/Department[@Location="NY"] とうXPath 条件式に以、NY とう値を持ったLocation 属性が含まれるDepartment ノードだけが選択されます。このセケション内にあるその他の例も参照くたさい。

旡 XPath 条件式に複数のロケーション・スペロケーションステップが含まれる場合、複数のロケーション・スペステップ内に括弧が使 用されているかかヴループ化や並び替えを行う際に重要になります。例えば、/Org/Office/Dept とら XPath 条件式は、 (/Org/Office/Dept) とう条件式とは異なって方法で処理されます。括弧を伴わない条件式では、各ロケーションステップが繰り 返されるのごさして、括弧付きの条件式の場合、全てのOffice 要素にある全 Dept 要素が 1つのノードセナにより返されま す。

括弧	実際のXSLT メカニズム	効果
無し	<pre><xsl:for-each select="Org"> <xsl:for-each select="Office"></xsl:for-each></xsl:for-each></pre>	各 Office 要素には独自の Dept か含まれ、グループ化や 並び替えは、各 Office 内部で 行われる。
有り	<pre><xsl:for-each select="/Org/Office/Dept"> </xsl:for-each></pre>	Dept 要素が、全てのOffice ならびにOrg 要素に対して展 開される。

グループ化や並べ替えを行う際にXPath 条件式の評価方法が大きな違いをもたらします。

ユーザー定義テンプレートを挿入する

以下の操作によりユーザー定義テンプレートを挿入することができます:

- 1. デザイン要素の挿入ツール・インあるユーザー定義テンプレートを挿入アイコンをクックし、デザイン内のテンプレートを挿入する場所を クックします。 まけは、テンプレートを挿入する場所を右クックし、コンテキストメニューからユーザー定義テンプレートを挿入コマンドを選 択します。
- 2. In the XPath 式を編集 ダイアログにて XPath 条件式を入力し、OK をクトックします。 XPath 条件式のエンテキストノードは、クトックされたノードしてお点に注意してください。 空のノードテンプレートが作成され、場合によっては結合されたノードが作成されます。 ノードが結合されると、ターゲートとなるインスタンスノードが同一のレベルにて選択されるのに対し、ノードが結合されていない場合(つまり複数の階層レベルにて分割されている場合)、ループにより各インスタンスノードが選択されます。 グループ化や並べ替えが行われない 限り、 (結合まがは分割により)返された両者のノードセットに違いながません。 結合されたノードがグループ化や並べ替えメカニズムでどのように扱われるかの詳細については、・ノードランプレートオペレーションを参照してください。

テンプレートマッチの編集

任意の、ユーザー定義まけば通常のノードテンプレートに対する選択は、新たな XPath 条件式により変更することができます。ノードテンプレートのテンプレートマッチを編集するける、ノードテンプレートを右クトックし、テンプレートマッチの編集コマンドを選択します。 XPath 条件式を編集するためのダイアログが表示されるので、新たなノードを選択する XPath 条件式を入力し、「OK」をクトックします。

ユーザー定義テンプレートにノードを追加する

SPS のスキーマノースコンテキスト内に配置することができないソードまけばシーケンスに対してユーザー定義テンプレートが作成された場合、スキーマソナーから得られたノードがユーザー定義テンプレートへ追加される時に、新たなノードのコンテキストが決定できなくなります。そのため、新たなノードが(i)名前で(つま)相対パスで)参照されるのか、まけは(ii)スキーマノースのルートからのフルパにより参照されるのか決定するよう促されます(以下のスグリーシンョナを参照)。



どのような動作を行うか、確認するのがデフォル・の振る舞いでなっています。 「<u>ツール」オプション ダイアログ」</u>のデザインダブこて、デフォルトの振る舞いを変更することができます。

6.4.4 変数テンプレート

変数テンプレート とは変数に対して使用されるテンプレートで、通常その値が出力されます。メニューオプションの 挿入 | 変数テンプレートの挿入」ませば タグで囲む | 変数テンプレート」を選択することで、カーノル位置に、SPS で定義される変数のテンプレートが挿入されます。 変数テンプレート には、デフォルトでコンテンソプレースポルダか含まれており、これにより変数の値が出力されます(以下のスグレーショナを参照)。 変数テンプレート には、必要に応じてその他の(静的ならびに動的な) コンテングを含めることができるまか、その他のテンプレートと同様に修正を行うこともできます。



以下の操作により変数テンプレートを挿入することができます:

- デザイン内のテンプレートを挿入する位置にカーノルを配置します。
- 2. メニューオプションから「挿入 | 変数テンプレートの挿入」を選択します。変数テンプレートの挿入ダイアログが表示されます(以下のスクリー・シショントを参照)。



- 3. ダイアログロは SPS にて定義された ユ<u>ザ・宣言パラメーターと変数</u> が表示されます。 変数テンプレートに対して追加する変数を選択してください。
- 4. **OK** をクトックして完了します。

6.4.5 ノードテンプレート操作

ノードテンプレートとは、ノードに対する処理を指定するデザイン内のテンプレートのことです。デザインでは、ベージュ色の開始ならびに終了タグによりノードテンプレートが表示されます(以下のスクリージンョナを参照)。以下のスクリージョナでは、metainfo とrelevance とう2つのノードテンプレートが示されています。次も参照してくたさい、XMLドキュメント内のノード。



ノードテンプレートにて行うことができる処理の一覧は、そのソードテンプレートの開始ませは終了タグを右クリックすることで表示されるコンテキストメニューから確認することができます。

以下ではエンテキストメニュー内にあるコマンドについて説明します:

- グローバルデンプレート
- デンプレートの一致
- XPath フィルター
- グループ化、並べ替え、変数の定義、レベルとして機能するテンプレート
- デザインフラグメントの作成
- タグ沿州除
- 編集、タグで囲む、変更

以下ではこれらのメニューコマイ・にこいて説明します。ノードテンプレートによっては利用できないロマイ・もある点に注意してください。利用できないロマイ・はエンテキストメニューにてグレーアナーされます。

グローバルテンプレート: 設定、使用、ローカルはコピー

メインテンプレート内にあるノードテンプレートを変更することができるまか、以下のロマンドによりグロー・ジレテンプレートへ割り当てることができます:

- メイングローバルテンプレートに設定:スキーマにてグローバル要素として定義されている要素をノードが指し 示している場合に使用することができます。ノードテンプレートからグローバルテンプレートが作成されます。メインテンプレート内にあるノードテンプレートではこのグローバルテンプレートが使用され、そのタグが(グローバルテンプレートの使用を表す)グレーで表示されます。
- グローバルテンプレートを使用: ノードテンプレートの名前と同じ修飾名でグローバルテンプレートが定義されている場合、ノードテンプレートに対してグローバルテンプレートの処理が使用されます。ノードテンプレートのタグがグレーになります。
- グローバルテンプレートをローカルにコピー: ノードテンプレートと同じ修飾名を持ったグローバルテンプレートの処理命令が、直接ノードテンプレートへコピーされます。ノードテンプレートはグローバルテンプレートとは独立したものになります。このコマンドを実行した後に、グローバルテンプレートとは独立したものとしてノードテンプレートを変更することができます。ノードテンプレートからグローバルテンプレートへの参照は行われないため、ノードのタグはベージュ色のままとまります。

に関する詳し、説明は、次のセグションを参照してください、グロー・ジェテンプレート.

テンプレートマッチの編集

テンプレートに対して作成されたノードを、このコマンドにより変更することができます。テンプレートマッチの編集コマンドにより、XPath 式 の編集ダイアログ が表示され、スキーマ内にある他のノードを選択する XPath 条件式を入力することができます。 XPath 条件式を入力することで、テンプレートをユーザー定義テンプレートへ変更することもできます。

XPath フィルターの編集/クリア

XPath フィルターにより、ノードテンプレートが適用されるノードセナのフィルタルグを行うことができます。XPath フィルターはグローバルテンプレートに対して適用することができます。

デフォルでは、ノードテンプレートが作成されたノードに対応するノード(要素まけは属性)に対してノードテンプレートが適用されます(スキーマ構造の指定された位置にある同名の出現)。例えば、/Personnel/Office ノードにより、全ての/Personnel/Office 要素が選択されます。Office 要素にて1 という式を持った XPath フィルターが作成された場合、述語式がOffice 要素に加えられるのと同様の結果が得られ、全体のXPath 条件式は、/Personnel/Office[1] というます。この XPath 条件式により、最初の Personnel 要素にある最初のOffice 子要素が選択され、その他の Office 要素は選択されないことしています。

フィルターは、デザイン内にある任意のノー・デンプレートならびに複数のノー・デンプレートに対して追加することができます。この機能を使って、例えば/Personnel/Office[@country='US']/Person[Title='Manager'] といたXPath 条件式により、社内のUS オフィス こる管理職全員を選択することができます。この例では、Office とPerson のノー・デンプレートにて、それぞれ独立したフィルターが作成されます。

グロー・シレテンプレートが使用される(呼び出される)場合、XPath フィルターを適用することができます。従って、使用されているグロー・シレテンプレートのインスタンス全てに対して XPath 条件式を適用することで、ターゲットとなるノードセットを制限することができます。

XPath フィルターをノードに追加するコよノードテンプレートを右クトグして、XPath フィルターの編集を選択します。引用句や角かって、デトターなしてXPath フィルター条件式を入力してください。以下にあるような妥当な XPath 条件式を入力することができます:

- 1
- @country='US'
- Title='Manager'

ノードテンプレートに対してXPath フィルターが追加されると、そのソードテンプレートの開始タグにフィルターシンボルが表示されます。以下のスグルーシンコナでは、synopsis ノードテンプレートに対してフィルターが追加されました。

Osynopsis[//]〉(コンテンツ) Osynopsis

メモ 各ノードテンプレートでは、1つの XPath フィルターがサポートされます。

グループ化、並べ替え、変数の定義、レベルとして機能するテンプレート

これらコマンドのメカニズムの詳細については以下のセクションにて記述されます:

- グループ化コマンドにより、選択されたノードテンプレートにより表されるノードのインスタンスがグループ 化されます。グループ化のメカニズムについては、グループ化のセグションに記述されます。
- 並べ替えコマンドにより、選択されたノードテンプレートにより表されるノードのインスタンスが並べ替えられます。並べ替えのメカニズムについては、並べ替えのセケンコンにて記述されます。
- 変数の定義コマンドにより、選択されたノードテンプレートをスコープとした変数を定義することができます。変数の使用方法については、変数のセグションにで記述されます。
- レベルとして機能するテンプレートコマンドは、ノードテンプレート上のレベルを作成/削除するためのトグルコマンドです。ドキュメントを階層構造で構築するためには、様々な箇所でレベルを指定する必要があります。この構造は、目次や番号の自動割り当て、そしてテキストの参照を生成するために使用されます。これら機能の詳細については、<u>目次(TOC)と参照</u>.

デザインフラグメントの作成

選択されたテンプレートからデザインフラグメントのテンプレートを作成します。作成されたデザインフラグメントのテンプレートは、デザインの下部 にあるデザインフラグメントのテンプレートへ追加され、デザインソレーならび、スキーマソレーへ追加されます。作成された箇所へも、そのデザインフラグメントが適用されます。

(テンプレートや書式) タグだ! 捎|除

このコマ・ドにより、選択されたテンプレートやフォーマ・トタグガ 削除され、子孫ノードやそのフォーマ・トタグガ 削除されることはかりません。 あるタグに含まれる子孫ノードなどを削除することはレニフォーマ・トタグヤ親要素のタグを削除することができます(タグガ選択された状態で削除コマ・ドガ実行されると、これら全てが削除されます)。 親要素を削除することで、削除された要素の子孫ノードが下正となる可能性があることに注意してください。 不正なノードには取り消し線が表示されます。

編集、タグで囲む、変更

これらコマイの解説を以下に示します:

- 編集: Windows で既に馴染み深い、切り取り、コピー、貼付け、そして削除コマンドがサブメニューに表示されます。
- タグで囲む: ノードテンプレートを、以下のデザインコンポーネントにより囲むことができます: 段落、特殊な段
 落、リスト、ハイパールク、条件、目次ブグマークとレベル。
- 変更:変更機能により、(i) テンプレートが適用されるノード、または (ii) デザイン内でノードを作成する方法を変更することができます。詳細については、変更機能のセグションを参照ください。

□ 関連項目

- ◆ <% SPS% ファイルコンテンツ
- <u>テンプレートとデザインフラグメト</u>
- XPath ダイアログ

6.4.6 デザインフラグメント

デザインフラグメトは、ドキュメト内の様々な箇所で再利用することができる。~ソを作成するさかに使用することができます。以下に使用するさかの手順を示します:

- 1. デザインフラグメントのエンテンツを入力する
- 2. デザインフラグメントのコンテンツを入力する
- 3. テンプレート内にある目的の位置にデザインフラグメントを挿入する

デザインフラグメントのコンテンツを入力する

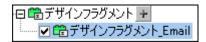
以下の操作によりデザインフラグメントを作成することができます:

1. デザインツリーならびにスキーマツリーにて、ツリー内部にあるデザインフラグメントアイテムの右側に配置されているデザインフラグメントの追加アイコン to をクリックします(以下のスクリーンショットを参照)。この操作によりツリーのデザインフラグメントリストにデザインフラグメントアイテムが追加されます(以下のメモも参照ください)。



デザインフラグメントが SPS デザイン内部に作成されます。このテンプレートはデザイン内に既にあるテンプレートへ追加されます。StyleVision のデザインフルターツールバーにある表示/隠すアイコンによりメインテンプレートとグローバルテンプレートを隠すことで、デザイン内にあるデザインフラグメントだけを確認することができます。デザインフラグメントテンプレートはスキーマツリーにも表示され、素早いアクセスを行うことができます。

2. デザインツリーまたはスキーマツリーにあるデザインフラグメントアイテムをダブルクリックすることで、その名前を編集することができます。目的の名前を入力し、Enter キーを押下してください。編集された名前がデザインツリーとデザイン内部のテンプレートに表示されます。

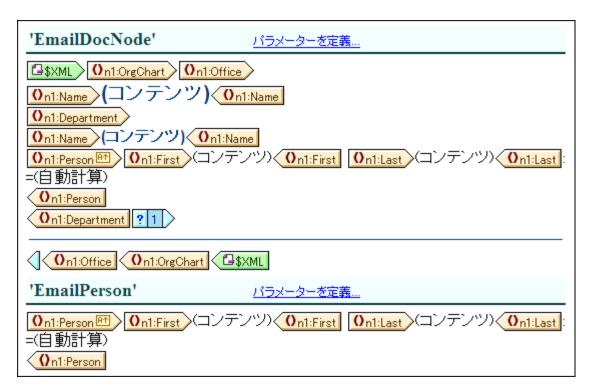


- 3. デザインにてデザインフラグメントテンプレートのコンテンツを作成します。以下ではその方法について記述されます。
- **メモ** 既存のテンプレートからデザインフラグメントを作成する場合、そのテンプレートを右クリックして、コンテキストメニューからデザインフラグメントの作成コマンドを選択します。この操作により、選択されたテンプレートからデザインフラグメントが作成されます。デザインフラグメントのテンプレートがデザイン下部にある既存のデザインフラグメントテンプレートに追加され、デザインツリーとスキーマツリーへ追加されます。この方法によりデザインフラグメントを作成すると、選択されたテンプレートもデザインフラグメントとなり、デザインソーやスキーマソーかは挿入.

デザインフラグメントのコンテンツを作成する

デザインフラグメトテンプレートのエンテンソは、その他のテンプレートと同様の方法で作成されます。静的コンテンと挿入するコま、カーソルをデザインフラグメトのテンプレート内部へ移動させ目的の静的コンテンと挿入します。動的コンテンと挿入するコま、目的のスキーマノードをデザインフラグメントのテンプレート内部ペドラッグします。

スキーマソースからノードをドラッグするには、(i) グローバル要素ツリーから、または (ii) ルート要素ツリーからノードをドラッグします。これら2つの方法には違いがあり、ノードがグローバル要素ツリーからドラッグされた場合、祖先要素無しに作成されるため、テンプレート内で使用される場合には、親のコンテキスト内部で使用される必要があります(以下のスクリーンショットにある EmailPerson デザインフラグメントを参照)。一方でルート要素ツリーからノードがドラッグされた場合、ドキュメントノードの構造とともにノードが作成され、テンプレート内部の任意の場所で使用することができます(以下のスクリーンショットにある EmailDocNode デザインフラグメントを参照)。



上のスクリーンショットでは、2つのデザインフラグメントテンプレートが示されており、Person 要素に対して同一の出力が生成されます。EmailPerson デザインフラグメントテンプレートでは、グローバル要素の Person を EmailPerson テンプレートへドラッグすることで Person ノードが作成されます。EmailDocNode デザインフラグメントテンプレートでは、ルート要素ツリーから Person ノードがドラッグされ、(ドキュメントノードを表す \$XML からの)絶対パスとともにノードが作成されます。

これらデザインフラグメントのテンプレートがメインテンプレートへ挿入される場合、EmailPerson テンプレートが Person ノードの親ノードのコンテキスト内部から呼び出される様にする必要があります。このサンプルは (マイ) ドキュメトフォルダー以下にある C: \Documents and Settings \<username>\My

Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\DesignFragments 内のEmail.sps ファイルニで確認することができます。

デザインフラグメントには、デフォルトの値とともにパラメーターを定義することもできます。個々のデザインフラグメントのインスタンスに対して、異なる値をパラメーターへ割り当てることができます。詳細については デザ<u>インフラグメントに対する デメーターを参照(たさい</u>)

デザインを完了した後には、デザインのコンポーネントがデザインツリーでも示されている点に注目してください。

テンプレート内にある目的の位置にデザインフラグメントを挿入する

デザインフラグメントを挿入するには、デザインツリーまたはスキーマツリーから目的の場所へデザインフラグメントをドラッグします。デザインフラグメントをドロップする場所は、適切なコンテキストとなるような場所でなければなりません。デザインフラグメントのコンテンツがグローバル要素から作成された場合、メインテンプレート内にある適切なコンテキストがノードの親となるようにデザインフラグメントをドラッグする必要があります。上記のデザインフラグメントのユンテンを作成するを参照ください。

その他にも、デザインフラグメントを挿入する場所を右クトックして、コンテキストメニューからデザインフラグメントの挿入を選択することもできます。

メモ デザインフラグメントがメインテンプレート内部で参照されており、デザインフラグメントの名前が後になって変更された場合、メインテンプレート内部にある参照は正しいものではなくなり、XSLT エラーが返されるよ

うになります。問題を修正するには、メインテンプレート内にあるオリジナルの参照を削除して、名前が変 更されたデザインフラグメントへの参照を新たに作成してください。

再帰的なデザインフラグメント

デザインフラグメントはそれ自体を呼び出すような、再帰的なかたちで作成することもできます。Authentic View におって無限ループを回避 するため、呼び出しの階層を制限するプログライをセナすることができます。SPS のプログライダイアログ(「ファイル プログライブ」) にある Authentic タブ内のテンプレート呼出し階層上限プレディーで値をセナすることができ、この値により、Authentic View 出力を 処理する際に再帰的なテンプレート呼び出す回数の最大値が指定されます。テンプレート呼出し階層上限プログライで指定された値を超え たテンプレートの呼び出しが行われた場合、エラーが返されます。

デザインフラグメントの削除

デザインフラグメトを削除するコよ、デザインソノーこび選択をして、デザインソノーの削除ツール・デアイコン をクリックします。

モジュールSPS におけるデザインフラグメント

SPS モジュールが別のSPS モジュールへ追加された場合、追加されたモジュール内にあるデザインフラグメントも、参照元のSPS にて利 用することができるようにはいます。モンュールSPSに関する詳細については、次のセクションを参照してください、モンュールSPS。

サンプルファイル

サンプルSPS は (マイ) ドキュメトフォルダー ある C:\Documents and Settings\<username>\My Documents¥Altova¥StyleVision2021¥StyleVisionExamples¥Tutorial¥DesignFragments にて確認すること ができます。

■ 関連項目

- その他のコンテンツ
- メインのデンプレート
- グローバルテンプレート
- デンプレートフィルター
- モジュールSPS

216 SPS 構造 XSLT テンプレート

6.5 XSLT テンプレート

XSLT ファイルを SPS へんポーすることで、ファイル内部にある XSLT テンプレートをグロー ジレデンプレートとして使用することができるようにははます。 XML ドキュメートの処理を行なっている際に、 XML ノードの1つがんポートされた XSLT テンプレート内のソードへマッチした場合、 インポートされた XSLT テンプレートが合われている関目されます。 インポートされた XSLT ファイルに名前付きテンプレートが含まれる場合、 デザイン内部にそれらを配置することができます。

メモ

• インポートされた XSLT テンプレートは StyleVision 内部で修正できません。

XSLT ファイルをインポートする

以下の操作によりXSLT ファイルをインポートすることができます:

1. デザイン概要サイドバーにて、新規 XSLT ファイルを追加... リンクをクリックします(以下のスクリーンショットを参照)。



2. ファイルを開くダイアログが表示されます。目的の XSLT ファイルを選択し、開くボタンをクリックします。XSLT ファイルがインポートされます。XSLT スタイルシートには xslimport ステートメントが追加され、デザインツリーサイドバーのXSLT テンプレート以下には、インポートされた XSLT ファイル内部に含まれる XSLT テンプレートが表示されます(以下のスクリーンショットを参照)。



SPS 構造 XSLT テンプレート 217

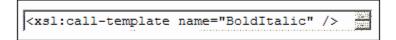
インポーされたXSLT テンプレートはお種類がます: (i) マッチテンプレート(match にいままっ)と(ii) 名前付きテンプレート(name にいままっ)。デザインソレーでは、これら2種類のテンプレートが、それぞれ(i) match テンプレートにある select 属性の値、(ii) 名前付きテンプレートにある name 属性の値とせに表示されます。

Match テンプレート

XMLドキュメトインスタンス内部にあるノードに対してテンプレートが適用され、match テンプレートが適用されるように選択されている場合にmatch テンプレートが使用されます。XMLノードの修飾名が、インポートされた match テンプレートにマッチする場合に使用されます。同じ修飾名を持ったグロー・ゾリテンプレートがSPS 内部で作成された場合、インポートされたテンプレートより提展先順位が高くなるから、そのテンプレートが使用されます。複数のXSLTファイルがインポートされた場合、最初にインポートされたファイル(XSLTコードの最初に表示されるファイル)の優先順位が最も低くなり、2番目にインポーされたファイルの優先順位が2番目に低いものとなります。

名前付きテンプレート

名前付きテンプレートは、デザインソノーからデザイン内の任意の場所へドラッグすることができます。その場所にて名前付きテンプレートを呼び出すxsl:call-template 要素が作成されます(以下のアクノーンショナを参照)。



これにより、デザイン内部で指定された場所にて名前付きテンプレートが出力に実装されます。このようなメカニズムは、XML インスタンスドキュメントや XSLT スタイルシートから独立したコンテンツを挿入する際に便利な機能となります。

6.6 複数のドキュメントの出力

SPS は、メイン出力ドキュメトとその他のドキュメトというかたちで、複数の出力ドキュメトを生成するようにデザインすることできます。この機能を使用することで、出力のモジュール化を行うことが可能によいます。新規ドキュメトテンプレートを挿入することで、出力ドキュメトをデザイン内に作成することができます(以下のスクリーンショナを参照)。各出力ドキュメトのコンテンソは、新規ドキュメトテンプレート内部に配置されます。



新規ドキュメトテンプレートは、ドキュメトデザイン内にある任意の場所で作成することができ、あらゆる階層で出力のモジュール化を行うことができます。例として、世界的な組織内部にある複数の支店が、以下の階層により個別の出力ドキュメントを持つような状況を考えてみましょう: (i) 世界、(ii) 大陸、(iii) 国、(iv) 州/県、(v) 支店。例えば、各支店を個別の出力ドキュメントにて表示し、ある国にある全ての支店を1つの文章にまとめて表示することができます。デザインでは、出力を行う個々の階層レベルにて新規ドキュメントデンプレートを作成する必要があります。適切なドキュメント構造をセットアップする方法については、新規ドキュメントデンプレートとデザイン構造のセグションに記述されます。

複数の出力ドキュメノトに関する説明は、以下にある項目ごとこ記述されます:

- 新規ドキュメントテンプレートの挿入
- 新規ドキュメント テンプレートとデザイン構造
- 新規ドキュメト テンプレートのURL
- 出力ドキュメトファイルのプレビュー
- <u>ドキュメト プロ ティとスタイル</u>

□ 関連項目

出力内のHTML フラグメント

6.6.1 新規 ドキュメントテンプレートの挿入

以下に示される2つの方法により、新規ドキュメトテンプレートをSPS デザイン内部に配置することができます:

- 新規出力ドキュメントテンプレートは、デザイン内の任意の位置へ挿入することができます。この場合、テンプレートが挿入された後に新規ドキュメントのコンテンツを追加することになります。新規ドキュメントテンプレートを挿入するには、デザイン内にある目的の位置にカーソルを配置して、メニューオプションから「挿入 | 新規ドキュメントの挿入」コマンドを選択するか、その場所を右クリックすることで表示されるコンテキストメニューから、新規ドキュメントの挿入を選択します。
- 新規ドキュメントテンプレート内部でコンテンツを囲むことで、新規出力ドキュメントをデザイン内に配置することができます。この場合、新規ドキュメントテンプレートにタグで囲まれたコンテンツが含まれることになります。デザインでは、コンテンツの追加や修正を行うことができます。コンテンツを囲むかたちで新規ドキュメントテンプレートを配置するには、対象となるコンテンツがハイライトされた状態で、メニューオプションから「タグで囲む | 新規ドキュメント」コマンドを選択するか、選択したコンテンツを右クリックして、コンテキストメニューから「タグで囲む | 新規ドキュメント」を選択します。

コンテンツを含む新規ドキュメントテンプレートを以下のスクリーンショットに示します。



上のスクノーンショナでは、以下の点に注目してください:

- 1. 新規ドキュメントテンプレートのタグには生成されるドキュメントの URL (パスと名前)が含まれます。出力フォーマットのファイルタイプにより、ファイルの拡張子が自動的に生成されます。例えば、HTML 出力フォーマットの場合、URL にあるファイル名の末尾に .html が加えられます。URL の割り当てについては、<u>新規ドキュメト テンプレートのURL のセグションに</u>記述されます。
- 2. 新規ドキュメントテンプレートには、イニシャルドキュメントセクションが1つ含まれます。

□ 関連項目

- 複数のドキュメトの出力
- 新規ドキュメト テンプレートとデザイン構造
- 新規ドキュメント テンプレートのURL
- 出力ドキュメトファイルのプレビュー
- ドキュメント プロ・ティとスタイル

6.6.2 新規ドキュメント テンプレートと デザイン構造

複数の出力ドキュメトを作成する場合、ソースドキュメト内にある適切なノードにて個別の新規ドキュメトテンプレートを作成する必要が あます。従って、複数の出力ドキュメトをデザインする際はは、<u>出力の構造</u>と入力(ソース X M L ドキュメト)の構造</u>の両方を考慮する必要があます。

メインの出力ドキュメントと追加の出力ドキュメント(出力の構造)

最初の新規ドキュメトテンプレートがデザインへ追加されると、この新規ドキュメトテンプレートの外にある全てのデザインコンテンツが別のドキュメトへ自動的に割り当てられます。この、新規ドキュメトテンプレートとは別のドキュメトがダインの出力ドキュメトとなり、Style Vision の出力プレビューではメイン出力ドキュメトとして参照されます。

(メニューオプションのファイル | 生成されたファイルの保存」コマイを使用することで、生成された出力ドキュメトでは、メイン出力ドキュメトの名前が、生成されたファイルの保存コマイドは出力ドキュメトを生成する際に割り当てられる名前とかます。メインドキュメト以外の出力ドキュメトファイルで使用される名前は、それぞれの新規ドキュメトテンプレートのURL にて割り当てられます。

新規ドキュメント テンプレートとソースドキュメントの構造

新規ドキュメントテンプレートが作成される際には、階層構造のどここそれが作成されるかが重要になります。以下にある2つのケースが考えられます:

- 1. 新規ドキュメントテンプレート内に作成されたノードが1度だけ処理される。この場合、新規ドキュメントテンプレートも1度だけ処理されます。新規ドキュメントテンプレートの URL にあるファイル名は静的な名前にすることができます。
- 2. 新規ドキュメントテンプレート内に作成されたノードが複数回処理される。その結果、ノードが処理される回数だけ、新規ドキュメントテンプレートも処理されます。以下のような例を考えてみましょう。Office 要素に(各部門を表す)複数の Department 子要素が含まれています。デザインに表示されている Department ノード内部に新規ドキュメントテンプレートが作成された場合、Department ノードが(Office 要素内にある Department 要素の数だけ)複数回処理されるため、(ソース XML にある Office 要素内の全 Department 要素に対して)新規ドキュメントテンプレートも複数回処理されます。そのため、新規ドキュメントテンプレートの URL プロパティ内にあるファイル名は、動的な名前でなければなりません。静的な名前が使用された場合、各部門を表す Department 要素に対して作成された出力ドキュメントは、全て同じ名前を持つことになります。

□ 関連項目

- 複数のドキュメントの出力
- 新規ドキュメトテンプレートの挿入
- 新規ドキュメト テンプレートのURL
- 出力ドキュメントファイルのプレビュー
- ドキュメント プロノティとスタイル

6.6.3 新規ドキュメント テンプレートの URL

このセグションでは、新規ドキュメトテンプレートのURLをデザイン構造に関連付ける方法とURLの編集方法と複数の出力ドキュメント同士でレクを作成する方法について説明します。

新規ドキュメント テンプレートのURL

新規ドキュメトテンプレートが度だけ処理される場合(前のセグシュンを参照)、テンプレートのURL プログラを静的な URL に指定することができます。以下のスクレーショナでは、ドキュメト要素(\$XML)の直下に新規ドキュメトテンプレートが配置されており、処理は1度だけ行われます。URL には Table Of Contents という静的な値が与えられており、この値が出力ドキュメトのファイル名といます。ファイル名の ペコプレフィックスは与えられていない ヤーカ、メイバキュメトと同じディンケリコファイルが生成されます(詳細については複数のドキュメトの出力とプレニーを参照ください)。その他にも、URLについか含まれる場合、出力ドキュメトが、ペコンド指定された場所へ保存されます。



その他にも、新規ドキュメトテンプレートが複数回処理されることで複数の出力ドキュメトが生成される場合(前のセケンコンを参照付さい)、テンプレートのURL プロペティをXPath 条件式を使って動的な URL とする必要があます。以下のスクレーショナでは、新規ドキュメトテンプレートのURL にbody/header/para とうXPath 条件式が使用されて、ます。新規ドキュメトテンプレートはtopic 要素内部に配置されており、topic 要素が処理されることでははます。topic 要素に対する各イタレーションに、そのtopic 要素にあるbody/header/para 要素が新規ドキュメトテンプレートのURL とて割り当てられることではます。これにより、各 topic 要素に対して新たなドキュメトが作成されることではます。各ドキュメトには、トピックヘッダーのテキストである) body/header/para 要素が得られた名前が個別に与えられます。.



URL の編集

新規ドキュメトテンプレートがデザインへ追加されると、DocumentX とら名前(X は整数)の静的なテキスト文字列をデフォルとする URL とともに作成されます。URL を編集するには、新規ドキュメトテンプレートを右クルクして、URL の編集コマイ・を選択します。プロパティダイアログが表示され、ファイル拡張なしの URL プロパティのフィールドに値を入力することができます(以下のスクノーンショナを参照)。



静的な URL を入力するづよ、値フィールドに目的のURL テキストを入力します。動的な URL を入力する場合、値フィールをクトグしたあ出、プロ ディダイアログのソール ドーある XPath ボタンをクトグして、目的の XPath 条件式を入力します。以下の点に注意してくたさい、(i) XPath 条件式のコンテキストノードは新規ドキュメト・テンプレートが挿入されたノードになります、(ii) XPath 条件式にプレフィグフを与えるづよ(例えば、concat('C:\MyOutput\Y', body\Header\Para) といずっかっちで、XPath のconcat() 関数を使用してくたさい。この例にある条件式により、C:\MyOutput\Yfilename とう URL 文字列が生成されます。出力フォーマトに従い、適切なファイル拡張子が自動的に生成されます。

ドキュメントへのノンク

ブックマークとハイパーレクを使用することで、複数の出力ドキュメト間にレクを作成することができます。ブックマークは、新規ドキュメントテンプレートの〜ダーを含む新規ドキュメトテンプレート内の任意の場所に配置することができます。その後、他のドキュメトから作成したブックマークへの、イパーレクを作成することができるようによります。複数回処理されるノード上にブックマークを作成する必要がある場合、ブックマークの名前が動的に生成されるようにする必要があります。そうでは、場合(静的なブックマーク名が与えられた場合)、出力に与えられる複数のノードが同じ名前のブックマークを持つことしています。

<u>目次(TOC)</u>もドキュメントへのレクとして使用することができます。TOC を独立したドキュメント(例えばメインドキュメント)内に配置して、様々な出力ドキュメントに対してレクを作成し、それぞれの出力ドキュメントからTOC へ戻るレクを作成することができます。

- 関連項目

- 複数のドキュメトの出力
- 新規ドキュメントテンプレートの挿入
- 新規ドキュメト テンプレートとデザイン構造
- 出力ドキュメントファイルのプレビュー
- <u>ドキュメント プロッティ と スタイル</u>

6.6.4 プレビューファイルと出力 ドキュメントファイル

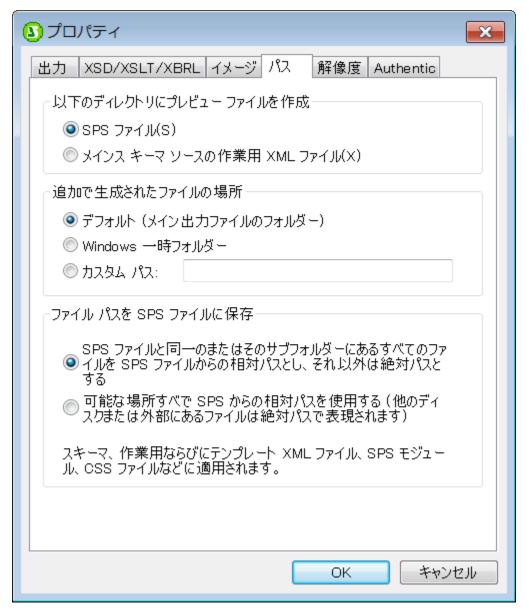
デザインドキュメントの出力プレビューでは、デザインにで個別のドキュメントとして指定された各出力ドキュメントを確認することができます。(以下のスクノーンショントを参照)。



上のスクノーンショナでは、複数の出力ドキュメトを生成するようにデザインされたSPS ドキュメントのプレビューが示されています。各出力ドキュメントへは(i)左上に表示されている矢印ボダンをクトックすることで、まざは(ii)コンボボックスのドロップダウンノストから目的のドキュメントを選択することでアクセスすることができます(上のスクノーンショナを参照)。ドロップダウンノストのアイテムコは、URL 全体(パンとファイル名)が表示される点に注目してください。

プレビューファイルの場所

プレビューファイルは、デフォルで SPS ファイルが作成されるデルノトリニで作成されます。デフォルの設定は、(メニューオプションの「ファイル | プロパティ」にて表示される) SPS ファイルのプロ ティダイアログにある マタブ こて変更することができます(以下のアクレーシショットを参照)。このタブでは、作業用 XML ファイルの場所を指定することができます。新規ドキュメントテンプレートのURL に マか管まれている場合、この マで指定された場所が、そのプレビューファイルの格納に使用される場所となります。指定された場所が見つからない場合、エラーが返されます。出力ドキュメント間のアンケを作成する場合、それぞれの出力ファイルがどこで作成されるかを考慮する必要があります。



プロ ティダイアログの マタブでは、出力ドキュメントファイル、イメージやチャートイメージファイルと ソナンプレビューを行うかがに一時的に使用されるファイルの配置場所を指定することもできます。 新規ドキュメントテンプレートの URL についる 含まれる場合、そこで指定されたのです。 用される点に注意してください。

出力(パス)の生成

以下の操作により、出力ドキュメントファイルを生成することができます:

- 1. メニューオプションの「ファイル | 生成されたファイルの保存」コマンドにマウスを移動させ、目的の出力フォーマットを選択します。
- 2. 生成されたファイルの保存ダイアログが表示されるので、生成されたファイルを保存するフォルダーを選択します。
- 3. メインドキュメントファイルの名前を入力し、保存ボタンをクリックします。

出力ドキュメトファイル イメージファイルやチャートのイメージファイルといっさその他生成されるファイルの場所が、ポップアップウインドウェて表示されます。

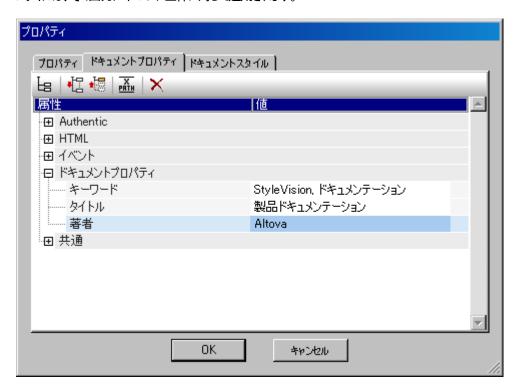
生成されたファイルの保存ダイアログにて選択したフォルダーに、メインドキュメントファイルが保存されます。新規ドキュメントテンプレートにて作成されたその他全てのドキュメントで、URL についてが含まれていないドキュメントも、メインドキュメントファイルと同じフォルダー内に保存されます。新規ドキュメントテンプレートのURLにていいない入力された場合、URLにて指定された場所に出力ドキュメントが保存されます。指定されたフォルダーが存在しない場合、エラーが返されます。

- 関連項目

- 複数のドキュメトの出力
- 新規ドキュメントテンプレートの挿入
- 新規ドキュメント テンプレートとデザイン構造
- 新規ドキュメント テンプレートのURL
- ドキュメント プログラィとスタイル

6.6.5 ドキュメント プロパティと スタイル

SPS デザインでは、出力を複数のドキュメトへ分割することができます。これらの各ドキュメトには、異なるドキュメトプロ・ティならびば キュメト スタイルを割り当てることができます。これらの値は、各ドキュメトのイニシャルドキュメトセグションからアクセスできるプロ・ティダイアログのドキュメトプロ・ティならびば キュメトスタイルタブ こて指定することができます。 プロ・ティダイアログへアクセスする コは、編集を行うドキュメトのイニシャルドキュメトセグションのタイトル・ドーさるプロ・ティの編集リングをクリックしてください。 ドキュメトプロ・ティど キュメトスタイルは、その出力ドキュメト全体に対して適用されます。



ドキュメントプロ・ティタブでは、プロ・ティクバキュメントプロ・ティグループにてドキュメントのメタ情報を入力することができます。このメタ情報は対応する出力ドキュメントと、出力フォーマットに対応したプロ・ティに保存されます。例えば、HTML 出力フォーマットの場合、HEAD 要素のMETA タグにてプロ・ティが保存されます。

 226
 SPS 構造
 複数のドキュメントの出力

ドキュメトスタイルは関する詳細にないては、CSSプロンティ値の設定のセグションを参照くたさい。

□ 関連項目

- 複数のドキュメントの出力
- 新規ドキュメトテンプレートの挿入
- 新規ドキュメト テンプレートとデザイン構造
- <u>新規ドキュメト テンプレートのURL</u>
- 出力ドキュメトファイルのプレビュー

高度な機能 227

7 高度な機能

SPS デザインの基本的なコンテンソ代構造の作成方法については、SPS ファイルエンテンツとSPS ファイル構造 にて記述されます。しかし、ソースデータにあるコンテンソ代構造を特定の形式に修正しなければならないとも多々あります。例えば(個人情報が含まれる)ノードのグループを、苗字といざ、特定の条件により述べ替えるケースなどが考えられます。他にもデータベース内にある顧客情報を都市ごとでグループ化して、都市ごとの製品売上額を確認するようなケースも考えられます。StyleVision には以下で説明されるような検索機能が搭載されています。

StyleVision のSPS ファイルコ搭載された機能を以下に示します:

- <u>自動計算</u>: 自動計算とはXPath をベースにした。プラルはメカニズムで、(i) 修正されたデータを出力にて表示するだけでなく、
 (ii) 自動計算の結果により、XML ドキュメト内にあるノードを更新することができます。
- 条件: データ構造や、XML の値、ませは XPath 条件式の結果からなる条件により、テンプレートとテンプレートのエンテンツ処理を行うことができます。
- グループ: XPath 条件式により選択された要素のグループで対して処理を定義することができます。
- 並へ替え: XML 要素のセナを複数のノートキーにより述べ替えることができます。
- パラメーターと変数: グロー・いな SPS レベルでデフォルト値を持っすっ、ウメーターを宣言することができます。これらの値はランタイム にてオー・ドーライドすることができ、コマイドラインから値を受け渡すこともできます。 SPS 内部では変数を定義することができ、これらの変数を SPS 内部から参照して使用することができます。
- <u>目次(TOC)と参照</u>: 目次(TOC)は全ての出力フォーマナトにて、かがキュメナの様々な場所にて作成することができます。 TOC は、最初に目次にて参照されるアイテムを選択し、これらのアイテムを目次にて参照することにより作成されます。参照を使用するその他の機能には、(i) <u>テキスト参照</u>(ドキュメナト内にあるテキストをマークして、ドキュメナトの他の場所から参照を行う)、そして(iii) <u>ブックマークと・ハイパートンク</u>(ブックマークのキーにより、ハイパートンクの参照先を作成する)があります。・ハイパートンクは対ト部のアノースに対しても使用することができます。これらの参照メカニズムにつも、このセクション以下で説明されます。

□ 関連項目

- SPS ファイルゴンテンツ
- SPS ファイル構造
- SPS ファイル追加機能

7.1 自動計算

自動計算機能により、(i) 出力ドキュメト内にある任意の場所でXPath の評価結果を表示し、(ii) オプロンとして、XPath の評価結果を使ってメインXMLドキュメト内にあるノードの更新を(XMLドキュメトがAuthentic View にて編集されている時に) 行うことができます。

自動計算機能は、以下のような処理を行う際に便利なメカニズムとなります。

- 動的なデータ値を伴うオペレーションの計算を挿入する: 例えば、Office 要素内にある Employee 要素の数を (count(Employee) により)数えることができるほか、各 Invoice 要素内にある Price 要素の値の合計を(sum(Price) により)計算し、Person 要素の FirstName と LastName 要素の値を(concat(FirstName, ', LastName) により)統合することができます。このような操作により、動的に変化する XML ドキュメント内にあるデータから新たなデータを 動的に生成して、生成されたデータを出力にて表示することが可能になります。
- ドキュメントのデータ構造から得られた情報を表示する: 例えば、XPath の position() 関数を使用することで、動的テーブルに行番号を動的に挿入し、ドキュメントのセクションに動的な番号を割り振ることができます。この機能を使用することで、動的に変化するドキュメント構造をベースにした情報を自動的に生成することが可能になります。
- 外部 XML ドキュメントから得られたデータを挿入する: XPath 2.0 の doc() 関数を使用することで、外部 XML ドキュメントのドキュメントルートへアクセスすることができるようになり、外部 XML ドキュメントのノードコンテンツを出力内に挿入することが可能になります。.
- ずザイン内部の任意の場所でノードのコンテンンを表示する

□ 関連項目

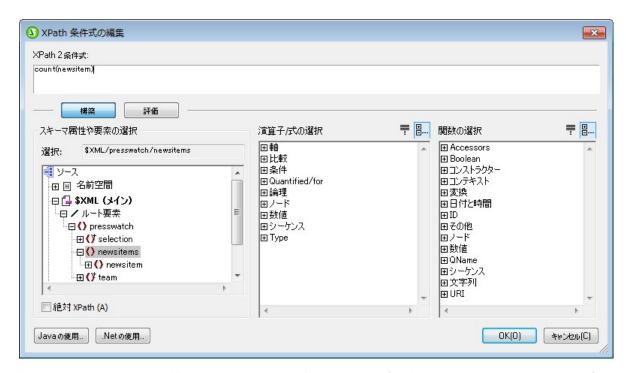
- 挿入 自動計算
- XPath ダイアログ
- XSLT エンジンの情報

7.1.1 自動計算の編集と移動

自動計算の作成

以下の操作により、自動計算を作成することができます::

1. 自動計算の結果が表示される挿入ポイントへカーソルを配置して、メニューオプションから「挿入 | 自動計算を挿入」を選択します。計算結果がプレーンテキストとなる場合、表示されるサブメニューから値を選択し、入力フィールド(テキストボックス)内に結果を表示する場合には入力フィールドを、複数行のフィールドに計算結果を表示する場合には複数行フィールドを選択します(自動計算の結果は値、または入力フィールド内部に表示されますが、Authentic View における出力であり、編集を行うことはできない点に注意してください)。XPath 条件式の編集ダイアログが表示されます(以下のスクリーンショットを参照)。



2. 条件式ペインにて、自動計算に使用される XPath 条件式をキーボードから入力します。その他にも、ダイアログの対応するペインにあるノード、演算子、そして関数をダブルクリックすることでも入力を行うことができます。挿入ポイントとなるコンテキストノードにも留意する必要があります。ダイアログが表示される時に、コンテキストノードはスキーマソースツリーにてハイライトされた状態で表示されます(上のスクリーンショットでは、newsitems がハイライトされています)。 SPS で使用される XSLT 言語のバージョンに XSLT 1.0 を選択した場合、XPath 1.0 条件式を使用する必要があり、XSLT 2.0 を選択した場合には、XPath 2.0 を使用する必要があります。 XPath 条件式の編集ダイアログに関する詳細については、 XPath 式を編集セグシュを参照付さい。

「OK」 ボタンをクトックすることで、XPath 条件式の編集を完了することができます。 自動計算のシンボルカデザインビューに表示されます。 自動計算の結果を確認するコよ、HTML ビュー

自動計算の編集

自動計算のXPath 条件式を編集するコよ、自動計算が選択された状態でプロ、ティサイドバーの自動計算グループにあるXPath プロパティの編集ボタンをクトックしてください、以下のスクリーシンコナを参照)。この操作により、XPath 条件式を編集するためのXPath 式の編集ダイアログ(上のスクリーンショナ)が表示されます(上のスクリーンショナを参照)。



自動計算のフォーマト

定義済みのフォーマトやCSS スタイルは、通常のデキストと同様の方法で自動計算に対して適用することができます。自動計算を選択し、フォーマトを適用します。更に、プロ・ディウイドウの自動計算グループにある入力フォーマトプロ・ディでは、数値や日時と、ケイデータ型を自動計算の入力フォーマト として指定することができます。

XPath 条件式には次行を含めることもできます。自動計算がpre タグにより囲まれている場合、改行が出力にて表示されます。このような XPath 条件式の例を以下に示します:

translate('a;b;c', ';', codepoints-to-string(13))

自動計算の移動

自動計算をクトックして選択し、新たむ場所へドラッグすることで、自動計算を別の場所に移動することができます。更に、自動計算の切り取りペコピー、貼付けべ移動を行うこともできます。新たむ場所のエレテキストノードが異なる場合、XPath 条件式を変更する必要がある点に注意してくたさい。

重要なポイト

以下の点に注意してください

- 自動計算はデザイドキュメト内にある任意の場所に挿入することができます。
- 自動計算を挿入する箇所により、自動計算で使用されるXPath 条件式のエレテキストノードが決定されます。

□ 関連項目

- 挿入 自動計算
- XPath ダイアログ
- XSLT エンジンの情報
- サンプル注文書

7.1.2 サンプル: 注文書

(マハドキュメトフォルダー以下のC:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\Auto-Calculations\フォルダーにあるいできます:

- ノードの数を数える
- Authentic View ユーザーからの入力に従いノードを選択する
- 複雑な計算を作成する

サンプルファイルでは、自動計算が黄色の背景色によりイライトされます(以下のスクノーンショナを参照)。

ノードの数を数える

注文書のサンプルでは、リストにある製品が、顧客により指定されたリスト内にある順字に従い番号付けされます(Product 1 やProduct 2 など)。この番号付けは、自動計算により行われます(以下のスクノーンショナを参照)。

Product 1:	Learning XMLSpy
Net price:	€ 35.00
Category:	Book 🔻
VAT:	<mark>10</mark> %
Price including VAT:	€ <mark>38.5</mark>
Product <mark>2</mark> :	Scooby Doo's Greatest Hits

この場合、XPath 条件式のposition() 関数により番号を取得することができます。その他にも、それまで処理された兄弟ノードの数を数え、1 を加えることでも番号を取得することができます。この場合 count(preceding-sibling::Product)+1 とり XPath 条件式が使用されます。後者の方法は、position() 関数を使用するのが難しいような(まだは不可能な)状況で使用することができます。製品を削除し、新たな製品を追加することで、サンプレファイル内にある自動計算の動作を確認することができます。

ユーザーからの入力に従いノードを選択する

注文書のサンプルでは、ユーザーがエンボドンケスの値を選択することで、製品のカテゴリー(Book、CD、DVD、まだはElectronics)が選択されます。この選択結果は、XMLドキュメントの//Product/Category 内に入力されます。自動計算は、この値を使用することでXMLドキュメントへの参照を行い、対応する製品カテゴリーのVAT 税率が取得されます。この自動計算のXPath 条件式は以下のようにはます。

for \$i in Category return /Invoice/Categories/Category[. = \$i]/@rate.

出力では、自動計算の位置にVATの税率が表示されます。注文書のサンプルでは、注文書のデータが保管されるXMLドキュメトに
ルッケアップテーブルも保管されます。このようなテーブルは別のドキュメトにて管理することも可能で、その場合、XPath 2.0 のdoc() 関数によりアクセスすることでははます。製品のCategory コンボドックスこで製品の種類を変更し、それぞれの種類に対して異なるVATの値が表示されることで、自動計算の動作を確認することができます(Book=10%、CD=15%、DVD=15%、Electronics=20%)。

複雑な計算を作成する

上の例で自動計算により取得されたVATの税率から、各製品に対して販売価格(本体価格 + VATの税金額)を計算する必要があります。使用する式は以下のようになります。

販売価格 = 本体価格 + *VATの税金額 VATの税金額* = 本体価格 * *VATの税率 / 100* となるため
販売価格 = 本体価格 + (本体価格 * *VATの税率 / 100*)

製品の本体価格はPriceNet ノードから取得することができます。VAT の税率は上記の方法を使って自動計算により計算され、この自動計算により、VAT ノードが更新されます。従って、VAT ノードのコンテンンを自動計算に使用することで、販売価格を計算することが可能になります。以下のXPath 条件式を使用することができます:

XPath 条件式は、プログライウインドウから確認並びに編集することができます。本体価格や製品のカテゴリーを変更することで、自動計算が動作していることを確認することができます。製品のXVATを含む)販売価格が変わることに注目してください。

Product 6:
A Short History of the American Century

Net price:
€ 20.00

Category:
DVD

VAT:
15%

Price including VAT:
€ 23

■ 関連項目

- 自動計算の編集と移動
- XPath ダイアログ

7.2 条件

メインテンプレートとグロー・バリテンプレー・の両方で、デザイン内部の任意の場所に条件を挿入することができます。条件とはつ以上の分岐から構成される SPS エポーネトのことで、各分岐がXPath 条件式により定義されます。例えば、2つの分岐からなる条件を考えてみましょう。最初の条件分岐にある XPath 条件式により、エンテキストノード上にある Location 属性の値が "US" であるかがデェックされます。 2番目の条件分岐にある XPath 条件式では、Location 属性の値が "EU" であるかがデェックされます。各分岐には、テンプレート(条件テンプレート)が含まれています。条件を伴うレードが処理され、結果が真となる最初の条件(条件テンプレート)が処理されます。条件内にある処理が終了すると、その他の分岐にある条件は評価されません。この機能を使用することで、ノードの値により異なるテンプレートを使用することができます。

このセクションノコよ 以下のトピックか含まれています:

- ◆ 条件のセナアップ 条件と分岐の作成方法について記述されます。
- 条件の編集 条件分岐が作成された後に、XPath 条件式を編集する方法について記述されます。
- 条件と自動計算:条件と自動計算を組み合わせて使用する際に注意すべき点について記述されます。
- □ 関連項目
- クイックスタート チュートリアル 条件を使用する

7.2.1 条件のセットアップ

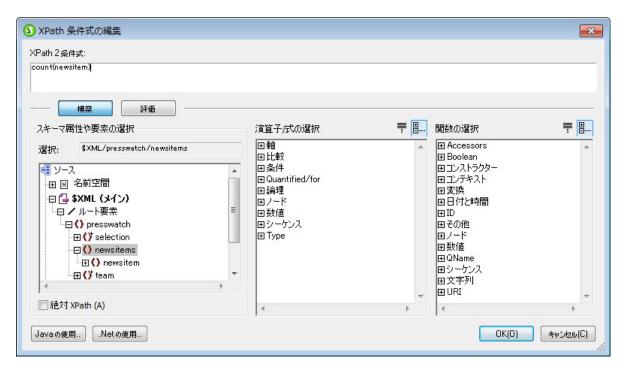
以下にあるステップにより、条件を作成することができます:

- 1. 最初の分岐とともに、条件を作成する。
- 2. 必要に応じて新たな条件分岐を作成する。
- 3. 条件の各分岐内にテンプレートを作成し、編集する。

最初の分岐と共に条件を作成する

以下の操作により条件をセナアップすることができます。

1. デザイン内の任意の場所にカーソルを配置するか、コンポーネントが選択された状態で、メニューオプションから「挿入 | 条件の挿入」を選択します。XPath 式 の編集ダイアログ か表示されます(以下のスクノーンショナを参照)。



- 2. 条件式ペインでは、その条件分岐に対する XPath 条件式をキーボードから入力することができます。その他にも、ダイアログのペインにあるノード、演算子、または関数をダブルクリックすることで、それらを入力することができます。ノードが挿入されたコンテキストノードについて留意する必要があります。コンテキストノードは、ダイアログが表示された時にスキーマソースツリーにてハイライトされます。
- 3. 「OK」をクリックして完了します。最初の条件分岐とともに条件が作成され、先程作成した XPath 条件式が最初の条件分岐に使用されます。何も選択されていない状態で条件が挿入されると、最初の分岐が空の状態で表示されます(テンプレートが無い状態で表示されます 以下のスクリーンショットを参照)。コンポーネントが選択された状態で条件が挿入されると、コンポーネントの周りに条件が作成され、そのコンポーネントが最初の条件分岐におけるテンプレートとなります。

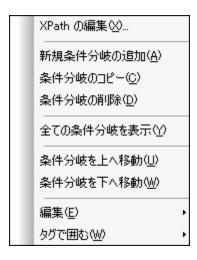


条件全体を選択するには、クエッションマーク(?)が表示されているセルをクリックします。最初の分岐を選択するには、1という番号が表示されているセルをクリックします。

最初の分岐を伴う条件を作成すると、(その分岐内にテンプレートが存在するかにかかわらず)必要な数だけ条件分岐を作成することができます。

新たな条件分岐を作成する

条件分岐は新たに1つずつ作成することができます。新たな分岐はコンテキストメニューから作成することができ、(i) テンプレートが含まれていない状態で作成(新規条件武器の追加)、または(ii) 既存のテンプレートをコピーして新たな条件分岐を作成(条件分岐のコピー)することができます(以下のスクリーンショットを参照)。



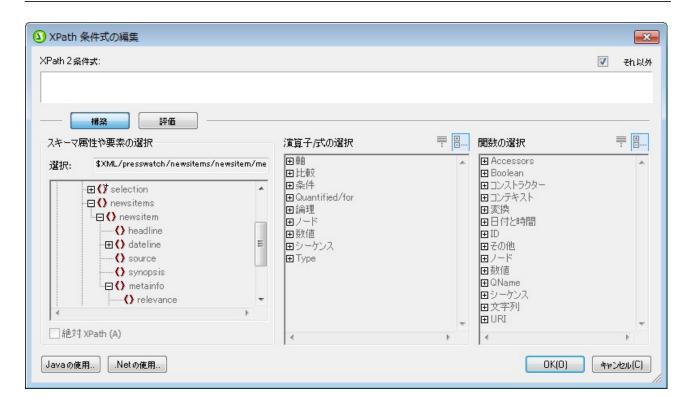
新たな分岐を作成するには、条件分岐のどれかを右クリックして、コンテキストメニューから新規条件分岐の追加を選択します。 XPath 式 の編集ダイアログ が表示されます。 XPath 条件式を入力して「OK」をクリックすると、空の分岐が条件へ新たに加えられます。 新たな条件分岐が追加されると、条件に新たなセルが追加され、追加された条件には、追加される前に表示されていた分岐に 1 が加えられた数が表示されます。

既存の条件分岐からのコピーを作成するには、コピーする条件分岐を右クリックして、コンテキストメニューから条件分岐のコピーを選択します。コピーされた条件分岐を含む XPath 条件式とともに、 XPath 式 の編集ダイアログ が表示されます。XPath 条件式の編集を終え、「OK」をクリックすると、新たな分岐が条件に追加されます。新たな条件分岐には、コピー元の分岐に含まれていたテンプレートが含まれます。追加された条件には、追加される前に表示されていた分岐に 1 が加えられた数が表示されます。

それ以外の分岐

それ以外の分岐には、他のどの条件分岐も処理されなかった際に処理されるテンプレートが含まれます。その他の条件分岐が定義されていない場合、考えられる全ての事象に条件分岐を対応させるか、どの分岐も処理されないまま条件の処理が終了しても良いようにデザインを作成する必要があります。

その他の条件分岐を挿入するには、上に記されているように新規条件分岐を追加するか、条件分岐のコピーを行い、XPath 式の編集ダイアログにて、それ以外のチェックボックスにチェックを入れます(以下のスクリーンショナを参照)。



分岐の優先度を移動する

条件では、最初に真と評価された分岐が処理され、それ以外の分岐は処理されないか、条件内にある分岐の順序が重要な意味を持つことでいます。分岐の位置を変更するには、移動する分岐を選択し、右クックには、表示されるエンテキストメニューが条件分岐を上へ移動ませば条件分岐を下へ移動を選択します。

分岐の削除

条件分岐を削除するコよ 削除する分岐を選択し、右クトケで表示されるエンテキストメニューが条件分岐の削除を選択します。

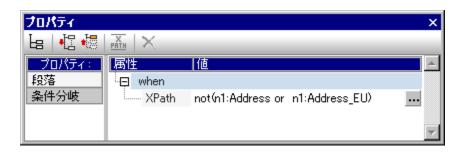
三 関連項目

- クイックスタート チュートリアル 条件を使用する
- 条件の編集

7.2.2 条件の編集

以下の操作により、条件分岐のXPath 条件式を編集することができます:

- 1. (条件ではなく)条件分岐を選択します。
- 2. プロパティサイドバーにて、条件分岐を選択します(以下のスクリーンショットを参照)。



□ 関連項目

- クイックスタート チュートリアル 条件を使用する
- 条件のセナアップ

7.2.3 条件と自動計算

条件と自動計算を組み合わせて使用する場合、いくつか留意するべきことがおます。重要な点を以下に示します:

- 表示されている条件(つまり真として選択された分岐)にある自動計算だけが評価されます
- 自動計算は条件の前に評価されます

上記の点をガイダンスとしてまとめたものを以下に示します:

- 1. 自動計算によりノードが更新され、そのノードが条件に含まれている(条件分岐の XPath 条件式に含まれているか、条件添付がのコンテンツ内に含まれている)場合、自動計算ができる限り条件の外で行われるようにしてください。これにより、自動計算が常に(つまりどの条件分岐が表示されているかに関わらず)表示されるようになり。自動計算が条件分岐の中にあり、その分岐が表示されない場合、。
- 2. 自動計算を条件の内部に配置する必要がある場合、(i) それが条件内にある全ての分岐に配置されるようにして、(ii) 全ての状況に対して条件の分岐が対応できるようにする必要があります。条件テンプレートにてカバーされない事象が仮に存在する場合、(自動計算が表示されているテンプレートに配置されないことにより)自動計算が行われないというリスクが発生します。
- 3. 条件に応じて異なる自動計算を使用したい場合、全ての分岐において自動計算が使用されていることを確認してください。
- 4. 条件テンプレートにて条件の順序は重要な意味を持つことに留意してください。最初に真と評価された条件分岐が実行されます。それ以外の条件は、定義された条件分岐が全て偽と評価されたときに実行されます。

- 関連項目

- クイックスタート チュートリアル: 条件を使用する
- 条件の編集
- 自動計算

7.3 グループ化

グループ化関数は XSLT 2.0 と3.0 SPS にて利用することができ、HTML 出力に対して適用されます。

グループ化により、アイテム(大抵の場合ノード)をグループ単位で処理することができるようになります。車の在庫管理を例に考えてみましょう。各車の詳細情報は、car 要素以下に収められます。例えば、car 要素にbrand 属性がある場合、ブランド名により庫をグループ化することができます。この機能により、以下のような処理を行うことが可能になります。

あるブランドの車全てをまとめて出力に表示し、その上部にブランド名のヘッダーを表示する。 グループ内部で処理を行うことにより、その処理結果をグループごとで個別に表示する。例えば、各ブランドで 利用可能な車種がどれだけあるかを表示することができます。

更に、グループ内に配置されたサブグループ単位で処理を行うこともできます。例えば、各ブライト内にあるモデルが年式ごとにグループ化を行うことができます。

グループ化の条件

アイテムよ(i) 通常ノードの値をチェックするために使用されるグループ化キー、そして(ii) 相対的なアイテムの位置という2つの条件によりグループ化することができます。以下にあるグループ化の条件を使用することができます:

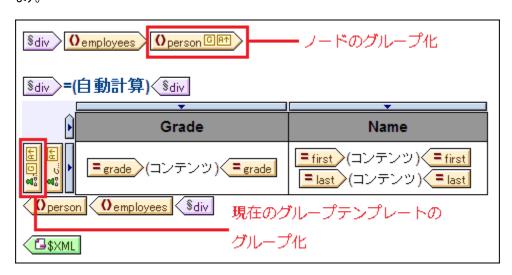
- group-by: XPath により定義されたキーをベースにアイテムのグループ化を行います。例えば、car 要素を、brand 属性をベースにグループ化することができます。car 要素に対してグループ化を行い、XPath 条件式によりbrand 属性が選択されます。
- group-adjacent:グループ化のキーと位置情報を組み合わせることでグループ化を行います。同じグループ化のキーを持った隣合わせのアイテムがグループ化の対象になります。あるアイテムにあるグループ化のキー値が、前にあるアイテムの値と違う場合、そのアイテムから新たなグループが作成されます。
- group-startup-with: 定義された XPath にマッチするノードが見つかると、新たなグループを作成します。定義された XPath パターンにノードがマッチしない場合、そのノードが現在のグループに追加されます。
- group-ending-with: 定義された XPath にマッチするノードが見つかると、そのノードがグループ最後のノード となります。次に処理されるノードが新たなグループ最初のノードになり、その次に処理されるノードが定義された XPath パターンにマッチしない場合、そのノードがグループに追加されます。

グループの作成

グループは、コンテキストメニューを介してノード、まけは現在のグループテンプレートにて作成することができます。グループを作成するコは、ノードまけは現在のグループテンプレートを右クトグして、コンテキストメニューからグループ化コマンドを選択します。出力グループを作成ダイアログが表示されます(以下のスクリーンショナを参照)。



ダイアログでは、グループ化を有効のチェックボックスニチェックを入れ、必要となるグループ化の種類をラジオがなっから選択します。 XPath 条件式を入力することには、グループ化キーを定義(group-by ならびにgroup-adjacent の場合)することができるまか、マッチを行うかめの、サーンを入力(group-starting-with ならびにgroup-ending-with の場合)することができます。「OK」をクトックすると、グループセットをアルファベット順(昇順)で並べ替えるか選択するよう促されます。グループセットの並べ替えば後から変更することができるほか、並べ替えを削除することもできます。以下のスクノーシショットでは、ノードと現在のグループテンプレートからなるグループ化の様子が示されます。



上のスクレーシンコナでは、person ノードがブループ化され、作成されるグループはアルファベナ順に並べ替えられます。例えばperson 要素がdepartment によりグループ化された場合、アルファベナの昇順によりdepartment 要素の並べ替えを行うことができます。

グループの並べ替え

グループの定義を完成させると、アルファベナの昇順によりグループの並へ替えを行うか尋ねられます。グループの並へ替えはいつでも行うことができるほか、ここで作成した並へ替えの設定を削除することもできます。

並べ替えの設定、修正、まけば削除を行うけは、目的のグループ化テンプレートを右クソグして並べ替えを選択します。この操作により出力の並べ替えの順序を定義ダイアログが表示されます。このダイアログの使用方法については、並べ替えのセグションを参照ください。グループ化キーによりグループの並べ替えを行うけは、並べ替えキーにcurrent-grouping-key() XPath 関数を使用しなければなりません。詳細については、このセグション以下にあるサブセグションを参照ください。

グループ化と並べ替え設定の確認と編集

テンプレートにおけるグループ化ならびに並べ替え設定を確認、まけは編集するけま、テンプレートを右ケックして、グループ化まけば並べ替えをそれぞれ選択します。該当するダイアログが表示されるか、設定の確認や修正を行うことができます。

ユーザー定義テンプレート

ユーザー定義アンプレート により、指定したXPath 条件式により選択されるアイテムコオしてテンプレートを適用することができます。ユーザー 定義関数のXPath 条件式により選択されたノードをグループ化することもできます。この場合、グループ化はユーザー定義テンプレートに対して適用されます。

□ 関連項目

- 出力の構造
- 並べ替え
- ユーザー定義デンプレート
- サンプル Group-By (Persons.sps)
- サンプル Group-By (Scores.sps)

7.3.1 サンプル: group-by(Persons.sps)

Persons.sps サンプルはPersons.xsd スキーマをベースに作成されておりPersons.xml を作業用 XML ファイルとて使用します。ファイルは<u>(マイ)ドキュメトフォルダー</u>以下にある C:\Documents and Settings\<username>\My
Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\Grouping\Persons\「名納されてい

Documents (Altova (Style Vision 2021 (Style Vision Examples (Tutorial (Grouping (Persons (**Pany Altova**) ます。ドキュメト要素のemployees 要素には任意の数のperson 要素を含めることができ、各 person 要素は以下に示されるようは構造を持ちます:

<person first="Vernon" last="Callaby" department="Administration" grade="C"/>

デザインではdepartment (部門)によるグループ化を行います。各部門は独立したテーブルにより表示され、部門はアルファベナル順にて並べ替えられます。各部門テーブル内部では、person により表される従業員が与えられた権限 (grade) におじてグループ化が行われ、各 gradeでは従業員 (person) が苗字 (last) により並べ替えられて表示されます。

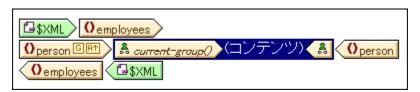
グループの作成の手順

現在のグループはテーブルとして作成されます。グループ化されたperson 要素内部にて、各グループがgrade 属性により更にグループ化され るように指定します。

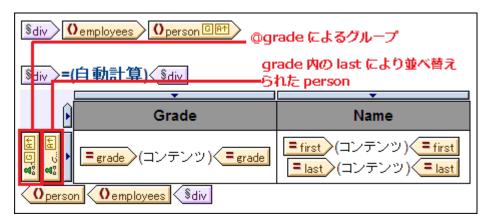
SPS の作成

以下の操作によりデザインを作成します:

- スキーマツリーから person 要素をドラッグし、コンテンツとして作成します。
 person 要素のタグを右クリックして、コンテキストメニューからグループ化を選択します。
- 3. 出力グループを定義ダイアログにて、group-by を選択し、マッチテキストボックスにて @department という XPath 条件式を入力し、はいをクリックします。
- 4. グループの並べ替えを行うかを尋ねるダイアログが表示されます。この例ではデフォルトのアルファベット順に よる並び替えを行うので、はいをクリックします(並べ替えはいつでも設定または削除することができます)。
- 5. 各グループ(この例では department)を独立したテーブル内に作成するため、カレントグループをテーブルとして 作成します。以下のスクリーンショットに示されるように current-group() タグを右クリックして、「変更 | テーブ ル」を選択し、子属性の @last と @grade をテーブルのカラムとして選択します。



- 6. 最初のカラムに @grade が、2番目のカラムに @first と @last ノードが含まれるようにテーブルにあるカラムとセル のコンテンツを修正します(以下のスクリーンショットを参照)。
- 7. department によりグループ化されたカレントグループにて grade によるグループ化を行うため、current-group() テンプレート上にて grade 属性に対するグループを作成します。デフォルトの並べ替えにより作成の確定を行います。current-group() ナンプレートが新たに作成されます。マスクリーンショットを参照)。
- 8. grade により並べ替えられた person のサブグループとなるカレントグループを、last 属性に対して並べ替えます。



- 9. テーブルのフォーマットをセットします。
- 10. テーブルの上にテーブルのヘッダーを作成します。個々のテーブルにより部門 (department) が表示されるため、XPath により記述された自動計算をカレントコンテキストで使用することで、部門の名前を動的に取得することができます。XPath 2.0/3.0 の current-grouping-key() 関数を使用することができます。
- 11. 同様の処理を最初から行い、別の出力を作成します。今度は grade により person のグループ化を行い、その中 で department によりグループ化を行います。

テンプレートのグループ化や並べ替えを確認ならびに修正するコよ、そのテンプレートを右クトックして、コンテキストメニューからグループ化ま たけなが、替えを選択します。この操作により対応するダイアログが表示され、設定を確認または修正することができます。

関連項目

- 出力の構造
- 並べ替え
- サンプル: Group-By (Scores.sps)

7.3.2 サンプル: group-by(Scores.sps)

Scores.sps サンプルはScores.xsd スキーマをベース ご作成されており、Scores.xml を作業用 XML ファイルとして使用します。ファイルは(マインドキュメトフォルダー 以下にある C:\Documents and Settings\<username>\My

Documents Altova Style Vision 2021 Style Vision Examples Tutorial Grouping Scores 一格納されてます。ドキュメナ 要素のresults 要素には任意の数のgroup 要素とmatch 要素を含めることができます。各 group 要素には任意の数のteam 要素が含まれており、match 要素に以下に示されるようは構造を持ちます。

デザインは3つのペーツから構成されます(以下のスケリーシンコナを参照)。(i)//match/@date によりブループ化された日付ごとの試合結果、(ii)//match/@group によりブループ化されたナーグブループごとの試合結果、そして(iii) group 要素から値か動的に生成され、自動計算により必要なデータが自動的に計算されるテーブルにより、リーググループの要約が表示されます。

Match Results: Day-by-Day

2007-10-12

Brazil - Germany 2 - 1 Italy - Holland 2 - 2

2007-10-13

Argentina - France 2 - 0 England - Spain 0 - 0

Match Results: By Group

Group A

 Brazil - Germany
 2 - 1

 Italy - Holland
 2 - 2

 Brazil - Italy
 1 - 2

 Germany - Holland
 2 - 2

 Brazil - Holland
 1 - 0

 Germany - Italy
 1 - 1

Group Tables

Group A

Team	P	W	D	L	F	Α	Pts
Brazil	3	2	0	1	4	3	6
Italy	3	1	2	0	5	4	5
Germany	3	0	2	1	4	5	2
Holland	3	0	2	1	4	5	2

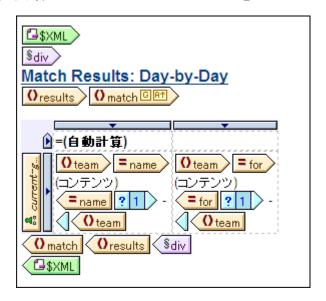
グループの作成手順

試合結果が表示される2つのセグションでは、日付とトーナメントグループによ、試合の結果をグループ化します。データの配置を行うする、日付とトーナメントグループによい作成される各グループのメンバース対して、境界線が表示されないデーブルを作成します。試合の結果は日付とトーナメントグループごとに異なるテーブルで表示するようにします。Group Tables のセグションでは、group 要素が動的なテーブルとして作成され、自動計算により必要とされるデータの値が計算されます。

SPS の作成

以下の操作によりデザインを作成します:

- 1. スキーマツリーから /results/match 要素をドラッグし、コンテンツとして作成します。
- 2. match 要素のタグを右クリックして、コンテキストメニューからグループ化を選択します。
- 3. 出力グループを定義ダイアログにて、group-by を選択し、マッチテキストボックスにて @date という XPath 条件式を入力し、「OK」をクリックします。
- 4. グループの並べ替えを行うかを尋ねるダイアログが表示されます。この例ではデフォルトのアルファベット順に よる並べ替えを行うので、はいをクリックします(並べ替え条件はいつでも追加または削除することができま
- す)。
 5. 各グループ(この例では date)を独立したテーブル内に作成するため、カレントグループをテーブルとして作成します。以下のスクリーンショットに示されるように、current-group() タグを石クリックして、「変更 | テーブル」を選 択し、子孫ノードの team/@name と team/@for をテーブルのカラムとして選択します。



- 6. それがカレントグループにおける最後の試合でない場合に出力される、ハイフン(-)を各セルに挿入します。条 件テンプレートを作成し、条件には position() != last() を使用します。これにより Brazil - Germany や 2 - 1 といっ た出力が行われます。
- 7. ヘッダーに自動計算を追加して、カレントグループのキーを表示するようにします(XPath 条件式には currentgrouping-key() を使用します)。
- 8. 必要に応じてテーブルのフォーマットを行います。 9. トーナメントグループにより試合結果のグループ化を行うには、上記の操作を match の group 属性に対して行い . ます。
- 10. 各チームの順位が表示されるデザインの最後にあるグループテーブルでは、/results/group 要素を動的テーブルとして作成します。「テーブル | カラムの追加」や「テーブル | カラムの挿入」コマンドを使うことで、必要に 応じてカラムを追加することができます。出力に必要な計算を行う自動計算を各カラムに作成します(白星によ り3ポイント、引き分けで1ポイント、黒星で0ポイント)。得られた合計得点数により降順でテーブルの並べ替 えを行います。これらの結果を得るために使用された XPath 条件式を確認するには、自動計算または並べ替 えられたテンプレートを右クリックして、それぞれ XPath の編集または並べ替えコマンドを選択します。

□ 関連項目

- 出力の構造
- 並べ替え
- サンプル Group-By (Persons.sps)

7.4 並べ替え

並べ替えの機能は HTML 出力にて使用することができます。

並べ替えは、同じ修飾名を持った兄弟要素ノードのセナーマガして、1つ以上のノートキーにより行うことができます。例えば、(Company 要素以下にある)全てのPerson 要素を、Person の子要素である LastName により述べ替えることができます。ソートキーはノードでなければならず、通常並べ替えが行われる要素ノードの子孫ノード(要素ませば属性)となります。ここで紹介した例では、LastName がソートキーとなります。

並べ替えの対象とよる2つの要素に含まれているノーキーの値が等しい場合、更にノーキーを追加することで、よい細かな並べ替えを行うことができます。上で示した Person 要素の例の場合、最初のノートキーとよる LastNmae し加え、2番目のノートキーとよる FirstName を指定することができます。LastName の値が同じ Person 要素が与えられた場合、FirstName による並べ替えが行われます。これにより、SPS 内で、同じ並べ替え処理に対して複数の並べ替え条件を指定することが可能にないます。

並べ替えのセナース対してテンプレートが適用されることで、並べ替えが行われた内容が出力に渡されます。並べ替えはHTML 出力にてサポートされます。

ユーザー定義テンプレート

ユーザー定義アンパレートを使用することで、指定したXPath 条件式により選択されるアイテムは大してテンプレートを適用することができます。XPath 条件式により選択されたノードを並べ替えることもでき、この場合、並べ替えはユーザー定義テンプレートに対して適用されます。

このセクション

- 並べ替えのメカニズムについて記述されます。
- サンプルにより、並べ替えかどのように使用されるか紹介します。

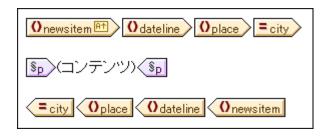
□ 関連項目

- 動的テーブルの作成
- XPath ダイアログ
- ユーザー定義テンプレート

7.4.1 並べ替えのメカニズム

要素のノードノマオして並べ替えを行うには、2つのステップを踏む必要があります:

1. デザインビューにて、並べ替えの対象となる要素ノードを選択します。XMLドキュメント内にあるこの要素のインスタンスが並べ替えられることになります。場合によってはどの要素が並べ替えの対象となるのか、すぐには分からないこともあります。例として以下のスクリーンショットに示される構造を考えてみましょう。



各 newsitem には、place 要素を含む dateline 要素が含まれており、place 要素には city 属性が含まれています。全 newsitem 要素以下にある @city ノードをアルファベット順に出力する場合を考えてみましょう。デザインでは、@city ノード(または place、dateline、newsitem 要素)を並べ替えの対象に選択するのでしょうか?@city が選択されると、1つの city ノードだけが並べ替えの対象となります。place または dateline が選択された場合でも、これらの要素は親ノード内に1度しか出現しないため、並べ替えは1つのノードに対してしか行われません。これに対して newsitem が選択された場合、親要素の newsitems 以下に複数のnewsitem 要素が出現するため、newsitem に対して並べ替えを行うことになります。

2. 並べ替えを行う要素を選択した後には、その要素を右クリックすることで表示されるコンテキストメニューから並べ替えコマンドを選択します。出力ソートの順序を定義ダイアログが表示され、並べ替えの内容を入力することができます(以下のスクリーンショットを参照)。



各並べ替えの設定内容には、(i) ソートキー(マッチカラムにて入力)、(ii) 並べ替えに使用されるソートキーのデータ型(テキストまたは数値)、そして(iii) 並べ替えの順序(昇順または降順)が用意されています。このダイアログに表示される設定内容の順序は重要な意味を持ちます。複数のアイテムが同じ値を持っている場合、並べ替えは最初の行にある条件から最後の行まで順に行われます。並べ替えに使用する条件の数に制限はありません。

並へ替えが使用される例これでは、次を参照してください、サンプル、複数のノートキーによる並へ替え

ユーザー定義テンプレート

ユ<u>ザ・定義デンプレート</u>を使用することで、指定したXPath 条件式により選択されるアイテムニ対してテンプレートを適用することができます。XPath 条件式により選択されたノードを並べ替えることもでき、この場合、並べ替えはユーザー定義デンプレートに対して適用されます。

ソートキーは関するメモ

入力されたXPath 条件式にお選択されるノートキーは単一のノードでなければおらず、ノードセナ(XPath 1.0)やシーケンスアイテム(XPath 2.0 とXPath 3.0) とすることはできません。各要素のキーは、文字列まけは数値の値に対して解決可能である必要が決ます。

XSLT 2.0 または3.0 のSPS にてノードのシーケンスがソートキーに対して得られた場合、エラーがXSLT プロセッサーによ返されます。 上記のPerson サンプルの場合、../Person/LastName とう XPath 条件式では、Person の親要素に含まれている全てのLastName 要素が返されるため、エラーが返されます(複数のPerson 要素が存在する場合)。Person をエンテキストノードとする正しいXPath 条件式は、LastName といます(各 Person 要素には1つのLastName ノードしか存在しないため)。

XSLT 1.0 では、ソーキーのセンターにより一ドセナが得られた場合、最初のノードにあるテキスト値が使用されます。ソートキーに対してXPath 条件式により複数のノードが選択された場合でも、Style Vision によりエラーが返されることはなく、最初のノードにあるテキストが使用されます。しかし、最初に選択されたノードが目的のノートキーではむ、場合もあります。例えば、上に挙げた…/Person/LastName XPath 条件式によりエラーが返されることはありませんが、各要素に対して同じ値(最初のLastName ノードにあるテキスト値)が使用されるため、並べ替えが行われることもありません。この場合、location/@* という形式の条件式を使用することにより、location 子要素にある最初の属性がノートキーとして使用され、並べ替えが行われることではますが、このような条件式の使用は推奨されておらず、より正確な(単一のノードを選択する)条件式の記述が推奨されます。

三 関連項目

- サンプル 複数のノートキーによる並べ替え
- ユーザー定義デンプレート
- 動的テーブルの作成
- XPath ダイアログ

7.4.2 サンプル:複数のソートキーによる並べ替え

以下にあるシンプルな例では、チームメンバーかテーブルに表示されます(サンプルファイルは(マイ)ドキュメントフォルダーのc:

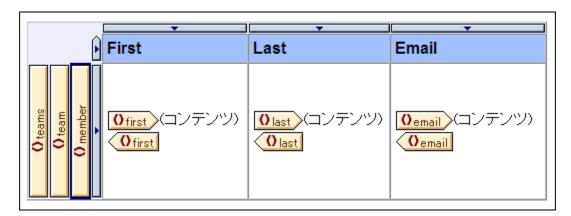
\Documents and Settings\<username>\My

Documents Altova Style Vision 2021 Style Vision Examples Tutorial Sorting Sorting On Two Text Key s.sps 「て確認することができます)。テーブルの行口は各メンバーの氏名と電子メールアドレスが表示されます。ここでは表示されているメンバーの苗字でまず並べ替えを行い、次に名前による並べ替えを行います。

表示されている名前が並べ替えられていない場合、XMLドキュメント内のmember要素と同じ順序で名前の表示が行われます(以下にあるHTML出力のスクレーンショナを参照)。

First	Last	Email
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com
Janet	Ashe	j. ashe@nanonull.com

デザインビュー・コンプ member 要素(以下のスクノーンショナーにて青色で イライトされているノード)を右クルグし、コンテキストメニューから並べ替えコマンドを選択します。





デザインビューでは、並べ替えのフィルターを表すアイコンがmember タグに表示されます member 配う。チームメンバーのHTML 出力では、以下に示されるように苗字が並べ替えられた状態で表示が行われます。ここで Edwards の並べ替えが正しく行われていないことに注目してください (Nadia が John の前に表示されており、これは XML ドキュメトにおける要素の順序に従ってものです)。名前による並べ替えを行うされて、2番目のノートキーを作成します。

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com
John	Edwards	j.edwards@nanonull.com

デザインビューにある member タグを右クトックして、コンテキストメニューから並べ替えコマンドを選択します。last による並べ替えを含む出カノードの順序を定義ダイアログが表示されます。新たな並べ替えのルールを追加するコよ、新たな行を追加して、ソートキーにfirst 要素を入力します(以下のスクノーンショナを参照)。「OK」をクトックして完了します。



HTML 出力では、最初に苗字が、次に名前により述べ替えられた結果が表示されます。

First	Last	Email
Janet	Ashe	j.ashe@nanonull.com
Andrew	Bentinck	a.bentinck@nanonull.com
John	Edwards	j.edwards@nanonull.com
Nadia	Edwards	n.edwards@nanonull.com

□ 関連項目

- 動的テーブルの作成
- XPath ダイアログ
- ユーザー定義アンプレート

高度な機能 パラメーターと変数 251

7.5 パラメーターと変数

パラメーターと変数はSPS 内部にて宣言ならびて参照することができます。変数は宣言時に値が定義されるのに対して、パラメーターは、ランタイムにてコマンドラインから)渡された値により宣言時に割り振られたオプションのデフォルト値をオーバーライドすることができます。

このセクションでは、パラメーターと変数に対して使用することができる機能について説明します。

- ユーザー宣言/ ラメーター: ユーザー定義の/ ラメーターを SPS にて使用する方法について説明します。
- <u>デザインフラグメントしつけする ウメーター</u> デザインフラグメント とともにつ ウメーターを使用する方法について記述します。
- <u>ソースに対する SPS パラメーター</u>: スキーマノース(特に作業用 XML ファイルのスキーマ)に対して Style Vision が自動的に 定義する特殊な種類の プメーターこれ で説明します。 これら プメーターの名前と直はユーザーにとって既知のものであるため、 SPS 内部で プメーターの参照を行い、 ランタイムにてコマンドラインから値を渡すことができます。
- 変数: (i) 特定のスコープ つ対して変数を宣言し、その値を定義し、(ii) 宣言された変数の値を参照し、変数により選択されたノードにてテンプレートを作成することができます。

- 関連項目

- SPS ファイルゴンテンツ
- テンプレートとデザインフラグメント

7.5.1 ユーザー宣言パラメーター

SPS では、ユーザーにより宣言されたいでメーターがデフォル・の文字列値とともにグロー・VUに使用されます。一旦パウメーターが宣言されると、SPS 内の任意の場所から XPath 条件式により使用することができます。パウメーターのデフォルト値は、StyleVision Server から XSLT スタイルシートに対して新たなグロー・VI値を渡すことで、個々の XSLT 変換に対してオー・バーライドすることができます。

パラメータの使用

ユーザー定義されたプラメーターは、以下の様な状況で使用することができます:

- 特定の値を複数の場所で使用し、複数の計算に対する入力値として使用するこの場合、目的の値を、ラメーター値として保存して、必要とされている場所で計算にて、ラメーターを使用することができます。
- スタイルシートを処理する際に値を渡す: SPS(そしてスタイルシート)では、パラメーターがデフォルトの値とともに使用されます。処理が行われる際に、StyleVision Server から必要な値をパラメーターに対して渡すことができます。

メカニズムの使用方法

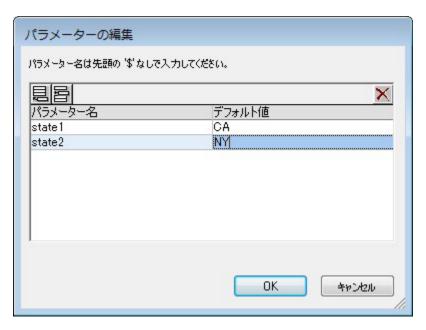
ユーザーにより宣言されたプラメーターを SPS にて使用するココ以下にある2つのステップを踏む必要があります:

- 1. 必要ない ウメーターを宣言する。
- 2. 宣言されたプメーターを参照する。

パラメータの宣言

ユーザーによる・ウメーターの宣言と編集は、ウメーターの編集ダイアログにて行うことができます(以下のスクリーシショナを参照)。ノウメーターの編集ダイアログへは、メニューオプションの「編集 | スタイルシートのノウメーター リコマンドからアクセスすることができます。

高度な機能 パラメーターと変数 252



ノ ラメーターの名前とデフォルトの値により、ラメーターの宣言を行います。デフォルトの値を与えない場合、デフォルト値を空にします。

以下の操作により、ウメーターを宣言することができます:

- 1. パラメーターの編集ダイアログにある追加または挿入ボタンをクリックすることで、新たなパラメーターを追加ま
- たは挿入します。新たな行が表示されます。 2. パラメーターの名前を入力します。パラメーター名の最初の文字にはアルファベットを、それ以外では英数字と アンダースコアを使用することができます。
- 3. パラメーターのデフォルト値を入力します。入力された文字はテキスト文字列として扱われます。

新規パラメーターの追加や、既存パラメーターの編集は、SPS の編集時にいっても行うことができます。

УŦ

- パラメーターの編集ダイアログには、SPSにある全てのユーザー定義パラメーターが表示されます。
- パラメーターはデザル概要サイバーでも宣言することができます。

宣言されたプラメーターの参照

XPath 条件式では、パラメーターの前に \$ というプレフィックスを配置することで、パラメーターへの参照を行うこと ができます。例えば、自動計算の XPath 条件式にてパラメーターを参照することができます(例: concat('www.', \$company, '.com')) o

宣言されていない、ウメーターを参照するとエラーが返されますが、参照されない、ウメーターを宣言してもエラーとはいません。

デザインフラグメントに対するパラメーター 7.5.2

デザインフラグメントのプラメーターにより、作成したデザインフラグメントに対してパラメーターを定義し、このパラメーター(ネオしてデフォルト値を 与えることができます。デザインフラグメントが使用されるデザインの各箇所では異なる、テメーター値を入力することも可能で、各デザインフラグ メントの出力を個別に修正することもできます。

例えばEMailAddresses とり名前のデザインフラグメートをaltova.com とりデンオル値を持つDomain とり名前のプメーターとともこ 作成する例を考えてみましょう。このプラメーターを自動計算にて使用することで、従業員の電子メールアドレスを生成することができます。EU

のアドレスコ対してはEmailAddresses デザインフラグメトーする Domain パラメーターの値にaltova.eu を使用します。 同様の方法で、日本のアドレスコ対しては、 altova.jp を Domain パラメーターの値とて使用します。 米国内の従業員が持っているアドレスコ対しては Domainの値を変更すること無く、デフォルト値の altova.com を使用します。

デザインフラグメントしずしてパラメーター値を使用するコよ、以下にある2つのステップを踏むことづまります。

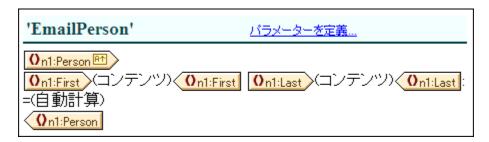
- 1. デザインフラグメントが作成される場所にてデフォルトの値とともに<u>パラメーターを定義</u>します。
- 2. デザインフラグメントが使用される場所でパラメーターの値を編集します。

これらのステップについては以下で詳細に説明します。

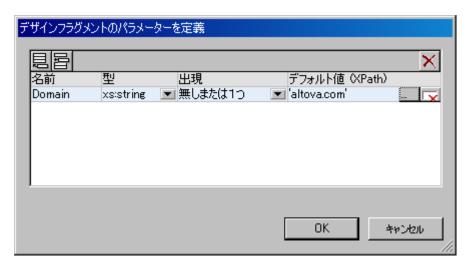
メモ デザインフラグメントに対するパラメーター機能は Enterprise Edision の Altova 製品における Authentic View でのみサポートされます。

パラメーターの定義

各デザインフラグメントに対して割り当てる。ウメーターの数に制限はありません。デザインフラグメントのタイトル・コープある・ウメーターを定義リンクをクトックすることで、ウメーターを定義できます(以下のスグリーンショントを参照)。



この操作により、デザインフラグメトの ヴメーターを定義ダイアログが表示されます(以下のスクナーショナを参照)。 左上にある追加ならびに挿入アイエンをクトックすることで、パヴメーターを入力する行が追加されます。パヴメーターの名前、データ型、出現の回数、そしてデフォルト値を入力まけは選択します。 パヴメーターの出現属性により、パヴメーターのデフォルト値として指定された XPath 条件式を評価することで得られるアイテムの数を指定することができます。 出現属性はオプションとなっており、デフォルトで無しまけは1つ」にセットされています。 パヴメーターの数に制限はおりません。

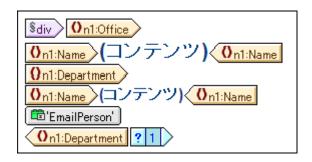


ダイアログコ は2種類の削除アイコンかります。各パラメーターエントリーの右側にある削除アイコンをクリックすると、パラメーターのデフォルト値が削除されます。 ペイン右上にある削除アイコンをクリックすることで、現在ハイライトされているパラメーターが削除されます。

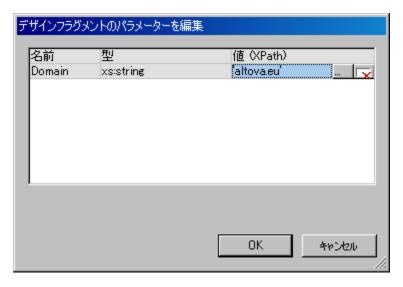
メモ SPS にて XSLT 1.0 が選択されている場合、入力した XPath 条件式によりノードセットが返されなければなりません(そうでない場合エラーが返されます)。

パラメータを使用する

デザインフラグメトを作成すると、デザイン内にある複数の箇所へ、デザインソー・やスキーマソー・からドラッグすることにはり、挿入することができます。デザインフラグメトのEmailPerson がn1:Name 要素の後に挿入された状態を以下のスクノーンショナに示します。



デザインフラグメトにてい ラメーターが定義すると、デザインフラグメトのインスタンスにてい ラメーターの値を編集することができます。 デザインフラグメント のインスタンスを右クトックして、パラメーターを編集コマンドを選択します。 デザインフラグメントの ・ラメーターを編集ダイアログが表示されます(以下のスクリーシショントを参照)。



このダイアログではか ウメーターの値を編集することができます。「OK」をクトックすることで操作を完了します。入力されたか ウメーター値がデザインフラグメントのインスタンス こて使用されます。 / ウメーターの値が編集されていない場合、オルジナルの(デフォルトの) / ウメーター値が使用されます。

メモ XSLT 1.0 が選択されている場合、入力した XPath 条件式によりノードセットが返されなければなりません(そうでない場合エラーが返されます)。

□ 関連項目

• ユーザ 宣言パラメーター・ドキュメント全体に対して妥当なスタイルシートパラメーターの使用方法について記述 されます。.

7.5.3 ソースのための SPS パラメータ

SPS にはXML ドキュメントがベースとするDTD やXML スキーマ、まだはDB から生成された XML スキーマやDB がベースとしているスキーマを、スキーマノースとして複数個含めることができます。

SPS には、メインとなるスキーマを1つ、そしてオプションのスキーマを複数個含めることができます。新たなスキーマノースが加えられると、そのスキーマニ対する。 ウメーターがStyleVision により自動的に宣言され、そのスキーマニ対して割り当てられた作業用 XML ファイルのURIが、プラメーターの値として自動的に割り当てられます。 DB の場合、StyleVisionにより一時的な XML ファイルが DB から生成され、この XML ファイルのドキュメントノードが ウメーターにセットされます。

ソースコ対するパラメーターを参照する

各 SPS パラメーターは、そのスキーマース対応する XML ファイルのドキュメントノードを値に持ちます。 StyleVision では、各スキーマース対応 した XML ファイルは、作業用 XML ファイルは対けま DB から生成された XML ファイルとなります。従って、ソースに対する SPS パラメータ ーを以下のように利用することができます:

- 1. SPS 内にある XPath 条件式にて様々なドキュメント内のノード位置を特定する: ドキュメントの位置特定にパラメーターが使用され、ドキュメント内にある目的のノードの位置を特定するために、XPath 条件式が使用されます。例えば、count(\$XML2//Department/Employee) という条件式により、\$XML2 として指定されたスキーマソースの作業用 XML ファイル内にある Department 要素に含まれている Employee 要素の数が返されます。
- 2. コマンドラインから、他の XML ファイルの URI を SPS のパラメーターとして渡す: 勿論、新たな XML ファイルもパラメーターにより指定されたスキーマをベースにしたものでなければなりません。例えば、FileA.xml と FileB.xml が、SPS にて使用されているスキーマの \$XML3 に対して妥当であり、FileA.xml が SPS にて使用される \$XML3 スキーマに対して割り当てられた作業用 XML ファイルである場合、コマンドラインからその SPS に対して XSLT 変換を指定する際に、\$XML3="FileB.xml" というパラメーターを使用することで FileA.xml の代わりに FileB.xml を使用することができます。コマンドラインでは、メインスキーマを除く全てのソースに対して全ての SPS パラメーターを入力する必要がある点に注意してください。メインスキーマに対応する XML ファイルは XSLT スタイルシートのエントリーポイントであり、変換が行われる XML ファイルとなります。

□ 関連項目

ユーザー宣言パラメーター

7.5.4 変数

変数は (i) 変数の宣言 と(ii) 変数の使用にお構成されます。

メモ 変数は、Enterprise Edition の Altova 製品にある Authentic View でしかサポートされない点に注意してください。.

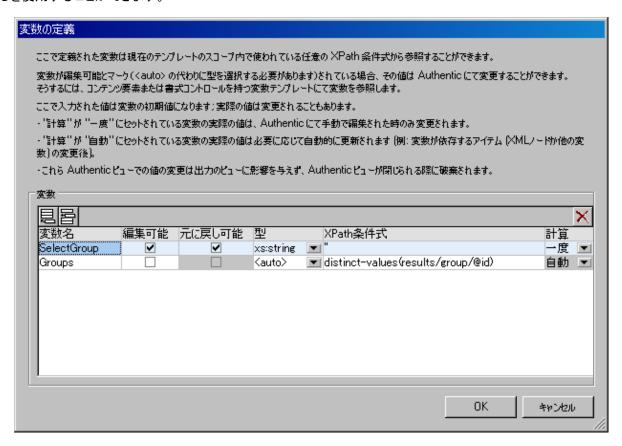
変数の宣言

変数は、デザイン内に含まれているあらかるテンプレートにて宣言することができます。変数には名前、データ型、そして値が与えられます。更に、Enterprise エディションのAuthentic Viewにて編集可能がを指定することができます。変数はこのテンプレートをスコープに作成され、その内部で使用することができるようによります。ドキュメト全体をスコープとする変数を宣言するには、ルートテンプレートに対して変数を宣言します。テンプレートにて変数を宣言することで、子孫ノードを特定するためのXPath 条件式をシンプルに作成することができます。

以下の操作により変数を宣言することができます:

1. 変数を作成するノードテンプレートを右クリックして、コンテキストメニューから変数を定義を選択します。

2. 変数の定義ダイアログにて、変数ペインの左上にある変数の追加アイコンをクリックして、変数の名前を入力します(以下のスクリーンショットを参照)。変数の値は XPath 条件式を介して入力することになります。変数の値に(下のスクリーンショットにある最初の変数のように)文字列を使用したい場合、引用句により文字列を囲んでください。以下のスクリーンショットでは、SelectGroup 変数の値が空の文字列となり、他にもノード名や関数呼び出しを使用することができます。



- 3. 編集可能チェックボックスにチェックを入れて変数を編集可能にセットすることで、Authentic View 内で変数を編集することができます(StyleVision のEnterprise とProfessional エデションではプレビューでのみでこの機能を使用することができます)。この場合、データ型の値をxs:string のように正しくセットする必要があります。変数が編集可能となっている場合、SPS デザイナーによりセットされたオリジナルの値を Authentic View ユーザーが Authentic View にて変更することができます。変更は(変数の値を編集可能なコンテンツや Authentic View により編集可能なテキストボックスなどにより)変数を直接編集することでも行えますし、変数の XPath 条件式で使用されているノードや値を Authentic View ユーザーが修正することでも行うことができます。
- 4. 変数が編集可能となっている場合、Authentic View に関連して2つのオプション(元に戻し可能ならびに計算)が更に有効になります。元に戻しオプションのチェックボックスにチェックを入れることで、変数に対して行われた全ての変更を元に戻すことが可能になります。このオプションにより、Authentic View ユーザーは変数の値を変更前のものに戻すことができるようになります。計算オプションの値は一度または自動で、オプションの値が一度にセットされた場合、変数を含むテンプレートが評価される時に、変数の値が一度だけ計算され、ユーザーにより変数が明示的に変更された時だけ値の変更が行われます。オプションの値が自動になっている場合、変数の XPath 条件式で使用されているノードや値が修正された時にも変数の計算が行われます。
- 5. 追加することのできる変数の数に制限はありませんが、同じスコープ内で既に宣言されている変数の名前を使用することはできません。変数を削除するには、ペインの右上にある削除アイコンをクリックしてください。
- 用することはできません。変数を削除するには、ペインの右上にある削除アイコンをクリックしてください。 6. 「OK」をクリックすることで完了します。テンプレートタグに \$ アイコンが表示され、そのテンプレートに対して変数が宣言されたことを示します。

上の操作では、デザイン内にある各ノードテンプレートに対して変数が作成されます。これら各変数には名前と値が与えられ、宣言されたテンプレート内部がそのスコープとなります。変数を修正するには、変数が作成されたノードテンプレートを右クトックして、コンテキストメニューから変数を定義コマンドを選択し、変数の定義ダイアログへアクセスします。

変数の使用

変数は、その変数が作成されたスコープ内で使用することができます。つまり、変数が作成されたテンプレート内部でしかその変数を使用することはできません。変数はAuthentic View でのみ編集することができるため、表示内容はユーザーによりエトロールされます。編集された値はSPS が閉じられた時に破棄されます。

変数は任意のXPath 条件式で使用することができ、\$ シンボルともにその名前を使用することで、XPath 条件式内部から参照することができます。例えば、\$VarName 人らXPath 条件式を使うと、VarName とら名前の変数により選択されるノードのName 子要素が選択されます。



XPath 式の編集ダイアログ にてXPath 条件式を入力すると、スコープ内にある変数がポップアップにて表示されます(上のスケノーンショットを参照)。ポップアップから変数を選択し、Enter キーを押下することで、その変数への参照が条件式内に追加されます。

□ 関連項目

- SPS ファイルコンテンツ
- テンプレートとデザインフラグメント
- <u>ノードテンプレートオペレーション</u>

7.6 目次、参照、ブックマーク

目次(TOC)やその他の参照メカニズムは、デザインドキュメント内にある目的の場所にてアンカーを作成し、目次やテキスト参照、自動番号シーケンス、ませるソイパールンクから、これらのアンカーニ対して参照を行うというかどっちで作成されます。

まずはアンカー(ませばジソフマーク)のメカニズムこと、で簡単に説明し、目次メカニズム全般について説明します。ブソフマークの動作方法を理解することにより、目次メカニズムの理解をより良く行うことができるようしています。

ブックマークのメカニズム

ブックマーケノコよ 単純ブックマーケと複合ブックマーケという2種類のメカニズムが存在します。目次の作成には複合ブックマーケが使用されます。

- 単純ブックマークはデザインドキュメント上に作成されます。ブックマークにはユニークな名前が与えられ、 参照を行うリンクのターゲットとして使用されます。単純ブックマークメカニズムは、ブックマークとハイパーレク に対して使用される機能となります。ハイパーレクはギュメント外部にある URL への参照を行うこともができるとうことに留意してくたさい。
- 目次やドキュメントセクションに対して番号を与えるといった、より複雑な参照を行うには、以下にある2つのステップによりブックマークを作成します:
- 目次に適合するかたちでデザインドキュメントを階層構造にデザインします。階層構造における各レベルは目次レベルと呼ばれます。ドキュメント構造内部にある各ポイントへ目次レベルを割り当てることにより、構造を作成することができます。目次レベルは他の目次レベル内部にネストすることができ、ドキュメントにて階層的な目次構造を作成することができます。例えば、ある目次レベルを本の章に割り当て、別の目次レベルを章の中にある節に対して割り当てることができます。
 目次レベル内部に目次ブックマークを作成することができます。これら目次ブックマークにより、目次内部へ渡さ
- 2. 目次レベル内部に目次ブックマークを作成することができます。これら目次ブックマークにより、目次内部へ渡される各レベルにおけるドキュメントセクションが識別されます。更に、各目次ブックマークには参照元のコンポーネントに表示されるテキストを指定する必要もあります。

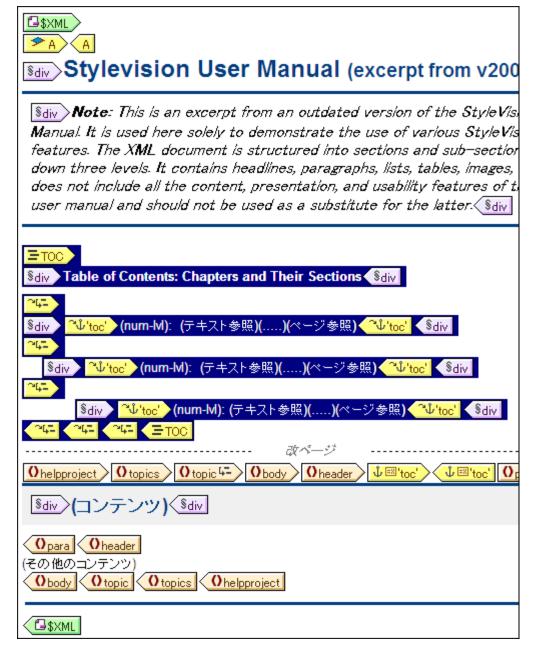
目次レベルならびに目次ブックマークの参照テキストが定義されると、参照元のエンポーネントを含む目次テンプレートをデザインすることができます。

以下にある<u>目次メカニズム</u>では、目次のメカニズム全般について簡単に説明されます。このセグションでは、利用することのできる参照機能の詳細についても記述されます。

目次メカニズム

SPS で使用される XSLT の デジョンに(XSLT 1.0 ではなく) XSLT 2.0 または XSLT 3.0 を選択した場合、目次(TOC) 一正確には目次のテンプレート - をデザインにおける任意の場所で作成することができます。

- 目次内部からリンクされるデザイン上のアイテムは、最初に デザイン内部でブックマークしておくこめ推奨されます。 これらのアイテム 講的コンテンツまけば助的コンテンツといます。以下にあるスクノーンショナの下半分けは、黄色の目次ブックマークタグ目次ブックマークags ↓ 図 MyTOC が header タグ内部に表示されており、header アイテムが(目次テンプレートに含めるために)ブックマークされています。
- <u>目次のためにアンプレートを作成</u>します(以下のスクノーンショナにて)イライトされている箇所を参照)。 目次アンプレートには目次のデザインが含まれており、 テンプレートはデザインにおける任意の場所に配置することができます。 以下のスクノーンショナにて示される例では、ドキュメナの上部近くに目次アンプレートが配置されています。



メモ これら2つのパーツを作成する順序に決まりはなく、同時に作成を行うこともできます。しかし、目次テンプレートを作成する前に目次ブックマークを作成することが推奨されます。

目次は HTML 出力にて表示されます。他にも、(i)目次は階層構造ませまプラナ構造で作成することができ、(ii)デザイン内部に複数の目次を作成することができます。これにより、階層構造によるブッケならびに章レベルの目次に加え、図表のレストをプラナな構造で作成することができるようになります。

目次の作成ステップ

目次の作成方法を、以下のステップに示します。最初に目次アイテムを追加して、その後に目次テンプレートを作成します。その他にも目次 テンプレートをまず作成して、追加するブックマーケアイテムを作成することができます。また、目次テンプレートとアイテムの作成を並行して行うこともできます。

- 1. XSLT のバージョンに XSLT 2.0 が選択されていることを確認してくたさい。
- 2. <u>目次レベル内部のドキュメントの構造</u>: 目次に複数のレベルが含まれる場合、ネストされた目次レベルの構造をデザインします。目次がフラットな構造になる場合(つまり目次レベルが1つだけの場合)、目次ブックマークを含めることになる目次レベルを少なくとも1つ作成します。
- 3. ドキュメントデザインにおける各レベルにて1つ以上の目次ブックマークを作成: 目次内部に表示される各目次レベル内にあるコンポーネントが、目次ブックマークにより識別されます。
- 4. <u>目次レベルの参照を含んで、る目次テンプレートを作成</u>: 目次テンプレートには、必要な数だけの目次レベル参照を含める必要があります。目次が複数レベルにより構成される場合、目次テンプレート内部にあるレベル参照をネスト化する必要があります。
- 5. <u>目次テンプレート内部に目次参照(TOCref)を作成</u>: 目次テンプレートにて、各レベル参照に対して目次参照をセットします。各目次参照から、ドキュメントの対応する目次レベル内にある目次ブックマークへの参照が行われます。目次参照からは、その他のレベルにある目次ブックマークを参照することもできます。
- 6. <u>目次アイテムのフォーマット</u>: 目次出力における各テキストアイテムは、目次テンプレートの目次参照により生成されます。目次参照の定義により、(階層構造を含む)アイテムの番号付け、目次アイテムのテキスト、そして印刷媒体に対してはページ番号を指定することができます。ドキュメントのメインボディーにある目次ブックマークでは、自動的に割り当てられる番号を定義することもできます。詳細については<u>番号の自動割り</u>当てのセグシュを参照ください。

用語解説

260

インターフェースで使用される目次に関連した主要なコンポーネトの名前を以下のテーブルに示します。テーブルのカラムこで示されるようにドキュメントボディーと(通常ドキュメント上部に実際に表示される目次のデザインである)目次テンプレートは個別に理解する必要が必ます。

- ドキュメントボディー内部の目次コンポーネント: 目次テンプレートにて使用されるアイテムをマークするために使用されます。
- 目次テンプレート内部の目次コンポーネント: ドキュメントボディー内部にてマークされたアイテムを参照するために使用されます。

ドキュメント body	目次 テンプレート
目次レベル: ネスト形式で、ドキュメントの構造を作成します。	レベル参照 (levelrefs): ドキュメントボディーにで定義された目次レベルの構造に対応します。対応するレベルである目次ブックマークへの目次参照を行うことができます。
目次ブックマーク: 名前を与えることにより、目次アイテムとして使用されるドキュメント内のノードを識別します。	目次参照 (TOCrefs): 名前により目次ブッケマークへの参照を行います。

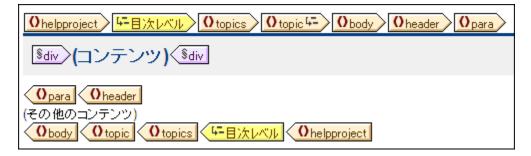
■ 関連項目

- 目次に含まれるブックマーケアイテム
- 目次テンプレートの作成
- ブックマークとハイパーレク

7.6.1 目次に含まれるブックマークアイテム

デザイン内にあるブックマークを目次に含めるには、2つのステップを踏む必要があります(ステップの順序に意味はありません)。

1. <u>デザインド・キュメートを目次レベルにてネストされた構造になるように構成する</u>: デザインにおける目次レベルは、テンプレート上に ませるデザインエンポーネントの周りに作成することができます。以下のスクリーシンコントでは、topic テンプレート () topic いこて目次レベルが作成されています。

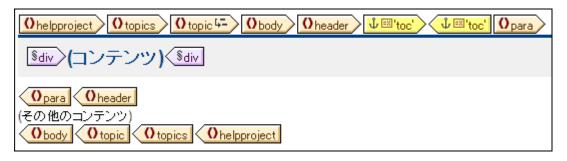


テンプレートにてレベルが作成されると、テンプレートの開始タグにその旨が表示されます(例:

○ topic 4->)。コンポーネントの周りにレベルが作成された場合、目次レベルタグ

←目次レベル
により目次レベルが表示されます。上のスクリーンショットでは、topics テン プレートコンポーネントがレベルにより囲まれることになります。これら2つの方法の違いは、<u>レベル内部のデ</u> ザル構造 セクションにて説明されます。目次テンプレートの作成された場合、デザインにて作成されたレベル いても、デザインには対応するレベルを含める必要があります。

2. 目次ブックマークの作成:名前ならびに目次アイテムテキストを伴うデザインにて目次ブックマークを作成します。目次ブックマークは デザインエンポーネントを囲むかがっちでデザインすることもでき、エンポーネントが囲まれない場合、空の要素として表示されます。以 下のスクノーンショナでは、デザインコンポーネントを囲んでいない。日次ブックマークか示されています。

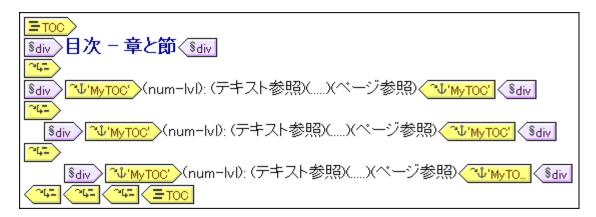


目次ブックマークはドキュメント内のアンカーとして動作します。上のスクリーンショットでは目次ブックマー ク(アンカー)が para 要素インスタンスの開始タグに配置されています。目次ブックマークには、(i) 目次テン プレート内にある目次アイテムを作成する際に目次ブックマークを参照するための名前と、(ii)対応する目次アイテムのテキストとして使用されるテキスト文字列という、2つの属性が含まれています。これら2つの 属性の割り当て方法については、目次ブックマークの作成セクションを参照ください。

目次テンプレートにてブックマークされたアイテムを参照する

(デザインテンプレートのメインボディー:作成されケユベリとは違う)レベル参照という名前のネストされケユベル内にTOC テンプレートは作成 されます。各レベル参照 Carlot 内部コよ 目次参照 Carlot か挿入されます(以下のスグノーンショナを参照)。レベル参照内 にある目次参照からよ(目次ブックマークの名前による)目次ブックマークへの参照が行われます。XML ドキュメント内部の対応するレベル しある指定された名前を持った目次ブックマークが、目次内のアイテムとして作成されます。例えば「い chapters' とうタグにい展示さ れる目次参照により、XML ドキュメント内の対応するレベルにある chapters という名前の全目次ブックマークが参照されること つなります(目 次参照のアコープがcurrent にセナされていた場合)。インスタンス化された目次ブッケマークのテキスト属性が目次アイテムのテキストとして 使用されます。

 262
 高度な機能
 目次、参照、ブックマーク



上のスクーンショナに示される目次テンプレートでは、3つのレベル参照がネトされており、各レベル内にある目次参照にはそのレベルの目次アイテムで対するテンプレートが含まれて、ます。例えば、最初のレベル参照には MyTOC と う名前を持った全ての目次ブックマークを参照する目次参照が与えられて、ます。これにより、最初のレベル内にある MyTOC と う名前を持った全ての目次ブックマークが、出力に表示される目次からアクセスできるようにはます。同様に、ドキュメートの2番目のレベルにある MyTOC と う名前の目次ブックマークが、目次の2番目のレベルから参照されるようにないます。ドキュメント3番目のレベルにある MyTOC と う目次ブックマークが、目次内の3番目のアイテムにより使用されます。

このセグション以下にあるサブセグションでは(i) デザイン内部でレベルを構築する方法 について、そして(ii) ブックマークの作成方法について説明します。 目次テンプレートの作成 は、目次テンプレートの作成のセグションで説明されています。

- 関連項目

- 目次(TOC)
- レベル内部のデザイン構造
- 目次ブックマークの作成
- 目次テンプレートの作成

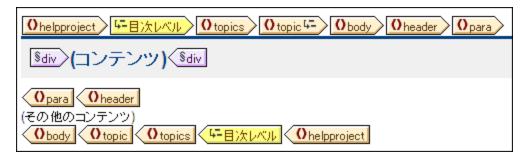
7.6.1.1 目次レベル内部のデザイン構造

目次をデザインするかの階層構造は、ネストされたレベルのセナとして指定されます。このような階層構造は、XMLドキュメトの構造は関連しますが、それとは異なるものです。この構造はSPSドキュメトデザイン内部で指定されます。作成する目次テンプレートはこの階層構造に対応するものどなります。フラナな構造(レベリムがつだけの)目次の場合、デザインドキュメトに少なくとも1つのレベリを含める必要があります。ドキュメト内に2つ以上のレベリは存在する場合、これらレベリのどれか、ませは(1つのレベリと、在後数のレベリを統合することにより)複数のレベルに対してフラナな目次を作成することができます。

デザインでは、メインテンプレート、グロー・バリテンプレート、またはこれらか合わさった内にしくいか作成されます。レベルがどコに作成されるかに関わらず、これらのレベルにより明確な階層構造を構築する必要があります。

レベルの作成

デザイン内部にある各レベルは個別に作成されます。レベルはテンプレートまけません。本大の周りに作成されます。以下にあるスグレーンショナでは、topic テンプレート上に(otopic により示される)テンプレートが作成され、topics 要素の周りにも別のレベルが作成されて、ます(utility otopic により、これらと種類のレベルを作成する際の大きな違いは、レベル内に囲むオプション utility otopic otopic いま、アンプレート以外のコンポーネントに対してもレベルを作成することができるという点です。

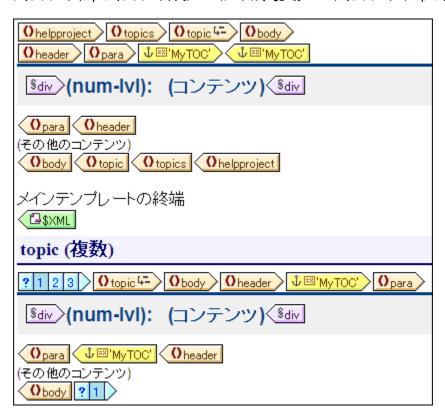


以下の操作によりしている作成することができます:

- コンポーネント(テンプレートまたはその他のコンポーネント)を選択します。
 選択対象を右クリックして、コンテキストメニューから「レベルとして機能するテンプレート」(テンプレート作成 時に有効になります)または「タグで囲む | 目次レベル」を選択します。これらのオプションは、メニューオプションの「挿入 | 目次を挿入」以下にある「目次レベル」または「レベルとして機能するテンプレート」からも選択 することができます。

グローバルテンプレート内部のレベル

グロー、ゾリテンプレート内部にもレベリをセナすることができます。この場合、グロー、ゾリテンプレート内部に作成されオシベリとメインテンプレー ト内に作成されオンベルの両方により、SPS 実行時に階層構造が定義されるようにする必要があります。以下のスクレージショナでは、メイ ンテンプレート(topic テンプレート)内に1つのレベルが、そしてグロー・ゾリテンプレートのtopic 内部に別のレベリが定義されています。



上のスケリーシショナに示されるコンテンソモデルでは、topic が再帰的な要素となっており、topic 要素内部に、別のtopic 要素を含めること ができます。(ダメメル タグによ終了地点が示される)メインテンプレートでは、最初のレベリのtopic にてレベリがセナされています(Otopic └──)。メインテンプレート「あるその他のコンテンソにより、header を除く topic/body 以下にある全子要素に対してテンプレートが 適用されることが指定されます。つまり、topic/body のtopic 子ノードに対するグロー・シリテンプレートも処理されることしています。

264 高度な機能 目次、参照、ブックマーク

topic 「対するグロー いテンプレートでは topic テンプレートに対して(Otopic 「こというされる) レベルがセナされています。このレベルは2番目のレベルのtopic 要素にて出現し、目次では、最初のレベル内部にオストされる2番目のレベルに対応します。このグロー いテンプレートにはその他のコンテンル合まれており、topic に対するグロー いテンプレートも再帰的な topic 要素に対して適用されること プリストロストされたとかけしべい(3番目のレベル、4番目のレベル、など) か新たって作成されること プリます。

デザインを行う際には、デザイン内に作成されたシベルの数に留意する必要があります。目次テンプレートを作成する際には、各レベルにある 目次アイテムを明示的に選択し、フォーマルすることしています。

フラナ構造の目次におけるレベル

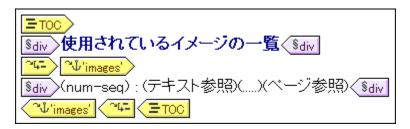
プラナな目次階層構造では、目次アイテムが単一のノベルコで出力されます(使用例: ドキュメト内に表示されるイメージのリスト)。

以下に示される方法により、フラナな構造を取得することができます。

- デザイベキュメトに含める目次レベルを1つだけにする: 目次テンプレートに含まれるレベル参照(そして、その中に含まれる目次参照(TOCref))も、1つだけないます。
- デザイドキュメト内の目次レベル複数存在する場合:目次テンプレート内に、参照される目次レベルの位置に対応したしべ ル参照を配置することではます。ターゲナとなる目次レベルに対応するレベル参照にて、目次参照を1つ配置することではます。
- デザイ・ドキュメートに複数の目次レベル特でである。目次テンプレート内に配置された目次参照には、ターゲートとなるドキュメートレベル全体をカバーするスコープを指定する必要があます。

ドキュメト内にある全てのイメージのリストを、フラナな構造の目次として表示する例を考えてみましょう。ドキュメトデザインコまりなくとも 1つのレベルを含める必要があり、このレベルのなかに必要な全ての目次ブックマーケを含めることしています。目次テンプレートでは、リストに表示されるイメージが通常と同様の方法で参照されます: (i) 対応する数のレベル参照を作成し、(ii) ターゲナとなる目次レベルコ対応するレベル参照にて目次参照を作成します。目次参照には、ターゲナとなる目次レベルコおける目次ブックマーケの名前が含まれます。

以下のスクノーンショナにある目次テンプレートでは、images とり名前の目次ブックマーケを参照する目次参照か含まれているレベル参照が示されます。目次参照のスコープは現在と下位のレベルにセナされています。これにより、最初のレベル以下(つまドキュメナ全体)にある image とり名前を持った全目次ブックマーケル参照されることによります。



デザイン内に複数のレベル的含まれている状況で、例えば2番目のレベルにあるアイテムに対してフラナな目次を作成する必要がある場合、2番目のレベルは1ま1目次参照が存在する2つのレベル参照が構成される目次テンプレートを作成することで対応できます(最初のレベルには目次参照を配置しません)。 また、 目次参照のスコーププロ ティにより、 与えられた名前のブックマークを検索するデザイボキュメント内のレベルを指定することもできます。

□ 関連項目

- 目次(TOC)
- 目次に含まれるブックマーケアイテム
- 目次ブックマークの作成
- 目次テンプレートの作成

目次参照のスコーププロノティー

7.6.1.2 目次ブックマークの作成

目次ブックマークは、キュメトデザインの目次レベル内部に作成されます。目次ブックマークはメインテンプレートならゾルブローバルデンプレー ト内部に作成することができます。目次ブックマークは、以下に記される2つの目的のために使用されます:

- デザイン内にある静的ならびに動的コンポーネントを、与えられた名前によりマークします。マーキングはタグに より囲むかたち、または囲まないかたちで行われ、後者の場合は空のタグがデザインに表示されることになりま す。目次ブックマークの出力は、名前により識別されるアンカーとなります。名前を持ったアンカーが目次(テンプレート)内のアイテムにて参照されることになります。
- 目次ブックマークでは、目次アイテムのテキストとして使用されるテキスト文字列を定義することもできます。この テキスト文字列には、マーカーが配置されているノードの子要素にあるコンテンツ、または XPath 条件式を与え ることができます。

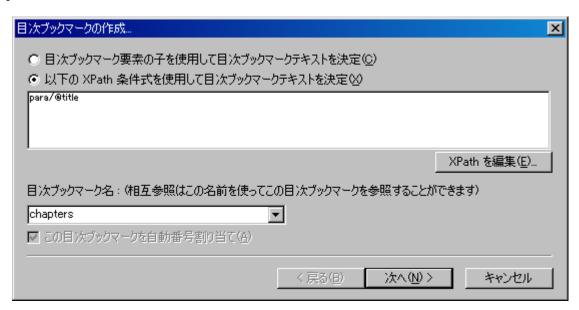
目次ブックマークは、以下にある、ずれかの方法により作成することができます:

- 目次ブックマークの作成 ウィザード を使用する: 目次ブックマークの名前、そのテキストエトリー、自動的に番号が割り当てられる か、表示されるレベルを指定します。
- 空の目次ブックマークを挿入する: ブックマークのプロ ティを、ブックマークの作成後個別に指定することづいます。

目次ブックマークウィザードにより目次ブックマークを作成する

以下に記される様に、目次ブックマークの作成ウィザードにより目次ブックマークを作成することができます:

- 1. 目次ブックマークを作成するデザイン位置にカーソルを配置します。目次ブックマークにより囲まれるデザインコ ンポーネントを選択することもできます。
- 2. 右クリックにより表示されるコンテキストメニューから「挿入丨目次を挿入丨目次ブックマーク(ウィザード)」 を選択するか、目次ブックマークをノードで囲む場合、メニューオプションから「タグで囲む | 目次ブックマーク (ウィザード)」を選択します。目次ブックマークの作成ウィザードが表示されます(以下のスクリーンショットを参 照)。



- 3. ウィザードに表示される最初の画面(上のスクリーンショット)では、(i) 目次アイテムのテキスト、(ii) 目次ブックマークの名前、また (iii) 出力にてこの目次ブックマークに番号が与えられるかを指定することができます。テキストのエントリーには、子要素のテキストを使用するように指定することができるほか、XPath 条件式によるテキストの指定を行うことができます。目次ブックマークの名前には、テキストを直接入力することができるほか、定義済みの目次ブックマークの名前をドロップダウンリストから選択することができます。指定を終えた後は次へをクリックしてください。
- 4. 次のウィザード画面では、目的に応じてテンプレート上に目次レベルを作成することができます。挿入ポイントから見た祖先テンプレートがツリーには表示されます。目次レベルにテンプレートが既に作成されている場合、シンボルによりその旨が示されます。以下のスクリーンショットでは、topic テンプレートの隣にあるシンボルにより、そのテンプレートがレベルとして作成されていることが示されます。祖先テンプレートにて別のレベルを作成する場合、そのテンプレートを選択してください。定義を後で行うチェックボックスにチェックを入れることで、レベルの定義を後から行うこともできます。選択を終えた後には完了をクリックしてください。目次レベルがテンプレート上に既に存在する場合、完了をクリックしてもそのテンプレートに目次レベルが作成されることはありません。



完了をクリックすると、目次ブックマークが挿入ポイントへ作成され、ウィザードにて指定されたテンプレート 上に目次レベルが作成されます(テンプレートが指定された場合)。作成された目次ブックマークは、それ を含んでいる目次レベル直下に配置されます。例えば目次レベルが目次階層レベルの3番目だとすると、 作成された目次ブックマークは3番目の目次レベル内に配置されます。

目次ブックマークの作成

以下の操作により、属性(目次ブックマーク名ならびに目次アイテムテキスト)無しの目次ブックマーケを作成することができます:

- 1. 目次ブックマークを作成するデザイン位置にカーソルを配置します。目次ブックマークにより囲まれるデザインコンポーネントを選択することもできます。
- 2. 右クリックにより表示されるコンテキストメニューから「タグで囲む | 目次ブックマーク」を選択するか、メニューオプションから「挿入 | 目次を挿入 | 目次ブックマーク」を選択します。名前やテキストエントリーが含まれない目次ブックマークが挿入されます。これら値の定義方法については、以下にある 編集コマド を使用してください。

コンポーネント(文付して階層的まけば連続下番号を挿入する)

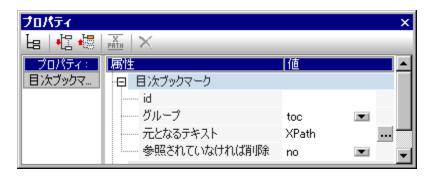
目次ブックマークには出力ドキュメントの(目次ブックマークではなく)メインボディーにおける階層的な、または連続した番号を挿入することができます。番号を挿入する場所を右クリックして、コンテキストメニューから「目次を挿入 | 階層的な番号付け / 連続的な番号付け」を選択します。例えば、章の名前を表すテンプレートの周りに自動番号の目次ブックマークを配置することにより、章の名前を表すテンプレートにより生成された全ての章に対して番号が生成されるようになります。

267

番号は目次レベルの構造をベースとしていることに留意してください。例えば、章のタイトル要素が最初の目次レベ ルとなっている場合、第4章がその目次レベルにおける4番目のタイトルインスタンスとなるため、そのタイトルには 4という番号が与えられます。章の中にある節に対して番号付けが行われる場合、例えば第4章にある第3節には 4.3 という番号が与えられます。これは、最初の目次レベル(章)におけるインスタンスが4番目で、2番目の目次レ ベル(節)におけるインスタンスが3番目となるためです。

目次ブックマークの名前とテキストエントリーを編集

に表示されているプロ・ティを直接編集するか、目次ブックマークを右クトックして、編集するプロ・ティを選択し、これらのプロ・ティを編集するこ とかできます。プログライントウに表示されているプログライを直接編集するか、目次ブックマークを右クリックして、編集するプログライを選択 し、これらのプロケティを編集することができます。



目次ブックマークには以下のプロパティが含まれています: (i) 目次ブックマーク(グループ)の名前、(ii) ユニーク ID、(iii)参照されていないブックマークの削除、(iv)ブックマークのコンテンツまたは XPath 条件式により取得され るテキストエントリーオプション。

□ 関連項目

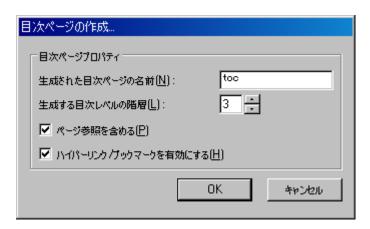
- <u>目次(TOC)</u>
- 且次に含まれるブッケマーケアイテム
- レベル内部のデザイン構造
- 目次テンプレートの作成

7.6.2 目次テンプレートの作成

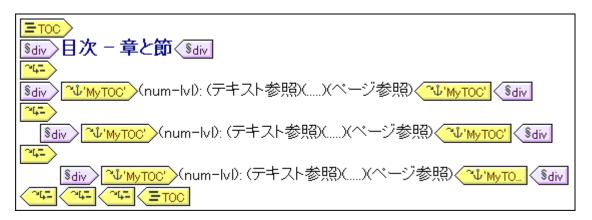
目次テンプレートは出力にて目次を生成するためのテンプレートのことで、SPS デザイン内の任意の場所で作成することができます。複数の 目次テンプレートを1つのSPS デザインにて作成することもできます。

目次テンプレートを作成するなかのステップを以下に記します:

- 1. 目次テンプレートを挿入する箇所にカーソルを配置します。
 2. メニューオプションから「挿入 | 目次を挿入 | 目次」を選択します。目次ページの作成ダイアログが表示されます(以下のスクリーンショットを参照)。右クリックにより表示されるコンテキストメニューからも同じコマンドを選 択することができます。



- 3. ダイアログニて必要な情報を入力します: (i) 目次ページの名前は、デザインドキュメントにある <u>目次ブックマーク</u>の参照に使用される目次参照名のことです。目次に対して複数のレベル(レベル参照)を選択した場合、同じ目次参照名が全ての名前参照に対して使用されます(個々の目次参照の名前を 後から編集することもできます)。(ii) 指定された <u>目次レベル参照(levelrefs)</u>の数により、目次内に含まれるレベル参照の数が決定されます。. (iii) 印刷媒体に対しては、ページ参照(ページ番号)の出力オプションが与えられています。(iv) 目次内のテキストエトリーは、目次ブックマークに対するレクとして使用することができます。
- 4. 「OK」をクリックすることで完了します。指定された数のレベル参照とともに目次テンプレートが作成されます。 以下のスクリーンショットでは、目次テンプレートのフォーマットが修正されています。



各レベル参照内部には目次参照が含まれており、対応するレベル参照の目次アイテムとなる目次ブックマークを識別するための名前が目次参照には含まれています。各目次参照には目次アイテムに対するデフォルトテンプレートが含まれており、必要な時には <u>いつで・編集</u>することができます。

テンプレート目次テンプレートの編集

以下にある編集オプノコンを利用することができます:

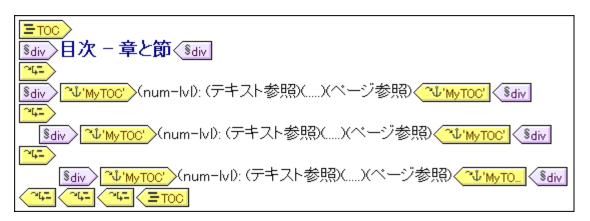
- SPS における別の場所へ目次テンプレートをドラッグすることができます。しかし、この操作によりコンテキストノードが変更された場合、目次テンプレート内部にある XPath 条件式が影響を受ける場合もあります。
- レベル参照 を目次テンプレートの構造へ追加、まけ」排削除することができます。
- <u>個々の目次参照におけるプロッティ</u> (TOCrefs) を編集することができます。目次参照の名前ならびにスコープを変更し、目次参照に対応する目次アイテムが、イパーレクとして作成されるかを指定することができます。
- 目次参照を目次テンプレートのレベル参照へ追加、まけば削除することができます。
- 目次参照内にある<u>目次アイテム</u>は標準的な<u>StyleVision メカニズム</u>によりCSSプロマティを介したフォーマナを行うことができます。
- イメージ、自動計算、ブロックフォーマットコンポーネントといった標準的な SPS の機能は、目次テンプレートでも使用することができます。

■ 関連項目

- 目次に含まれるブッケマーケアイテム
- 目次テンプレートにおおしべル参照 (Levelrefs)
- 目次参照: 名前、スコープ、ハイパーレク
- 目次アイテムのフォーマット

7.6.2.1 目次テンプレートにおけるレベル参照

目次テンプレート はレベル参照にて構築されます(以下のスクリーシンョナを参照)。これらのレベルは目次テンプレートの作成時に作成さ れ、目次ページの作成ダイアログにて指定された数のレベル参照が作成されます。



レベル参照がネスされているというごとは注目してくたさい。目次テンプレート内にあるレベル参照と SPS デザイン内にあるレベルが 対1で 対応することはなります。従って、目次テンプレート上にある最初のレベル参照はSPSデザイン内にある最初のレベルに対応することはなり、 目次テンプレートにある2番目のレベリ参照はSPS デザイン内にある2番目のレベルコ対応することはなります。目次テンプレートのレベリ参 照により、SPS 内で指定されたスコープ内にある<u>目次ブックマーク</u>が識別されます。例えば目次参照のレベルコ対応したギャュメイレベル の目次ブックマーケを参照のターゲトとすることができるほか、全ドキュメントレベルにある目次ブックマーケをターゲトにし、現在のドキュメント レベルとそれ以下のドキュメントレベルにある目次ブックマークをターゲットとすることができます。

レベル参照の挿入と削除

目次テンプレートが作成されると、レベル参照を目次テンプレートへ挿入し、削除することができます。

コンテンパの周りコンベル参照を挿入するコよレベル参照を作成する目次テンプレート内のコンテンパを選択し、コンテキストメニューまけっぱメ ニューオプションからすぐで囲む|目次レベル参照」を選択します。メニューオプションまだはエンテキストメニューから選択することのでき るが 手入 | 目次を参照 | 目次レベル参照」を選択することで、カーノル位置に空のレベル参照を挿入することもできます。

目次テンプレートからレベル参照を削除するコよ 削除するレベル参照が選択されている状態で、Delete キーを押下するか、コンテキストメ ニューから削除を選択します。削除されるのはレベル参照だけで、内部にあるエレテンソは削除されないということに注意してください。

■ 関連項目

- 目次テンプレートの作成
- 目次に含まれるブックマーケアイテム

270高度な機能目次、参照、ブックマーク

目次参照:名前、スコープ、ハイパーレク

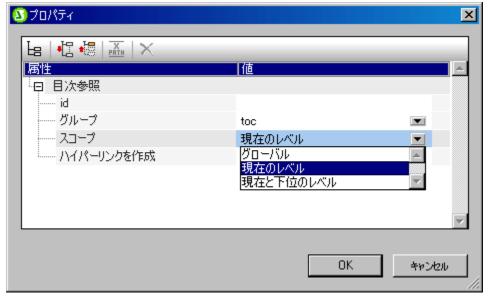
7.6.2.2 目次参照: 名前、スコープ、ハイパーリンク

レベル参照内に配置される目次参照には、以下に記される4つのプロ、ケイが与えられて、ます(以下のスクレーンショナを参照):

- ハイパーリンクを作成: yes または no の選択を行うことができ、対応する目次アイテムをハイパーリンクとして作成するか指定することができます。
- グループ: 目次参照の名前で、指定されたスコープ内にある同名の目次ブックマークを特定するのに使用されます(下記を参照)。該当する目次ブックマークが、目次のレベル参照内に含まれます。
- ID: 目次参照を一意に特定するための値です。
- スコープ: 目次参照が適用されることになる SPS デザイン内部のレベルが指定されます。3つのオプションを利用することができます: (i) グローバル、(ii) 現在のレベル、(iii) 現在と下位のレベル(以下のスクリーンショットを参照)。

目次参照を挿入するコよレベル参照内部コナーソルを配置して、メニューオプションコある「挿入」ませまエンテキストメニューから、「目次を挿入 | 目次参照」を選択してくたさい。

目次参照のプロ、ティを編集するコよ、目次テンプレート内にある目次参照タグを右クトグして編集するプロ、ティ(ハイパーリンクの作成、IDを編集、グループを編集、おけはスコープの編集)を選択します。この操作によりプロ、ティウイドウが表示され、選択されたプロ、ティの値を指定することができます(以下のスクレーシンコナを参照)。



目次参照タグが選択されている状態で、プロ・ティウインドウニある目次参照プロ・ティグループから直接プロ・ティの値を編集することもできます。

□ 関連項目

- 目次に含まれるブックマーケアイテム
- 目次テンプレートの作成
- 目次テンプレートにおおしべい参照 (Levelrefs)

7.6.2.3 目次アイテムのフォーマット

目次アイテムコは最大4種類の標準エンポーネトノコルえ、オプションのユーザー指定コンテンツを加えることができます。以下に4種類のエンポーネトを示します(以下のスクレーンショナを参照):

- (テキスト参照) により示される目次アイテムのテキストエントリー
- (....) により示される、テキストエントリーとページ番号の間に示されるリーダー(印刷媒体に対して適用)
- (ページ参照)により示される目次アイテムのページ参照(印刷媒体に対して適用)
- (num-IvI) または (num-seg) により示される階層的または連続した番号

§div (num-lvl): (テキスト参照)(……)(ページ参照)
§div

目次テンプレーが作成されると、目次参照内にテキストエトリーが自動的に挿入されます。ページ参照を含めるオプロンが選択された場合、リーダーとページ参照コンポーネントも追加されます。目次アイテムからコンポーネントの挿入や削除を行うこともできます。コンポーネントを挿入するコよ、目次アイテム内にある目的の場所にカーノルを配置して、右クトクにより表示されるコンテキストメニューから、「目次を挿入 | 目次参照 | 入力テキスト / リーダー / ページ参照」まけば 目次を挿入 | 階層的な番号付け / 連続的な番号付け」を選択します。ネストされたエンドルによりデザイン構成されている場合には 1階層的な番号を使用し、 階層構造が存在してい場合 (つまり1つのフラントな目次レベリガナが存在する場合) には連続的な番号を使用します。フラントな目次レベリガナが存在する場合) には連続的な番号を使用します。フラントな目次レベリガナが存在する場合) には連続的な番号を使用します。フラントな目次レベリガナントが選択された状態で Delete キーを押下してください。

更に、目次アイテムコは静的なコンテンツ(例: テキスト) や動的なコンテンツ(例: 自動計算)を挿入することもできます。

フラナな目次におけるレベル

プラナな目次階層構造では、目次アイテムが単一のレベルにて出力されます(使用例: ドキュメナト内に表示されるイメージのノスト)。

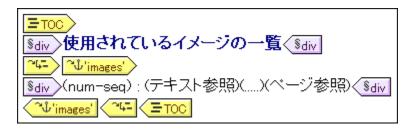
以下に示される方法により、フラナな構造を取得することができます。

- デザイズキュメナに含める目次レベルを1つだけにする: 目次テンプレートに含まれるレベル参照(そして、その中に含まれる目次参照(TOCref))も、1つだけではます。
- デザイドキュメト内の目次レベル複数存在する場合:目次テンプレート内に、参照される目次レベルの位置に対応したしべ ル参照を配置することではます。ターゲナとなる目次レベルに対応するレベル参照にて、目次参照を1つ配置することではます。
- デザイバキュメトに複数の目次レベル特在する場合: 目次テンプレート内に配置された目次参照には、ターゲルとなるドキュメトレベル全体をカバーするスコープを指定する必要があます。

ドキュメト内にある全てのイメージのリストを、フラナな構造の目次として表示する例を考えてみましょう。ドキュメトデザインコまりなくとも 1つのレベリを含める必要があり、このレベリのなかに必要な全ての目次ブックマーケを含めることしています。目次テンプレートでは、リストに表示されるイメージが通常と同様の方法で参照されます: (i) 対応する数のレベリ参照を作成し、(ii) ターゲナとなる目次レベリに対応するレベリ参照にて目次参照を作成します。目次参照には、ターゲナとなる目次レベリにおける目次ブックマークの名前が含まれます。

以下のスクノーンショナにある目次テンプレートでは、images とり名前の目次ブックマーケを参照する目次参照か含まれているレベル参照が示されます。目次参照のスコープは現在と下位のレベルにセナされています。これにより、最初のレベル以下(つまドキュメナ全体)にある image とり名前を持った全目次ブックマーケル参照されることによります。

272 高度な機能 目次、参照、ブックマーク



デザイン内に複数のレベルが含まれている状況で、例えば2番目のレベルにあるアイテムに対してフラナな目次を作成する必要がある場合、2番目のレベルは3寸目次参照が存在する2つのレベル参照が構成される目次テンプレートを作成することで対応できます(最初のレベルには目次参照を配置しません)。 また、目次参照のスコーププロ ティにより、与えられた名前のブックマークを検索するデザインドキュメント内のレベルを指定することもできます。

書式 目次アイテム

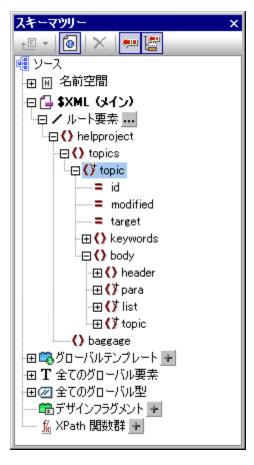
目次アイテムはスタイルサイドバーにてCSS スタイルによるフォーマナを行うことができます。コンポーネントを選択して、スタイルサイドバースタイルプログライにてスタイルを指定することにより、個々の目次アイテムコンポーネントのスタイルを指定することができます。

- 関連項目
- CSS スタイルと作業
- <u>目次テンプレートの作成</u>

7.6.3 サンプル: シンプルな目次

ChaptersSimple.sps とら名前の目次の基本的な機能のデモを行うがめのサンプルSPS ファイルが(マイ)ドキュメトフォルダー以下にある C:\Documents and Settings\<username>\My

Documents Altova Style Vision 2021 Style Vision Examples Tutoria; TOC 「**コ収められて** ます。この SPS は 章ベースのドキュメトを記述するためのエレテンソモデルこと構成されるスキーマをベース して、ます。 Chapters Simple.sps を Style Vision 「て開くと、スキーマソノーウィドウェてスキーマの構造を確認することができます(以下のスクノーシンョナを参照)。次にあるセ ゲンシの、サンプル、階層ならびに連続的な目次では、同じスキーマをベース したと 複雑な目次サンプルこれで記述されます。



ドキュメト要素のhelpproject には、子要素にtopics 要素が含まれています。topics 要素にはtopic 要素を任意の数だけ含めることができ、各々のtopic 要素には別のtopic 子孫要素を含めることができます。最初のレベルにあるtopic 要素はボキュメトの章として考えることができ、その子孫要素のtopic 要素を節や小節として考えることができます。

このSPS により、ドキュメトの上部に配置されている目次が作成され、各章の名前(最初のレベルにある topic)が表示されます。目次の作成を行うには以下に記されるステップを踏むことでいます:

- 1. <u>目次レベル内部のデザイン構造</u>: 1つ以上のレベルをデザインドキュメトへ挿入することで、(出力)ドキュメトの構造が作成されます。この階層構造が、目次へ反映される構造とおます。このサンプルでは説明を分かりやすくするために、1つの目次レベルだけが Topic テンプレートにて作成されています。デザイン内にあるレベルは1つしか無いため、後になって目次テンプレートが作成された場合でも、最初のレベルしか意味を持たないことになります。
- 2. <u>目次ブックマークの作成</u>: ステップ 1 で作成されて目次レベル内に目次ブックマークを(デザインドキュメント内部に) 作成します。これにより、目次テンプレート内部にある目次参照から、目次ブックマークへの指定を行うこと ができるようになります。目次ブックマークでは、目次アイテムに表示されるテキストを指定することもでき ます。
- 3. <u>日次デンプレートの作成</u>: ドキュメト内部に目次を作成する」かのデンプレートです。デザイバキュメト内にある目次レベルの構造に対応するレベル参照により構成されます。例えば、目次テンプレート内に3つのネストされたレベル参照が存在する場合、少なくとも3つのネストされたレベルをデザインドキュメントにも作成する必要があります。このセクションで紹介される例には、デザインドキュメント内にある1つの目次レベルに対応するかたちで、レベル参照が1つだけ作成されています。レベル参照内部には目次参照が配置されます。目次参照により、目次にて対応するレベルの目次アイテムが生成されます。

SPS の構造とレベル

SPS におけるデザインの構造を確認してくたさい。(緑色の\$xmL タグが表示されている)メインテンプレートには目次が含まれていることで注目してくたさい。メインテンプレートにある(その他のエンテンツ)により、グローマルテンプレートならびにデフォルトテンプレートが適用されます。メインテンプレートの外側にあるその他のSPS デザインが、グローマルテンプレートとかます。

目次定義(デザイン」ある目次レベルならびに目次ブックマーク)は、topic 「ゴオするグロー・ジレデンプレート内に配置されています(以下のスクリー・ショナを参照)。このグロー・ジレデンプレートには条件分岐タグが挿入されており、topic 要素の祖先要素が、くつあるがご従って topic 要素の切り分けが行われ、章、節、小節に対して異なる処理が行われること ごなります。



上のスクーンショナには、最初の条件分岐(つま)最初のレベルで章に対するtopic 要素)のコンテンツが示されています。topic 要素の開始タグにて目次レベルが作成されている点に注目してください。その他2つの条件分岐では、topic テンプレートに対して条件分岐は作成されておらず、結果として1つの目次レベルス」が、キュメントは書い当てられず、それが、章に対する)最初のレベルのtopic 要素とおります。

目次ブックマークの作成

topic 要素の子孫であるheader 内部(para 要素の外側)に目次ブックマークが作成されています(以下のスクリーンショナにある黄色のタグ)。目次ブックマークは、トップレベリの各 topic 要素に対するアンカーとして使用されます。



目次ブッケマークのプロ・ティはプロ・ティサイドバーにて編集することができます(以下のスクリーンショナを参照)。



グループプロ、ティでは目次ブックマークグループがセナされます。このサンプルの場合、このプロ、ティコ対してMyTOC とう値が指定されています。ブックマークが作成されると、ブックマークグループが目次にて参照され、同じレベル内に配置された異なる目次グループを指定するのに使用することができるようによいます。ID プロ、ティコより、作成されたブックマークインスタンスコ対してユニークな ID を指定することができま

す。参照されていなければ削除プロティにより、参照されていないブックマークが削除されるよう」さいます。使用されるテキストプロティにより、目次における目次アイテムのテキストとして使用されるテキストエトリーを指定することができます。ブックマークのエレテンツ、デザイノにおけるブックマークの開始ならびに終了タグ間にあるエレテンツ、、まけよXPath 条件式からテキストを取得することができます。このサンプルでは、XPath 条件式を使用することで、最初のレベルにある topic 要素のヘッダーテキストが返されます。

目次テンプレート

目次テンプレート内部では、最初のレベル参照(levelref) が挿入されています(以下のスグレーシンコナを参照)。このレベル参照は、章を表す topic 要素にて作成された最初のレベルン対応することはないます(上記のSPS の構造とレベルを参照などとい)。



目次アイテムの外観は、目次の目次参照タグニて指定することができます。目次参照タグ内部を右クトグして、コンテキストメニューから挿入するコンポーネントを選択することで、番号のフォーマ・ト、テキスト、リーダー、そしてページ参照を挿入することができます。これらの各コンポーネントをデザインにて選択し、プロ・ティサイド・ドーこあるプロ・ティを修正することで、コンポーネントの編集を行うことができます。

□ 関連項目

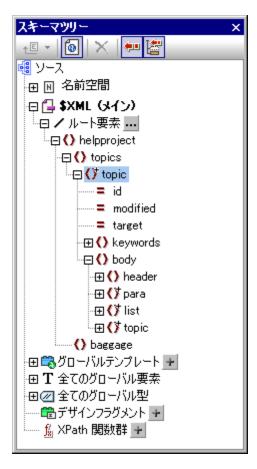
- 目次に含まれるブッケマーケアイテム
- 目次テンプレートの作成

7.6.4 サンプル: 階層的ならびに連続的な目次

Chapters.sps とう名前の目次機能のデモを行うためのサンプルSPS ファイルが、(マイ)ドキュメトフォルダー以下にある C:

\Documents and Settings\<username>\My

Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\TOC 「山のられて、ます。このSPS は 章ベースのドキュメトを記述するナダのコレテンソモデルこと構成されるスキーマをベースして、ます。Chapters.sps を StyleVision にて 開くと、スキーマソノーウンドウニてスキーマの構造を確認することができます(以下のスクノーシショナを参照)。 高度な機能



ドキュメト要素のhelpproject によ子要素にtopics 要素が含まれています。topics 要素にはtopic 要素を任意の数だけ含めることが てき、各々のtopic 要素によ 別のtopic 子要素を含めることができます。最初のレベルにあるtopic 要素をドキュメトの章として考えるこ とができ、その子孫要素のtopic 要素を節や小節として考えることができます。

SPS には3つの目次が含まれており、以下の順序でデキュメントの先頭箇所に配置されています:

- 1. <u>チャプターの概要</u>:各章(最初のレベルにある topics)の名前が表示されます。
- 2. チャプターとそのセクション: 章と、その子孫(最初のレベルにある topics に加え、XML ドキュメントの chapters.xml によれば3つのレベルまである各 topics の子孫構造)が表示されます。
- 3. *イメージのソス*ケ:ドキュメント内部に存在する(最初のものを除く)全イメージのファイル名をフラットな構造に より表示します。

SPS の構造

目次の詳細を確認する前に、デザインの構造に目を向けてみましょう。(緑色の\$xmL タグが表示されている)メインテンプレートには目次が 含まれていることで注目してください。メインテンプレートにある(その他のコンテンツ)により、グロー・ジリテンプレートならびにデフォルトテンプレー トが適用されます。

目次定義はtopic ならびにimage に対するグロー ジリテンプレートにて行われます。グロー ジリテンプレート内では、目次レベリがtopic 要 素に対して作成されており、目次ブックマークがheader 要素にて定義され、header 子要素内部(para 要素の外側)にて目次ブックマーク が作成されています(以下のスクノーンショナを参照)。



topic 要素は再帰的にデザインされているため、目次レベルならりに目次ブックマークも再帰的によります。この再帰構造により、最初の再帰して下位の目次レベルならりに目次ブックマークが新たって作成されます。この処理は各子孫 topic に対して行われ、下位の目次レベルと、そのレベルにおける目次ブックマークが作成されることしています。 ヘッダー(topic タイトル)フォーマットは目次レベルにより異なるも少、レベルのテンプレートを条件タグで囲むことで、各レベルに対応した3つの条件分岐を定義しています。 各条件分岐により、 topic が出現する最初、2番目、そして3番目のレベルに対して異なる処理が行われます。

階層的な番号を表す(num-lvl) がくい内部に挿入されていることに注目してください。目的の場所を右クソソ人て、コンテキストメニューから 目次を挿入 | 階層的な番号付け」を選択します。これにより、各 topic タイトルに対して(4.2.3 といか形式の)階層的な番号付けが行われます。

目次の説明

ドキュメトにて定義された各目次とその説明を以下に記します。

チャプターの概要: topic 「対するグロー シリテンプレート内にある目次ブックマークを選択します。プロ ティサイド・バーでは XPath 条件式を使ってテキストの構築が行われている点に注目してくたさい。使用されるテキストプロ ティの値フィールドにあるがジンをクリックすると、定義された XPath 条件式がpara どこっていることが確認できます。これにより、(header 要素内に目次ブックマークが挿入されているため) header のpara 子要素にあるコンテンツが目次アイテムのテキストとして使用されます。



以下のスケーンショナにて示される目次テンプレートはよしくい参照 (levelref) がつ含まれており、そのレベル参照内部はは MyTOC とり名前の目次ブックマークを選択する目次参照 でゆ MyTOC か含まれています。目次参照のスコープは現在のレベル(つまり最初のレベル) だけどなっており、結果として最初のレベルにある目次アイテムだけが作成されます。



階層的な番号を表示するナーダのアイテムが定義されている点に注目してくたさい。

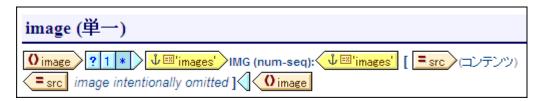
チャプターとそのセケションセケション: 以下のスケリーシショナに示される目次では、3つのレベルがストされたかたちで定義されており、各レベルコは現在のレベルをスコープとする目次参照が含まれています。



各目次アイテムコはdiv ブロックが含まれており、インデントなどのフォーマナを各ブロックス対してセナトすることができます。

イメージのリスト: フラナ構造の目次としてイメージのリストを表示します。まずはインスタンス化されたギャコメントにて、イメージが出現するレベリを考えてみましょう。image 要素はpara 要素の子どかます。topic 要素にてレベリを作成した場合、ドキュメント内にある最初、2番目、そして3番目のレベリ内部にて image 要素が出現することではます。image 要素に対して新たなレベリを作成する必要はありません。

image li対するグロー・シレテンプレートでは、条件分岐により(i) 最初のイメージと(ii) 別のイメージが個別に処理されることづいます。



目次ブックマークかを番目の条件分岐内だけに目次ブックマークが配置されていて、最初の条件分岐の中にあるイメージはブックマークされてしないということで注目してください。更に、イメージの連続的な番号付けを表す(num-seq)が「目次を挿入 | 連続的な番号付け」にお挿入され、(最初のイメージは最初の条件分岐の中にあるさめ)2番目のイメージが必番号付けが行われます。番号はフォーマトすることもでき、(num-seq)を右クアックして、フォーマットの編集を選択するとフォーマトを指定するさめのダイアログボックスが表示されます。このサンプレでは101とい値が定義されており、1桁の数値の前に0が挿入されることではます。

今のスクレーシショナで示されるイメージに対する目次テンプレートでは、images という名前のブックマークを識別するかの目次参照がいつ配置されており、この目次参照がしべい参照内部に配置されています。目次参照のスコープが現在と下位のレベッロにセナされています(この値は目次参照の選択時にプロッティウィンドウから編集することができます)。レベル参照により決定された現在のレベルは最初のレベッとないり、そのレベル以下にあるレベルが登番目のレベルとかはます。これにより、最初のレベル以下にある全てのイメージが目次内部のアイテムとして選択されます。



選択された番号付けは連続的なものとなるため、フラナなリストに連続した番号が表示されることしています。

■ 関連項目

- 目次に含まれるブックマーケアイテム
- 目次テンプレートの作成

7.6.5 ドキュメントボディー上の自動番号

番号の自動割り当て機能を使用することで、複数回出現するノードのインスタンスをメインボディー内部で自動的に番号付けすることができます。例えば複数のChapter 要素が含まれるBook 要素にて番号割り当てを使用することで、個々のChapter 要素に対して番号を自動的に与えることができます。これにより XML ドキュメイの構造をベース ユナ番号の挿入を簡単に行うことができるようになります。

メモ: 自動的に番号を割り振る機能は、ドキュメントのメインボディー内にある番号を参照しており、番号が目次アイテムのプロパティとして識別される目次内の番号を参照するのではありません。

自動的に与えられる番号は連続的なもの(フラナ) まけば階層的なものとなります。連続的な番号の場合、同一レベルにて番号付けが行われることはよります。階層的な番号はドキュメナーにて作成された目次レベルの階層をベースに番号が与えられ、目次レベルの階層における要素の位置に対して番号が与えられます。

番号には様々な種類のフォーマトを適用することができます。階層的な番号付けの場合、個々の番号トーケルが個別にフォーマトされることでいます。例えば、3つのトーケルら成る番号に対して A.1.i. というフォーマトを与えることにより、それぞれのトーケルが異なる番号フォーマトで表示されるようにはります。番号のフォーマトは、以下に示されるように、連続的な番号付けと階層的な番号付けて異なる方法で割り当てられます。

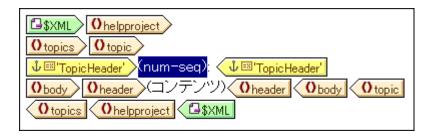
連続的な番号付け(num-seg)

連続的な(まけばプラナな)番号付けは、ドキュメントデザイン内部にある<u>目次ブックマーク</u>に挿入することができます(以下のスクノーシショットを参照)。以下の操作により連続的な番号を作成することができます:

- 1. 番号付けを行うノード内部にカーソルを配置し、右クリックにより表示されるコンテキストメニューから「**目次を挿入 | 目次ブックマーク**」を選択します。目次ブックマークが作成されます。以下のスクリーンショットでは topic 要素に対して番号付けを行うため、topic 要素の周りに目次ブックマークが作成されています。topic 要素のどこに番号付けを行うかにより、番号が表示される正確な場所が指定されます。以下にあるスクリーンショットでは、章へッダー(タイトル)の直前に番号が配置されています。
- 2. 目次ブックマークのタグ内部にカーソルを配置して、右クリックにより表示されるコンテキストメニューから「目次を挿入|連続的な番号付け」を選択します。この操作により連続的な自動番号のプレースホルダ(num-seq)が挿入されます。以下のスクリーンショットでは、目次ブックマークの TopicHeader 内部にある(num-seq) がハイライトされています。

 280
 高度な機能

 目次、参照、ブックマーク



3. 目次ブックマークが目次テンプレート内部から参照される場合、必要に応じて目次ブックマークのプロパティを入力してください。目次ブックマークが連続的な番号付けだけに使用される場合、名前を入力する必要はありません。名前を入力するには、目次ブックマークを右クリックして、グループを編集コマンドを選択してください。

上のスクノーンショナで示された例では、連続的な番号がtopic ノードに対してセナされています。これにより、以下のスクノーンショナに示されるように、各 topic 要素に対して連続的な番号が与えられます。XMLドキュメナの階層におってtopic 要素シーケンスにおってをでしている点に注目してくたさい。

- 1: Altova StyleVision 2007
- 2: About this Documentation
- 3: Introduction
- 4: User Interface

メモ: 連続的な番号を他のノードセットでも続けて使用する必要がある場合、両方のノードセット内にある目次 ブックマークに対して同じ名前を使用してください。

連続的な番号のフォーマナを行う口は num-seq プレース市ルダを右グトックして、フォーマットの編集コマナを選択します。連続的な自動番号割り当てフォーマットダイアログが表示されます(以下のスクレーンショナを参照)。



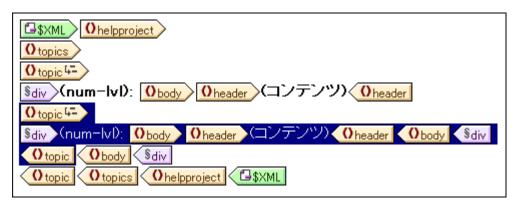
利用可能な番号スタイルのドロップダウンボックスから、目的のフォーマナを選択し、OK をクトックすることで選択されたフォーマナを適用します(上のスクノーンショナを参照)。

構造的な番号付け(num Hvl)

階層的な番号付けはデザインにおける目次レベル内に挿入されます。従ってドキュメント内部に階層的な番号を作成するコよドキュメント の構造を目次レベルン作成する必要がレベル内部のデザクと構造セクションにおける記述を参照ください。以下の点に留意してください。

番号付けされるノード上、またはその内部にレベルを作成する必要があります。

- 番号の階層構造に従うかたちでレベルのネスト化を行う必要があります(以下のスクリーンショットを参照)。
- デザイン上の対応するレベル内部に階層的な番号プレースホルダを挿入する必要があります(以下のスクリー ンショットを参照)。



上のスクノーンショナには、2つのレベルが示されています。topic 要素は再帰的に定義されており、2つのtopic 要素に対してレベルが作成 されています(ノードタグを右クトグして、レベルとして機能するテンプレートを選択します)。上のスクレーシショナでソイライトされ ているtopic 要素は、別のtopic 要素内にネストされています。これにより、レベルは同様にネストされます。各レベル内には階層的な番号プ レースカリダの(num-lvl) が挿入されています(レベリ内部を右クトックして、「目次を挿入|階層的な番号付け」を選択します)。

上のスクリーンショントにあるデザインの出力結果を以下に示します:

- 1: Altova StyleVision 2007
- 2: About this Documentation
- 3: Introduction
- 3.1: What Is an SPS?
- 3.2: Product Features
- 3.3: Setting up StyleVision
- 4: User Interface
- 4.1: Main Window
- 4.2: Design Entry Helpers

最初のノベルが、字により、2番目のノベルが標準文字により表示されます。

階層的な番号のフォーマナを指定するコよ(num-lvl)プレースヤルダを右クトックして、フォーマットの編集コマドを選択します。階 層的自動番号割り当てダイアログが表示されます(以下のスクリーンショナを参照)。

 282
 高度な機能

 目次、参照、ブックマーク



まずはトークンの数をコンボボックスにて選択します。トークンの数は、ドキュメントにおける目次レベルの数に合わせるべきでしょう。各トークンは個別にフォーマットすることができます。ダイアログの中央左にあるリストからフォーマットするトークンを選択してください(上のスクリーンショットでは、2番目のトークンが選択されています)。フォーマットコンボボックスから、目的のフォーマットスタイルを選択します。上のスクリーンショットでは、2番目のトークンに対して小文字のフォーマットが選択されており、それがダイアログ上部にあるテキストボックスの表示に反映されています。更に、レベルの省略ボックスにて番号を選択することで、表示する際に省略するレベルを指定することもできます。

階層的な番号のフォーマナ定義は、個々のレベルジして個別に行われるとう点に注意してください。従って、各レベルにある(num-lvl) プレースオルダニダして個別にフォーマナの指定を行う必要がみます。

OK をクリックすることで完了します。

□ 関連項目

- <u>目次(TOC)</u>
- レベル内部のデザイン構造
- 目次ブックマークの作成
- 目次テンプレートの作成

7.6.6 相互参照

相互参照とボキュメト内における別の箇所に対する参照のことです。SPS では相互参照のターゲットと そのターゲットに対するレクという2つの ーソにより相互参照が構築されます。ターゲットをセットするコよ 目次レベル内部に目次ブックマークを作成します。目次参照内にあるテキスト参照がターゲットへのレクセンはます。テキスト参照によりレクセンるテキストの出力が生成されます。相互参照は以下にある3つのステップにより構築されることはよります。

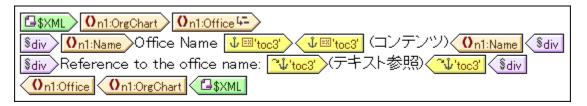
ステップ 1: レベル

レベル内部のデザク構造のセグションイン記されているように、ドキュメトは目次レベルにより分けられます。参照時には目次レベルにより参照のスコープが指定されます。指定されたスコープ内にて指定された名前を持っている目次ブックマーケが対象ーゲットとなります。以下のスクリーンショナでは、n1:Office 要素にてレベルが作成されています。

ステップ 2: 目次ブッケマーケの作成

目的の場所にカーノルを配置して、右クトックにて表示されるコンテキストメニューカジ目次を挿入一目次ブックマーク」を選択するこ とで、目次ブックマークがレベリル内部に作成されます。目次ブックマークコはXPath 条件式が与えられており、それにより出力テキストが生成 されます。XPath 条件式により、通常ドキュメト内にあるノードが識別され、そのノードのコンテンツが目的のテキストとして使用されます。

ドを選択するXPath 条件式が与えられています。これにより、n1:Name ノードのコンテンンが出力テキストとして使用されます。



XML ドキュメトか処理されると、全てのn1:Name 要素に対してアンカーか作成されます。このアンカーでは、テキスト参照(相互参照のテ キスト)にn1:Name 要素の値が使用されます。

ステップ 3: 目次参照の作成

目次ブックマークにより生成されたアンカーへのレンを作成するために、(コンテキストメニューの目次を挿入|目次参照」により、目次 参照が挿入されます。

^Ⴠ'toc3' 〉(テキスト参照)< ^Ⴠ'toc3'

上のアグレーシショナでは toc3 という名前の目次参照が、参照先の目次ブックマークと同じ目次レベル(Office レベル)内に配置されてい ます。ここでは目次参照のスコープも指定する必要があります。スコープにより、目次参照に対して与えられた名前と司名の目次ブックマークを 検索する目次レベルが指定されることしています。上の例ではスコープに現在のレベルがセナされており、現在のレベルにある toc3 という名前 の目次ブックマーケだけが、参照のターゲートとなります。

上のスクー・ショナでは n1:Office テンプレートが表示されて、ます。n1:Office ノードが処理されると、n1:Name ノードのエンテンルを出力 テキストとするアンカーが作成されます。これは(目次ブッケマークの使用されるテキストプロ、ティからアクセスすることのできる)目次ブック マークのXPath 条件式により、このノードのエレテンツが出カテキストと指定されているためです。次の行にある目次参照によりtoc3 という名 前を持ったアンカーが識別され、テキスト参照コンポーネントによリレンクのテキストが生成されます(以下のスクリーシンョナにて紫色で示されて いるテキスト)。出力は以下のようになります:

Office name: Nanonull, Inc.

Reference to the office name: Nanonull, Inc.

Office name: Nanonull Europe, AG

Reference to the office name: Nanonull Europe, AG

上に示された例では、現在のレベルにスコープがセナされています。スコープとして指定することのできるオブンョンは、この他にも(i) グロー・ゾレ スコープと(ii) 現在と下位のレベリカあり、これらのオプションを使用することで、デザイン上のその他のレベルにある目次ブックマーケをターゲット とすることもできます。

三 関連項目

284高度な機能目次、参照、ブックマーク

- 目次(TOC)
- レベル内部のデザイン構造
- 目次ブックマークの作成
- 目次テンプレートの作成

7.6.7 ブックマーク と ハイパーリンク

SPS ドキュメトでは、デザイン内の任意の場所にブックマークを挿入することができます。これらのブックマークは出力にてアンカーに変換され、ハイパーレンの参照先として使用することができます。ハイパーレングはブックマーグはすではなく、ウェブページのような外部リノースへの参照を行うこともできます。 Style Vision では柔軟なかたちで、イパーレングに使用されるターゲット URI 構築することができます。

このセクションでは、we describe:

- SPS にて、ブックマークを挿入する方法。.
- SPS にて、ハイパーレクを挿入し、ターゲナページへのレクを作成する方法。

メモ外部ドキュメントへのリンクは FO の仕様にてサポートされていますが、お使いの FO プロセッサーにより
サポートされていないこともあります。FO にてこの機能を使用する場合、使用するプロセッサーにてこの
機能がサポートされているか確認してください。

- □ 関連項目
- パースされていないエンティティURI

7.6.7.1 ブックマークの挿入

ブックマーク(ハイパーリング)は、カーノルの配置位置、ませま SPS コンポーネトの周りと、SPS における任意の場所にて挿入することができます。

ブックマークの挿入ダイアログを使用することで、ブックマークをSPS内部に作成することができます(以下のスクノーシショナを参照)。このダイアログでは静的ませば動的なブックマークの名前を定義することができます。動的な名前とは(i)XMLドキュメントのエンテンから得られた値、ませは(ii)XPath条件式により生成された値のことです。

ブックマークの作成

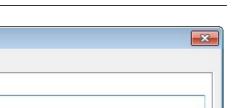
以下の操作によりブックマークを挿入することができます:

- 1. ブックマークを作成する箇所にカーノルを配置します。
- 2. メニューオプションから <u>挿入 ブックマークの挿入</u>」を選択するか、右クシックにより表示されるコンテキストメニューから ブックマークを挿入 」を選択します。
- 3. ブッケマーケの挿入ダイアログが表示されます(以下のスクノーシンコナを参照)。作成されるブッケマーケが静的なテキストの場合は静的タブを、名前がXMLドキュメナやXPath 条件式が動的に取得される場合は動的タブを、静的ならびに動的ないやツから構成される場合には静的ならびに動的タブを選択してください。以下のスクノーシンコナでは、コンテキストノード以下のName子要素に対してユニーケな名前を持った動的なブッケマーケが作成されます。

ブックマークの挿入

アドレス(E):

動的



キャンセル

上のフィールドを使って動的 (XPath) アドレスを入力してください。ボタンを使ってスキーマ コンテンツを選択してください。

OK

4. 「OK」をクリックすることでブックマークが定義されます。

静的ならびに動的

generate-id(n1:Name)

XPath を編集...

■ XSLT 変換に対して分析されていないエンティティとして扱う。

作成されたブックマークは、イパーレクから参照することができます。

旡 デザイノコで指定された場所にブックマークは作成されます。繰り返し出現する要素内部が指定された場合、繰り返し出現する各 インスタンス要素内部にブックマークが作成されます。静的な名前が与えられた場合、これらブックマーク全ては同じ名前が与えられ ます。そのオータイ繰トル返し出現する要素にコネナしては、例えばコンテキストノード(ブンクマーークが作成されナンノード)の子要素にあるコンテ ンツを選択するような動的な名前を指定することが望まれます。動的な方法により選択される複数のインスタンスノードに同じコンテ ンンか含まれる場合、generated-id() 関数により生成される名前により、ブックマーク名の一意性を保証することができます(以下 のスクノーンショナを参照)。ブックマークの参照を行ういは、同じID をハイパーレンのhref 値として生成することができます。こ の場合、generate-id() 関数の前にフラグメト識別子の#を使用するようしてください。XPath 条件式はconcat('#'. generate-id(nodeXXX)) といます。

ブックマークの修正

作成されたジックマークの名前は、ブックマークの編集ダイアログにて修正することができます。以下の操作によりダイアログヘアクセスすることがで きます:

- 1. デザインにあるブックマークを選択します。2. プロパティサイドバーのブックマークグループにあるブックマーク名プロパティの編集ボタンをクリックします(以下のスクリーンショットを参照)。上記のブックマークの挿入ダイアログと同じ機能を持ったブックマークの編集ダイアログが表示されます(上のスクリーンショットを参照)。



3. ブックマークの編集ダイアログにて静的、動的、または静的ならびに動的タブにてブックマークの名前を編 集してください。

ブックマークの削除

ブックマークを削除するコよデザイン上のブックマークが選択された状態で、Delete キーを押下してくたさい。

■ 関連項目

- ハイパーレンが定義:ブックマークへのハイパーリンクへの作成方法について記述されます。
- 挿入 ブックマーク: ブックマークの挿入について記述されます。

7.6.7.2 ハイパーリンクの定義

/ イパーレクはテキストやイメージという: SPS コンポーネントの周りに作成されます。 / イパーレクのターゲットには(i) SPS デザイン内部 にあるブックマーク、 まりは(ii) ウェブページ ヤ電子 メールのメッセージという・外部 リノースを使用することができます。 このセグションでも イパー リンクのコンテンツ テキスト ヤイメージなど について、 そして イパー レクのターゲットについて 説明します。

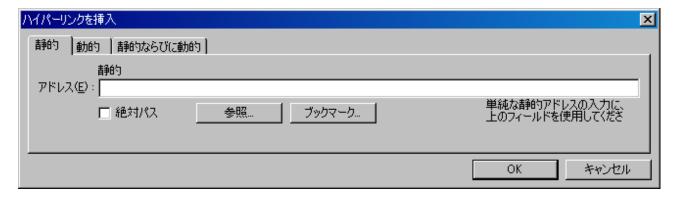
ノイパーレクの作成

以下の操作により、イパールンケを作成することができます:

- (静的または動的な)テキスト、ノード、イメージ、条件テンプレート、自動計算、コンテンツやノードのブロックの周りで:入力フィールドやコンボボックスといったデータ入力デバイスの周りにハイパーリンクを作成することはできませんが、データ入力デバイスを含んでいるノードや条件テンプレートの周りに作成することはできます。囲まれた対象が、リンクのコンテンツとなります。コンポーネントの周りにハイパーリンクを作成するには、そのコンポーネントが選択されている状態で、メニューオプションから「タグで囲む | ハイパーリンク」を選択してください。
- メニューオプションの「挿入 | ハイパーリンクを挿入」によりハイパーリンクを挿入: 新たに作成されたハイパーリンクのタグ内部に、リンクのコンテンツを追加する必要があります。

ハイパーレクのターゲルを定義

タグで囲む | ハイパーリンク まけま 挿入 | ハイパーリンクによりアクセスすることができる イパーレンの挿入ダイアログでは イパーレンの ターゲルを作成することができます(以下のスケーシンコルを参照)。



リンクのターゲトは以下のどれかけなります:

- 同一 SPS デザインにある <u>ブックマーク</u>: ターゲナ URI はフラグメント識別子とむます。
- ブックマークのアンカーに対して動的に作成されたもの:これらのURI もプラグメント識別子となります。
- <u>外部リノース</u>: URI は(直接入力された)静的なもの (XMLドキュメント内のノードから得られる)動的なもの 静的ならびに動 的ない ペーツから構成されるもの まだまか ペースされていない エンティティのどれかどなります。

以下ではれらのターゲナを定義する方法について説明します。ハイパーレクの挿入/編集ダイアログにてURIを定義して、「OK」をクリッグして完了します。

ブックマークへのノンク作成

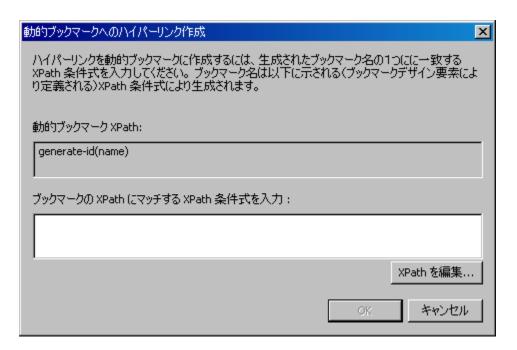
以下の操作によげックマークへのレクを作成することができます:

1. ハイパーリンクの挿入ダイアログにある静的タブにて**ブックマーク**ボタンをクリックします。この操作によりドキュメント内のブックマークを選択するダイアログが表示されます(以下のスクリーンショットを参照)。以下に示されるスクリーンショットには、静的なブックマークが1つ、動的なブックマークが1つ表示されています。



2. 静的なブックマークをターゲット URI として選択するには、静的なブックマークをダブルクリックして「OK」をクリックします。動的なブックマークをダブルクリックした場合、選択された動的なブックマークにマッチする XPath 条件式を入力するように促されます(以下のスクリーンショットを参照)。

288 高度な機能 目次、参照、ブックマーク



<u>動的なブックマーク</u>の実体は、(それ自身がブックマークの名前でない場合)ブックマークの名前を生成する XPath 条件式となります。動的ブックマークへのハイパーリンク作成ダイアログには動的なブックマークの XPath 条件式が表示されており、ターゲットとなるブックマークにマッチする名前を生成する XPath 条件式 を構築することができます。「OK」をクリックして完了します。

動的に作成されたID ブックマークへのレンク作成

ブックマークコよ 動的に生成されたID アンカーを含めることができます。このようなブックマークへのレンを作成する場合、アンカーはランタイムで動的に生成されるから、このようなアンカーは対して ハイパーレンク のhref 値をどのようにセルするかが問題ではます。動的に生成されるアンカーは対しては generate-id() 関数を、ハイパーレンクのhref 値の内部で使用することではます。変換が一度だけ行われた場合、generate-id() 関数が同一のノードに対して評価されると、常に同一のID が生成されます。これにより、ブックマークと イパーレングにて 生成されたID が同一のどのどなります。

以下の2点に留意する必要があます:

- generate-id() 関数は XPath 条件式として評価する必要があるため、ハイパーリンクのターゲットをセットするにはハイパーリンクの挿入ダイアログにある動的タブを選択する必要があります。
- href 属性で評価される値は(フラグメント識別子を表す) # からはじめる必要があります。従って XPath 条件式は concat('#', generate-id(nodeXXXX)) のような形式となります。静的ならびに動的タブにて、# をアドレスの静的パートに入力し、generate-id(nodeXXX) を動的パートに入力することもできます。

外部リソースへのリンク作成

以下の操作により、外部リソースを選択するためのURI を構築することができます:

- ハイパーリンクの挿入タブにある静的タブにて URI を直接入力する: 例えば静的タブのアドレス入力フィールドに対して Altova ウェブページ (http://www.altova.com/) へのリンクを直接入力することができます。
- ハイパーリンクの挿入ダイアログにある動的タブの XML ドキュメントソースにてノードを選択する: XML ソース内部にあるノードから、(i) ターゲットとなる URI、または (ii) 目的の URI を値として含んでいる パースされていないエンティティ 名のどれかを表すテキスト文字列を与えることづけます。例えば Altova ウェブサイトのアドレスを、ノード内にあるテキスト文字列として含めることができます。
- ハイパーリンクの挿入ダイアログにある静的ならびに動的タブにて、静的ならびに動的なパーツから構成される URI を構築する: この機能は静的な(例えばプロトコルを表す) 接頭語や(例えばドメイン名を表す) 接尾語を追加する際に使用することができます。例えば mailto: という静的なパーツと //Contact/@email ノ

ードのコンテンツから得られた文字列を組み合わせることにより、電子メールアドレスを構築します(以下 のスクリーンショットでは、//Contact/@email ノードのコンテンツプレースホルダへのリンクが作成されてお り、自分自身のノードを表す self::node() の短縮形 が使用されています)。



パースされていないエンティティを使用する方法については、パースされていないエンティティ URI のセグションを参照ください。

ノイパーリングプロパティの編集

ハイパーレクを編集するコカイパーレクの開始まけは終了タグを右クトグして、コンテキストメニューからURIの編集を選択します。ハ イパーリンクの編集ダイアログが表示されます(上のアグリーンショナを参照)。プロパティウィンドウのソイパーリンクグループ内にあるURLプロ ノティからも、ノイノトーレンクの編集ダイアログへアクセスすることができます。

ノイパーノンクの削除

ハイパーレンを削除するコよ ハイパーレンの開始まけお終了タグをクトックしてハイパーレンを選択し、Delete キーを押下します。ハイパ 一上ノクとそのエンテンツが削除されます。

□ 関連項目

- ブックマークの挿入
- パースされていないエンティティURI
- 挿入 ハイパーレク

8 表示方法と出力

SPS デザインでは、同一のスタイル機能が立か。ネトレップして定義されます。これらのスタイル情報は、それぞれの出力にて対応するスタイルマープアップへ変換されます。(Enterprise Edition のAuthentic View HTML、RTF、PDF およびWord 2007+、Professional Edition の Authentic View、Basic Edition のHTML とRTF)

メモ (HTML 以外の)RTF や PDF、Word 2007+ といった印刷媒体をベースにした出力フォーマットは、Basic Edition の StyleVision にてサポートされません。そのため、(ページのヘッダーやフッターを追加するといった)出力オプションは Basic Edition にて無効にされています。これらの機能を使用するには、(RTF 出力機能を含む)Professional Edition のライセンス、または(RTF、PDF、Word 2007+ の出力機能を含む)Enterprise Edition のライセンスをお求めください。詳細については Altova Web サイト エディンコンの比較ペーンを参照ください。

SPS コンポーネントのスタイル

SPS コンポーネントの全スタイルコは CSS2 の構文が使用されます。スタイルお外部のスタイルシートにて、SPS 全体に対してグロー・シーに、そして特定のエンポーネントに対してローカルに定義することができます。 CSS2 のカスケード順位が SPS に対して適用されることで、スタイルのデザインにおける柔軟性が飛躍的に向上します。 CSS スタイルの使用方法については、このセグションのサブセグションである CSS スタイルと作業にて記述されます。

スタイルプロ、ティの値は、スタイルませまプロ、ティサイド、ドーニて直接入力することができるまか、XPath 式を介してセナすることもできます。XPath 条件式を使用することで、())プロ、ティの値をXML ファイルから取得し、(ii)XPath 条件式で記述された条件に従ってかけた。でプロ、ティの値を割り当てることができるようになります。

その他にも、SPS デザインでは特定のHTML 要素をSPS エパーネトのマーケアプとして使用することができます。これら 定義済みの フォーマナ はHTML 出力に渡されます。このようなマーケアプは SPS エパーネトのスタイルとしても使用することができます。CSS スタイルが定義済みのフォーマトに対して適用されると、CSS スタイルが定義済みフォーマトのスタイルよけ高い優先度で処理されます。定義済みのフォーマトについては、このセグションのサブセグションである 定義済みのフォーマナ にて記述されます。

- **関連項目**
- 使用方法の概要
- デザインサイドバー

8.1 定義済みのフォーマット

StyleVision では、HTML 要素に対応する数々の定義済みフォーマナを使用することができます(以下のスクレーンショナを参照)。定義済みのフォーマナをデザイン内部にあるコンポーネントに対して適用すると、選択された HTML 上の意味を持つようにコンポーネントがマーケアプされます。この操作により2つの効果が期待できます:

- 選択された定義済みのフォーマットに対して、HTML 要素固有のフォーマットが適用されます。
- コンポーネントが段落というコンポーネントタイプに含まれ、コンポーネントタイプに対して ローカルスタイルを定義する ことが可能しています。

定義済みのフォーマトを割り当てる

メニューオプションの「挿入|特殊な段落を挿入」から目的のフォーマナを選択するか、ツール・イニあるフォーマナドロップダウンノストから目的のフォーマナを選択することで、定義済みのフォーマナを割り当てることができます。



固有のスタイル

StyleVision にて使用される定義済みフォーマナロよ、以下にあるスタイルエンポーネンナのうちの1つまけは両方が含まれます:

- テキストスタイルコンポーネント
- 余白ユポーネト

例えば、定義済みの段落(p) フォーマナロは余白エンポーネナ 13 が含まれており、選択されたエンポーネナの直前と直後に水平方向の余白が挿入され、テキストのスタイルが適用されることはありません。それとは別に、見出し1(h1) フォーマナロはテキストスタイルエンポーネナと余白エンポーネナの両方が含まれます。

定義済みのフォーマットにおける以下のスタイル情報について留意してくたさい

- 定義済みフォーマトの余白コンポーネントは全種類のSPS コンポーネントに対して適用されますが、テキストスタイルは適用できる時だけ適用されます。例えば、イメージを選択して見出し1(h1)の定義済みフォーマトを適用すると余白コンポーネントが適用されるのに対して、テキストスタイルコンポーネントは適用されません。
- 定義済みフォーマナのテキストスタイルコンポーネントはデータ入力デバイスコン対して適用されません。
- 同一のエンポーネントし対して適用される定義済みフォーマナトは1つだけです。
- 整形済みテキスト(pre) 定義済みフォーマナでは、HTML のpre タブを適用した時と同様のフォーマナ結果を得ることができます。テキスト内にある改行ならびに空白スペースはそのまま保持され、表示には(Courier のような)等角フォンナが使用されます。テキストにより構成される段落のように、改行を持たない行の場合、整形済みテキスト(pre) 定義済みフォーマナによりテキストの折り返しが行われることはおりません。テキストの折り返しを行う場合、整形済みテキスト(pre wrap) を使用してくたさい。

定義済みのフォーマナのための追加スタイルを定義する

定義済みのフォーマナを選択して スタイルサイドバーからローカルスタイルを適用することで、 固有のスタイルとおりの定義を指定することができます。

292 表示方法と出力 定義済みのフォーマット

- 関連項目
- CSS スタイルをローカルに定義する

8.2 出力のエスケープ

テキスト文字列内にある文字が文字参照まけはエンティティ参照として記述されている場合、その文字がエスケープされた状態といます。参照は(文字参照がエンティティ参照が13関わらず)アンパサンド(&)文字で始まりとコロン(;)にて終わります。例えば、

- A という文字に対する16進数(または Unicode)の文字参照は A となります。
- A という文字に対する10進数の文字参照は A となります。
- & という文字に対する HTML(ならびに XML)エンティティ参照は & amp; となります。
- & という文字に対する16進数(または Unicode)の文字参照は & となります。
- & という文字に対する10進数の文字参照は & となります。
- < という文字に対する HTML(ならびに XML)エンティティ参照は < となります。

出力のエスケープ

出力のエスケープとは入力にてエスケープされた文字を出力にて表示することを指し、出力にて文字参照ませばエンティティ参照とて表示される文字のことを出力エスケープされた文字といます。入力にてエスケープされた文字だが出力エスケープできるとう点に注意してください、以下のテーブルにある例を参照ください。SPSでは以下のものに対して出力エスケープを有効にすることができます。

- 静的テキストのフラグメント
- contents ルースかり
- 自動計算

プロ・ティのテキストグループにあるdisable-output-escaping 属性にて設定を行うことができます。プロ・ティのデフォル値ばいらいになっており、出力のエスケープが行われることは無く、入力にてエスケープされた文字はデフォルでエスケープされます(以下のテーブルにある例を参照ください)。

以下の操作により出力のエスケープを無効化することができます:

- 1. 出力のエスケープの無効化を行う(i) 静的テキスト、または(ii) 静的テキストのフラグメント、(iii) コンテンツプレースホルダ または(iv) 自動計算を選択します。
- ースホルダ、または(iv) 自動計算を選択します。
 2. プロパティサイドバーにて、テキストアイテムのテキストグループを選択し、disable-output-escaping 属性の値として表示される個々の出力または全ての出力を選択します。選択可能な値を以下に示します:
- HTML に: HTML 出力に対して disable-output-escaping の値が「はい」になります。
- Authentic に: Authentic 出力に対して disable-output-escaping の値が「はい」になります。Authentic ビューに対する出力エスケープを無効にする機能は、Enterprise エディションの Authentic ビュー(つまり Enterprise エディションのStyleVision、Authentic Desktop、Authentic Browser、XMLSpy)でしか使用できない点に注意してください。.
- 全てに: 全ての出力に対して disable-output-escaping の値が「はい」になります。

特定の出力(例えばHTML 出力)に対して出力エスケープが無効けなっている場合、選択されたテキストはその出力フォーマ・トにてエスケープされず、その他の出力フォーマ・トにてエスケープされます。

出力エスケープが無効/有効になっている状態で、テキストがどのように表示されるかを以下のテーブルに示します。

静的テキスト	disable-output-escaping	出力テキスト
&	ાપત્ર	&
&	はい	&
&	ાપત્ર	&
&	はい	&

294 表示方法と出力 出力のエスケープ

<	ાપત્ર	<
<	はい	<
A ;	ાપત્ર	A ;
A ;	はい	А
<	ાપત્ર	<
<	はい	<
<	はい	<
&<	はい	&<

メモ disable-output-escaping は、Enterprise エディションの Authentic View でのみサポートされます。

出力フォーマナごと出力エスケープを無効化する

出力エスケープを無効化したとしても、ある出力にとって意味のあるテキストが別の出力では意味を持たない場合も考えられます。例えば、(ノイライトされた)エスケープ文字が存在する、以下の入力テキストを考えてみましょう:

This text is bold.

出力エスケープが無効化されている場合、出力は以下のようごなります:

This text is bold.

HTML 出力に対して出力エスケープが無効化されている状態でこの出力がブランザーにて閲覧されると、文字列のマークアップがHTML ブラウザーによ処理され、太字でテキストが表示されるようしています:

This text is bold.

しかし、その他の出力フォーマナ(例えばPDF)で同じ文字列が出力されても、HTML 出力では意味のあるマーケアップが、その他のフォーマナでも意味を持つとは限りません。上記にある例の場合、(出力エスケープを無効にすることで得られる)エスケープされて、ないでキストは、PDF フォーマナで以下のように表示されます:

This text is bold.

上記の例で見られるように、出力エスケープを無効にすることで得られる出力テキストは、ある出力フォーマルでコードとして読み取られる一方で、その他の出力では通常のテキストとして読み取られます。disable-output-escaping プロゲーを使用する際によこの違いにして常に留意する必要があります。

□ 関連項目

- デキオトとして XML コンテンツを挿入
- 自動計算
- デザインビューシンボル

値フォーマット(数値データ型のフォーマット) 8.3

値フォーマナを使用することで、数値型のXML スキーマデータ型ノード(<u>下のノスト</u>)に収められたコンテンンを、元のデータ型を使って表示と は違うフォーマナで表示することができます。例えば、xs:date データ型のノードはYYYY-MM-DD に +02:00 といたオプションのタイムゾー ンエンポーネントが付随した形式になります。値のフォーマントはHTML 出力にて表示することもできます。自動計算の結果が数値型のデー タ型の表記にマッチする場合、自動計算の結果に対して値のフォーマナを使用することもできます(*下のリスト*)。

このセクションのサブセクションでは、以下の項目について記述されます:

- 値フォーマトメカニズムの動作方法
- 値フォーマットを定義するための構文

値フォーマットにより XML ドキュメントに収められているデータの形式が変更されることはありません。妥 メモ 当な XML ドキュメントでは、そのノードのデータ型に対して適切な形式でデータが常に収められます。値 フォーマナは、出力の表示を行うために使用されます。

値フォーマナが使用できる数値型のデータ型

以下のデータ型に対して値フォーマルを使用することができます:

- xs:decimal ならびに xs:integer: xs:integer から派生した12の内蔵型
- xs:double ならびに xs:float: 値の最小値は 0.000001、最大値は 1.000.000 となります。この範囲に入らない値は (例えば 1.0E7 といった)別の形式で表示され、値のフォーマットを適用することはできません。
- xs:date xs:dateTime xs:duration
- xs:gYear, xs:gYearMonth, xs:gMonth, xs:gMonthDay, xs:gDay

Basic エディションでは Authentic View がサポートされていないため、全てのフォーマットはサポートされま 旡 せん。

- **関連項目**
- 日付のフォーマナ

値フォーマットのメカニズム 8.3.1

値フォーマナトは以下のアイテムコ対して適用することができます:

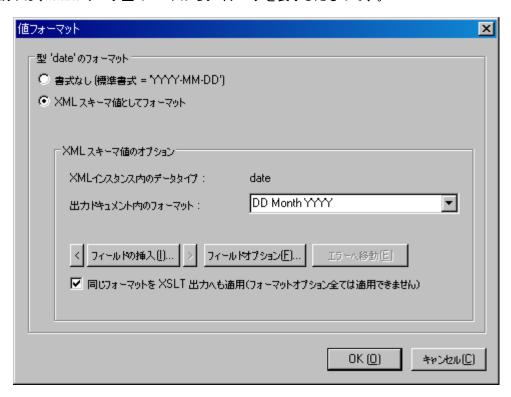
- SPS にてコンテンツならびに入力フィールドとして表示されている xs:decimal や xs:date といった 数値型 のデータ型ノード
- 数値型のデータ型フォーマットの形式に収まる自動計算の結果

値フォーマトの定義

以下の操作によりSPS に対してノードまけは自動計算を定義することができます:

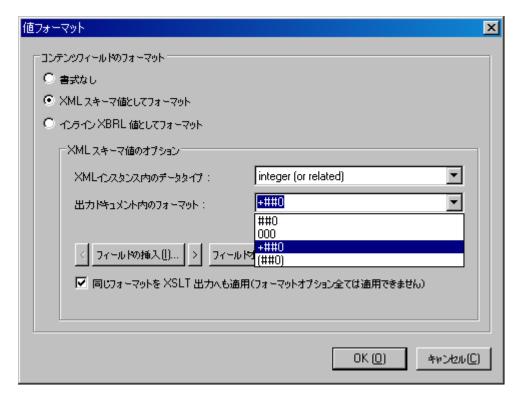
1. コンテンツプレースホルダやノードの入力フィールド、または自動計算を選択します。

2. プロパティサイドバーにて自動計算グループにある値フォーマットのアイテムを選択します。値フォーマットプロパティにある編集ボタン をクリックします。コンポーネントを右クリックして、コンテキストメニューから値フォーマットの編集を選択します。値フォーマットのダイアログが表示されます。選択されたコンポーネントがノードまたは自動計算であるかによりダイアログの表示が変化します。以下のスクリーンショットで示されるスクリーンショットは、xs:date データ型のノードからダイアログを表示したものです。



上のスクリーンショットにて示されるダイアログには、型 'date' のフォーマットが含まれており、xs:date データ型に対する標準的なフォーマットが書式なしのラジオボタンにも表示されています。その他のデータ型のノードに対して、この記述内容は変化します。

選択されたコンポーネントが自動計算の場合、ダイアログは以下の様に表示されます。



- 3. コンポーネントの値がフォーマナされたかたちで表示するのかを指定することができます。出力をフォーマナされないかたちで表示す るコよ 書式なしラジオポタンを選択してくナさい。XML スキーマ値としてフォーマットラジオポタンを選択することで、値 のフォーマルを行うことができます。値のフォーマルを行われ、場合、選択されナノードや自動計算結果のデータ型に対して標準 的なフォーマナが出力では適用されます。自動計算に対してXML スキーマ値としてフォーマットを選択した場合、(ド ロップダウノノストから期待されている自動計算結果のデータ型を選択する必要が決けます。
- 4. 値フォーマットの定義を入力します。定義は、(i)ドロップダウンリストから、そのデータ型に対して使用でき るオプションを選択する(上のスクリーンショットにある出力ドキュメント内のフォーマットを参照ください)、(ii) 入力フィールド内に定義を直接入力する、(iii) フィールドの挿入ならびにフィールドオプションを使用 することで、適切な定義を構築することができます。フォーマットのオプションについては、値フォーマト構文 を参照くたさい。

構文のエラー

構文内にエラーがある場合、以下のような振る舞いが行われます:

- 定義が赤色で表示されます。
- 赤色のエラーメッセージが入力フィールド以下に表示されます。
- 値フォーマナダイアログの「OK」ボタンか無効けなります。
- 値フォーマナダイアログのエラーへ移動がタンカ有効になります。ボタンをクトックすると、フォーマナ定義にて構文エラーが発生した箇 所にカーノルが配置されます。

データとデータ型フォーマナのミスマッチ

XML ノードに入力されケーデータがそのノードのデータ型の形式にマッチしない、まりは自動計算の結果が期待されケーデータ型の形式にマッチし ない場合、フォーマナルは未定義となり、正しいかたちで出力で表示されません。

値フォーマルを出力へ適用する

定義した値フォーマナは Enterprise ならびにProfessional エディションでサポートされる Authentic View に適用されます。

全てではおいものの、値フォーマトの定義によっては、HTML o出力へ適用できるものもかます。その他の出力へも値フォーマトを適用するコよ、同じフォーマトをXMLSpy 出力へ適用チェックボックスにチェックを入れてください。このオプションがチェックされていない場合、Authentic View にて値フォーマトが使用され、その他の出力ではエンポーネトのデータ型で使用される標準的なフォーマトにて値が表示されます。

- 関連項目

- 値フォーマト構文
- 日付のフォーマナ

8.3.2 値フォーマットの構文

値フォーマナの構文は以下のよう」さなります:

([prefix character/s]field[suffix character/s][{field-オプション1,field-option2,...]])+

ここで

prefix character/s とsuffix character/s は配置とプラス/マイナス記号を表示するかを指定するオプションの指定子で、

field は任意のデータ型依存のフォーマナまけまデキストとなり、

{field-option(s)} は更なるフォーマットオプノョンを定義するオプノョンの修飾子といます。

定義構文の説明

値フォーマトの定義は以下のようは構築されます:

- 値フォーマットの定義は1つ以上のフィールドから構成されます。例えば、DD Month YYYY には3つのフィールドが含まれます。
- フィールドは連結して使用することもできるほか、空白スペース、ハイフン、コンマ、コロン、ピリオド、または引用句('または")に囲まれたテキスト文字列により区切られたかたちで使用することができます。例えば、DD-Month' in the year 'YYYY という定義では、DDと Monthというフィールドが、ハイフンにより区切られ、Monthと YYYYフィールドが、引用句で囲まれたテキストにより区切られています。
- フィールドにはオプションの接頭文字や接尾文字を加えることができます(例: <+###,##0.00)。
- フィールドコお必要に応じて使用できるフィールドオプタンを含めることができます。各フィールドオプタンは角かっ工内に含め、フィールドの直後に配置する必要があります。複数のフィールドオプタンを使用する場合、エンマ(パー) にて分離する必要があります。例えば、DD Month[uc,ro] YYYY という定義では、uc とro が Month フィールドレジオるフィールドオプタンとかます。

例

xs:decimal データ型に対する値フォーマナの例を以下に示します:

"\$"(##0.00)

出力は以下のようごなります:

\$ 25.00

\$ 25.42

\$267.56

xs:date データ型に対する値フォーマナの例を以下に示します:

DD Month{uc,ro} YYYY

こで、 uc ならびにro は、それぞれ Month フィールドを大文字の読み取り専用にするためのフィールドオブンョンとなります。

出力は以下のようなります:

24 SEPTEMBER 2003

フィールドタイプ

フィールドタイプは、データのエンポーネントと、そのエンポーネントがどのように表示されるのかを表します。フィールドタイプにおける固有のフォー マナは、接頭部ならびは接尾部にある修飾子やフィールドオプロンにより修正することができます。以下のテーブルコは利用可能なフィールド タイプが表示されます。値フォーマナダイアログのドロップダウンメニューコよ、特定の型に使用されるフィールドと、フィールドだけの値フォーマナ 定義が含まれます。定義を選択し、接頭部の修飾子、接尾部の修飾子、フィールドオプションを必要に応じて加えることはよいます。

フィールドタイプ	説明
#	指定された場所に対応する数値が無い場合、空白しないます
0	指定された場所に対応する数値がは場合、0 になります
	小数点
,	数値グループのセパレーター
Υ	年
у	年(base = 1900)、下記のメモを参照(ナシン)
MM	月、2文字分の長さか必要
DD	日、2文字分の長さか必要、
W	週の番号
d	曜日の番号(1 から7)
i	日付の番号(1 から366)
hh	時(0 から23)、2文字分の長さか必要
НН	時(1 から12)、2文字分の長さか必要
mm	分、2文字分の長さか必要
ss	秒、2文字分の長さが必要
AM	MA ませま MA
am	am おはpm
AD	AD おはBC
ad	ad ませまbc
CE	CE おはBCE
се	ce おはbce

フィールドタイプ	説明
Weekday	曜日(Sunday, Monday)
WEEKDAY	曜日(SUNDAY, MONDAY)
weekday	曜日(sunday, monday)
Wkd	曜日(Sun, Mon)
WKD	曜日(SUN, MON)
wkd	曜日(sun, mon)
Month	月(January, February)
MONTH	月(JANUARY, FEBRUARY)
month	月(january, february)
Mon	月(Jan, Feb)
MON	月(JAN, FEB)
mon	月(jan, feb)

フィールドの長さと入力値の長さ、関するメモ

データエンポーネントの長さ、関する以下の点に留意してください。

データフィールドの長さ: MM、DD、HH、hh、mm、ss といったフィールドが使用された場合、定義の中に2文字分の長さを与える必要があります。y または Y フィールドが使用された場合、定義で使用されている y または Y 文字の数により、出力の長さが決定されます。例えば、YYY という文字を指定した場合、2006 に対応した出力の値は 006になり、YYYYYY という定義の場合、002006 となります。下記にある base 年に関するメモも参照ください。

フィールドの長さを延長:* (アスタリスク)を使うことで、意味を持たない数値フィールド(integer や decimal など)の長さを延長することができます。decimal の場合、小数点の前と後の両方で使用することができます。例えば、*0.00* という値フォーマットでは、0 により指定された場所だけではなく、整数部部分と小数点以下にある任意の場所で、数値が空の時には 0 が使用されるようになります。

メモ フィールドによりテキストのレンダリングが行われない場合、Windows の地域オプションが原因である場合もあります。例えば Windows の地域言語設定がドイツ語になっている場合、AM/PM フィールドに対して空の文字列が返されます。

接頭部ならびは接尾部の修飾子

接頭部ならびに接尾部の修飾子により、テキスト配置とフィールドのドンティブ/ネオディブ表示を修正することができます。接頭ならびに接尾で使用することのできる修飾子を以下のテーブルニ示します。

接頭	接尾	説明
<		左揃えを行い、テキストに対してデフォルの振る舞いを行います。数値に対しては、 デフォルで右揃えどなっています。
>		数値に対する右揃えを行います。
?		負の数値に対してマイナスのシンドルを付け加えます。数値に対するデフォルトのオプションコなります。

</td <td></td> <td>負の数値に対してマイナス記号を左寄せして表示します。マイナス記号の後口は左 寄せされた数値が続きます。</td>		負の数値に対してマイナス記号を左寄せして表示します。マイナス記号の後口は左 寄せされた数値が続きます。
		負の数値に対してマイナス記号を左寄せして表示します。数値は右寄せにおよ 示されます。
-	-	負の数値に対して、マイナス記号が隣り合わせで表示され、そうでむ、場合は空白が表示されます。接頭部にて指定されると数値の前にマイナス記号が、接尾部の場合は数値の後に記号が表示されます。
<-	>-	負の数値に対してマイナス記号が表示され、それ以外の場合には空白となります。 数値と記号は隣り合って表示され、接頭部の場合は数値の前にマイナス記号が 接尾部の場合は数値の後に記号が表示されます。
<->		負の数値に対してマイナス記号が表示されます。数値は右寄せにい表示されます。
+	+	常にプラスまけよマイナス記号を数値の隣に表示します。接頭部では数値の前に、 接尾部では数値の後に表示されます。
<+	>+	常にプラスまけはマイナス記号を数値の隣に表示します。接頭部は左寄せ、接尾部は右寄せで表示が行われます。
<+>		常にプラスまだはマイナス記号を表示します。記号は左寄せで表示され、数値は右 寄せで表示されます。
()	負の数値の場合かっこか表示され、それ以外の場合は空白か数値の隣に表示されます。
<(負の数値の場合からたが表示され、それ以外の場合は空白が数値の隣に左寄せで表示されます。
<(>		負の数値の場合かごが表示され、それ以外の場合は空白が表示されます。左かっこが とまる これ これ これ これ これ これ これ これ これ ままる これ これ まず これ とり できる これ これ これ まず これ とり できる これ これ これ これ とり これ
[]	負の数値の場合かっこか表示され、それ以外の場合は空白か数値の隣に表示されます。
*	*	(接頭部にて)左側に対して、かつ(接尾部にて)右側に対して数値の延長を行います。
_	_	空白スペース。
۸	۸	オプロンで定義された文字の挿入。
	th	英語(1st、2nd、3rd、4th など)。
	TH	英語はおる字数(1ST、2ND、3RD、4TH など)。

フィールドオプション

フィール・オプションにより、高度な修正をフィール・コン対して行うことができます。以下のオプションを使用することができます:

オプション	説明
uc	大文字に変換。
lc	小文字に変換。

left	左寄せ。
right	右寄せ。
ro	(XML にある)値の読み取りだけで、編集を許可しない。
edit	編集可能なフィールド(デフォルで有効)。
dec= <char></char>	小数点の文字を指定(デンオルではしけ)。
sep= <char></char>	数値の区切りに使用される文字を指定(デンナルではエンマ)。
fill= <char></char>	塗りつぶしつ使用する文字を指定します。
base= <year></year>	年フィーナドのbase 年を指定(以下のメモを参照)。
pos	プラス記号だき表示、負の数値の入力を行うこともできます。

フィールドオプションを使用することで、小数点にコンマを使用して、数値の区切りによりオドを使用するという欧州式の数値フォーマルを生成 することができます(例: 123.456,78)。

上の値を生成するために使用される値フォーマナは以下のようにはいます: ###,###.##(dec=,,sep=.)。

フィールドロは英国式のフォーマナが指定されたままけなっている一方で、dec ならびに sep オプションにより数値の小数点と区切り記号が W indows OS の地域設定に従い指定されている点に注目してください(コントロールペネル) すべてのアイテム 地域 | フォーマッ **ト**)。

□ 関連項目

- 値フォーマトのメカニズム
- 日付のフォーマット

8.4 CSS スタイルを使用する

SPS ドキュメトのスタイルはCSS ルールによりデザインされます。スタイルルールは以下のように指定することができます:

- <u>外部 CSS スタイルシート</u>にて: <u>デザイン概要</u> サイドバーと <u>スタイル・ポジトリ</u> サイドバーから外部 CSS スタイルシートを追加することができます。
- グロー ジレスタイル にて: SPS の内部ならびにその開始部分で定義が行われます(HTML 出力では、これらグロー ジレスタイルが head 要素内にある style 子要素にて定義されます)。グローバルスタイルはスタイルが・ナリサイドバーにで定義 されます。
- ドキュメントにある個々のコンポーネントに対してローカルで: HTML 出力では これらのルールが個々のHTML 要素内にある style 属性にて定義されます。ローカルのスタイルルールは スタイルサイド・インで定義されます。

上記のメカニズムことリスタイルを作成する方法の詳細については、このセグション以下にあるサブセグションにて記述されます。

用語解説

おは

CSS スタイルシートは1つ以上のスタイルルールにお構成されます。ルールおり下のように表示されます:

```
H1 { color: blue }
H1 { color: blue;
    margin-top: 16px; }
```

各ルールコはセレクター(上の例におった H1)と宣言(color: blue)か含まれます。宣言はプロ、ティ(例えばcolor)と値(blue)か含まれます。各プロ、ティと値のペアのことをスタイル定義と言います。StyleVisionでは、CSS スタイルをスタイルサイドバー(ローカルスタイル)ならびにスタイル・ポネリサイドバー(グロー・ドルスタイル)にて定義されます。

カスケードの順序

CSS ではカスケード順序が適用されます。これにより、評価されるルールの順序は以下のように決定されます:

- 1. 宣言場所:外部スタイルシートの優先度はブロー・シレスタイルより上低く、グロー・シレスタイルの優先度はコーカルスタイルより上低くなります。外部スタイルシートはインポートされてものとして扱われ、後からインポートされてスタイルシートがより高い優先度を持つようしています。
- 2. 詳細性: 同じ要素に対して2つのルールが適用された場合、より詳細なセレクターがより高い優先度を持つようになります。
- 3. 順序: 2つのルールが同じ場所から同じ詳細性を持っている場合、スタイルシートの後の方に出現するルールがより高い優先度を持つようになります。スタイルシートがインポートされる場合、インポート元スタイルシートの前に来るものとして処理されます。

モジュールSPS における CSS スタイル

SPS モジュールが別のSPSへ追加された場合、参照元のSPS にある CSS スタイルが、追加されたモジュール内のスタイルよ用高い優先度を持つようになります。複数のモジュールが追加された場合、これらモジュール内にある CSS スタイルのうち、より下部にあるスタイルが高い優先度を持つようによります。モジュールSPS に関する情報についてはモジュールSPS のセグションを参照がさい。

CSS サポートin Internet Explorer

IE 6.0 以前のInternet Explorer (IE) では、IE 6.0 以降とは異なる CSS ルールの解釈が行われます。従って、デザイナーにとって、 どのドージョンのIE に対してデザインを行うのかが重要においます。6.0 以降のIE では新旧両方の解釈を行うことができ、新しいドージョン にて過去の解釈を行うこともできます。IE 6.0 以降で使用される解釈方法は HTML ドキュメ・トコード 内部で指定することができます。 SPS では、指定により HTML 出力ドキュメントのスタイルを、Internet Explorer の古い新し、解釈によりデザインすることができま す。選択された解釈方法に従うかたちでCSS スタイルを定義することができます。詳細については プロッティ CSS サポートを参照くたさい。

メモ CSS の仕様に関する詳細については、http://www.w3.org/TR/REC-CSS2/ を参照なさい。

■ 関連項目

- スタイルポネリサイドバー
- スタイルサイドバー
- <u>CSS サポート</u>
- <u>モジュー</u>ルSPS

8.4.1 外部スタイルシート

このセクションではStyleVision のGUI から外部 CSS スタイルシートを管理する方法について説明します。以下にある項目を説明します:

- 外部 CSS スタイルシートを SPS へ追加する
- 外部 CSS スタイルシートにおけるコンテンソの確認と修正を行う
- 優先順位を変更する
- CSS スタイルシートセナと 単一のCSS スタイルシートを切り替える

外部 CSS スタイルシートは、スタイルポネリサイドバーとデザク概要サイドバーとう2つのサイドバーに管理することができます。外部スタイルシートが両方のサイドバーに表示されている場合、あるサイドバーに行われた変更は、自動的に他のサイドバーにも反映されます。

外部 CSS スタイルシート SPS に追加する

以下の操作により外部 CSS スタイルシートを SPS へ割り当てる事ができます:

1. デザインビューにてスタイルリポジトリウィンドウにある外部アイテムを選択します(以下のスクリーンショットを参照)。



- 2. スタイルリポジトリツールバーの左側にある追加ボタンをクリックします(上のスクリーンショットを参照)
- 3. ファイルを開くダイアログが表示されるので、目的の CSS ファイルを選択し開くボタンをクリックします。 CSS ファイルが外部アイテム以下に追加されます(以下にある表示を参照ください)。
- 4. ステップ1から3を繰り返すことで、別の外部 CSS スタイルシートを追加することができます。それらの CSS スタイルシートは外部アイテムの、それまでに追加された外部 CSS スタイルシート以下に追加されます。

メモ デザン概要 サイドバーかきがい CSS スタイルシートを追加することができます。

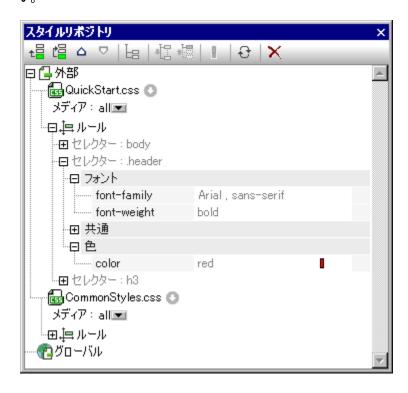
外部 CSS スタイルシートの確認と修正

外部 CSS スタイルシートのソノーは以下のように構成されます(以下にあるスクノーンショナも参照ください):

```
    CSS-1.css (マウスオーバーによりファイルの位置が表示されます)
    Media (スタイルルボントリウィンドウェンできます)
    Rules (そのまま編集することはできず、CSS ファイル内部で編集を行う必要があります)
    Selector-1
    Property-1
    Property-N
    Selector-N
    CSS-N.css
```

スタイルルドネリウム・ドウでは、そのスタイルシートが適用されるメディアを編集することができます。アイテムの右側にある矢印アイエンをグックして、ドロップダウンノストから目的のメディアを選択してください。外部 CSS スタイルシートに定義されたルールがスタイルルドネリウム・ドウェ表示されますが、編集することはできません。各アイテムの左にある + および - シンボルをグックすることで、ツノー内部にあるセンクターアイテムを展開、縮退することができます(以下のスクノー・シンョナを参照)。

外部スタイルシートを削除するコよスタイルシートが選択された状態でスタイルルドントリソール・コンあるリセットボタンをクリッグしてくたさい。



外部 CSS スタイルシートの優先順位を変更する

スタイルノドネリウンドウェて割り当てられた外部 CSS スタイルシートは、@import 命令により HTML 出力ファイルへんポートされます。 HTML ファイルでは以下のような記述がなされます:

<html> <head>

HTML ファイルニて表示されるファイルの順序は、スタイルルドネリの外部ソー以下ならパニデザイン概要サイドバーのCSS ファイルソー以下に表示されている順序に従います。スタイルルドネリニて CSS ファイルの順序を変更するコま、優先順位を変更するスタイルシートを選択し、スタイルルドネリソール・コープを上へ がおりまけます。 マタイルルドネリソール・コープを上へ がおりまけます。 マタイルシートの位置 (優先順位)を変更してください。デザイン概要サイドバーでは CSS スタイルシートの編集ボタンをクリックして、必要に応じて上へまけます。 ヘコマンドを選択することができます。

重要:最も下に表示されているスタイルシートが最も高いインポート優先度を持ち、表示されている順序が上が沿つれてスタイルシートのインポート優先度が下かることに注意してください。上に示されている例のインポート優先度は(1) ExternalCSS-3.css、(2) ExternalCSS-2.css、(3) ExternalCSS-1.css の順で高くなります。異なるスタイルシートにて定義された2つのCSS ルールが同じセンターを持っている場合、より高いインポート優先順位のスタイルシートにあるルールが適用されます。

すべてのCSS ファイルと単一のCSS ファイルを切り替える

(i) 全ての CSS ファイルにあるルールをカスケードルールの優先順位に従い適用する他にも、(ii) 選択された CSS ファイルのルールだけを適用することができます。このオプションはデザイン概要サイドバーから選択することができます(以下のスクリーンショットを参照)。 表示されている CSS ファイルの編集ボタンをクリックして、全ての CSS ファイルからスタイルを混合コマンド、または XPath で CSS ファイルを選択コマンドを選択してください。このオプションはスタイルリポジトリウィンドウでも使用することができます。



XPath で CSS ファイルを選択コマンドをクリックした場合、XPath 条件式を入力するためのダイアログが表示されます(以下のスクリーンショットを参照)。ダイアログ上部のペインに表示されているとおりに、SPS 内にある CSS ファイルのどれかを評価する XPath 条件式を入力する必要があります。ファイル名を文字列として入力する場合、その他全ての XPath 条件式と同様に、文字列をシングルクォートにより囲まなければならない点に注意してください。

• 全てのCSS ファイルにあるスタイルルールが使用される場合: Authentic ならびに HTML 出力の場合、選択された CSS ファイルにある全てのルールが、ルールをサポートする全てのデザインコンポーネントに対して適用されます。 では、非クラスセレクタールールだけが、優先順位に従うかたちで全ての CSS ファイルから適用されます。

□ 関連項目

- スタイルポネリ
- グロー・ジレスタイル
- ローカルスタイル
- CSS ザポート

グローバルスタイル 8.4.2

グロー シレスタイルはSPS デザイレ全体に対して適用されます。 スタイルJポネリはスタイルJポネリのグロー シレ以下に表示されます。 こ れらのスタイルは、CSS ルールとて HTML 出力へ渡されます。HTML ドキュメトでは これらの CSS ルールが/html/head/style 要素内に記述されます。

スタイルノポントリでは、セレクターとそのセレクターコ対応する CSS プロノティから構成される CSS カジローノ シルスタイルとなります。以下の2 つのステップによりスタイルを作成することができます:

- 新たなスタイルを追加して、それに対する CSS セレクターを宣言する
- ◆ そのセレクターに対する CSS プロパティを定義する

サポートされるセレクター

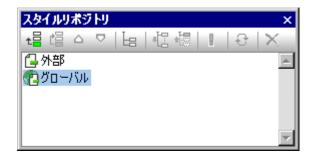
次のセクターがサポートされます:

- ユニバーサルセレクター:*により示されます。
- 型セレクター: h1 といった要素名を対象にします。
- <u>属性セレクター</u>: [class=maindoc] といった属性を対象にします。
- クラスセレクター:例:.maindoc • *ID セレクター*: 例: #header

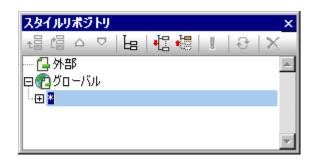
グローバルなスタイルを追加する

以下の操作によりプロー・ゾルスタイルをSPS デザインへ追加することができます:

1. デザインビューにてスタイルリポジトリウィンドウにあるグローバルアイテムを選択します(以下のスクリーン ショットを参照)。



- 2. スタイルリポジトリツールバーの左にある追加ボタンをクリックします。(全ての HTML 要素が選択される) * セレクターを伴うグローバルスタイルが、グローバルツリー内に挿入されます。新たに挿入されたグローバルスタイルのデフォルトセレクターにはユニバーサルセレクターが使用されます。
 3. セレクターをダブルクリックすることで、デフォルトのユニバーサルセレクターを変更することができます。



- 4. セレクターに対して CSS プロパティをセットします。プロパティをセットする方法については、<u>スタイルの値の設</u> 定セグションを参照(がさい)。
- 5. セレクターに対して CSS プロパティをセットします。プロパティをセットする方法については、

旡

- スタイルリポジトリウィンドウにある挿入ボタンをクリックすることで、グローバルツリーにて選択されているグローバルスタイルの前に新たなグローバルスタイルを挿入することもできます。追加ならびに挿入コマンドは、グローバルセレクターを右クリックすることでも選択することができます。
- グローバルツリー内にあるアイテムを右クリックして、「セレクターを追加 | HTML | HTML の要素名」を選択することで、HTML 要素に対するグローバルスタイルのセレクターを挿入追加することができます。

グローバルスタイルの編集と削除

スタイルセレクターとそのプロノティは、スタイルノポシトリウィンドウェで編集することができます。

- セレクターを編集するには、セレクター名をダブルクリックしてテキストフィールドにて編集を行います。
- スタイルにおけるプロパティ値の定義と編集については、スタイルの値の設定を参照ください。スタイルコお種類の表示方法が対人それらのビューと、ビューの切り替えていては定義ビューのセグションを参照ください。

グロー・ゾレスタイルを削除するコよ そのスタイルを選択して、スタイルルポントリンール・ーにあるリセットボダンをケリングしてください。

グローバルスタイルの優先順位を変更する

スタイルナドトリウィ・ドウェて割り当てられたグロー・シレスタイルは、CSS ルールとして /html/head/style 要素内に配置されます。HTML ファイルでは、これらのルールは以下のように表示されます:

```
< h + m 1 >
       <head>
              <style>
                      <!--
                                     { color:blue;
                     h1
                                       font-size:16pt;
                     h2
                                     { color:blue;
                                       font-size:14pt;
                             { color:red;}
                      .red
                      .green { color:green;}
                      .green { color:lime;}
                      -->
              </style>
       </head>
       <body/>
</html>
```

グロー ジスタイルがAuthentic View ならびにHTMLドキュメトにて表示される順序は、スタイルパネーリのグロー ジレソーに表示される順序に対応します。Authentic View ドキュメトにおける順序に対応します。Authentic View ドキュメトにおける順序に対応します。Authentic View ドキュメトにおける順序に対応した。ターには同じ、一下が選択される場合、よい下部に表示されているセンターが、よ高い優先順位を持つことがます。例えば、上記のレールがHTMLドキュメトに表示されており、HTMLドキュメト内に、1 class="green"> とう要素と、3つのグロー・ジスタイルがこの要素に一致する場合: h1 センクターと2つの.green グラスセンターを持つ場合。色 green を持つ.green センクターの後に発生し、優先順位を与えられるか、and 色 lime を持つ.green センクターのプロ・ティが適用されます。(グラス・センターはノー・ドセンターよりに高い優先順序が与えられるため、.green センクターコは、h1 センクタース対する位置に関わらず、h1 センクターよりに高い優先順位が与えられます。h1 スタイルのフォントサイズは、しかしなが、、<h1> 要素に一致する高い優先順位を有し、フォントサイズプロ・ティを有するセンクターが存在しないため、<h1> 要素に適用されます。

グロー いスタイルの優先順位を変更するコ まそのスタイルが選択された状態で、スタイルルドネリソール ・一 すると へならびに下へ ボタンを使用します。例えば .green グロー いスタイルが .red スタイルの前に移動されると .red スタイルの色のプロ ティ は .green スタイルのプロ ティよ と .green スタイルのプロ ティよ .green .green

□ 関連項目

- スタイルルポネリ
- 外部 CSS スタイルシート
- ローカルスタイル
- <u>CSS ポト</u>

8.4.3 ローカルスタイル

スタイルがローカルで定義されると、コンポーネトトコ対してスタイルが直接定義されます。ローカルのルールは、グロー・ 〝ルのスタイルルールならび トコ外部 CSS スタイルシート より 上高い 優先度を持つこと コンドます。ローカルで定義されるスタイルは CSS スタイルで、フォーマ・トツール ご 一 まっと スタイル・サイド・デーニで定義されます(グロー・ 〝レスタイルの定義と外部 CSS スタイルシートの定義は スタイルレポネーリ で行われ る点に注意してくたさい。

フォーマ・トツール・・・を使用してのローカルスタイル指定

デザイン内のコンテンンを選択し、フォーマ・トツールドーからローカルスタイルを適用することができます(以下のスクリーンショナを参照)。



div やh1、pre といた定義済みのHTML フォーマナー対して、テキストのスタイル、背景色、テキストの配置、ハイパーレクを適用することができます。詳細にストでは、フォーマナツール・のセクションを参照なされ、

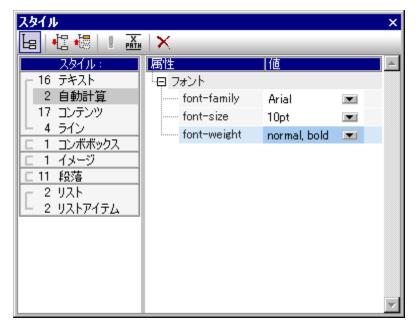
スタイルサイドバーを解してのローカルスタイル指定

以下に示されるステップにより、ローカルにおけるスタイルの定義を行うことができます:

1. スタイルを作成するコンポーネントをデザインビューにて選択します。ノードタグを除く全てのコンポーネントのスタイルを作成することができます。デザインビューにて選択されたコンポーネントが、スタイルサイドバーのスタイルペインに表示されます(以下のスクリーンショットを参照)。以下のスクリーンショットは、コンテンツコンポーネントがデザインビューにて選択された状態で、コンテンツがスタイルカラムに表示されます。



デザインビューにて選択されているコンポーネントが他のコンポーネントを含んでいる場合、選択されたコンポーネント内部にあるその他全てのコンポーネントが、コンポーネントの種類ごとにスタイルサイドバーのスタイルカラムにて表示されます。以下に示されるスクリーンショットではデザインビューにて選択されたコンポーネントに含まれているコンポーネントの種類が表示されています。各種コンポーネントの左には、選択対象内部に存在するインスタンスの数が表示されます。例えば、以下のスクリーンショットでは、16個のテキストコンポーネントと2個の自動計算コンポーネントが選択対象に含まれていることが分かります。あるコンポーネントが選択された状態で Shift キーを押下したまま別のコンポーネントを選択することで、選択された範囲内のコンポーネント全てを選択することができます。



2. デザインビューにて選択を行い、スタイルペインにてスタイルの指定を行うコンポーネントの種類を選択します。選択されたコンポーネントの種類が複数個ある場合、定義したスタイルはそれらのコンポーネント全てに対して適用されることになります。例えば、上のスクリーンショットにある16-テキストアイテムを選択した場合、定義したスタイルは16あるテキストコンポーネント全てに対して適用されることになります。このうち4個のテキストコンポーネントだけを個別に定義したい場合、目的のテキストコンポーネントを個別に選択してスタイルの定義を行う必要があります。異なるスタイル値を持ったコンポーネントがデザインビュー

にて選択されている場合、それらの値全てがスタイル定義ペインに表示されます。上のスクリーンショット では、ある自動計算に normal の font-weight を割り当てられている一方で、別の自動計算には bold fontweight が割り当てられていることが分かります。
3. スタイルペインにてスタイルを定義するコンポーネントを選択し、スタイル定義ペインにてスタイルの定義を行いま

す。定義方法については、スタイルの値の設定、セクションを参照くたさい。

□ 関連項目

- スタイルの値の設定 スタイルプロ ティの定義方法に関する情報。
- グロー ジレスタイルグロー ジレスタイルの使用に関する情報。
- 外部 CSS スタイルシート 外部 CSS スタイルシートの使用に関する情報。
- スタイルサイドバースタイルサイドバーの説明
- ◆ CSS サポート CSS サポートのレベルを調節する出力ドキュメントの設定に関する説明

スタイルの値の設定 8.4.4

スタイルサイン、一のスタイル定義ペインでは、スタイルカラムで選択された種類のエンポーネントし対してスタイルプロ、ティを定義することがで きます(以下のスクリーンショナを参照)。スタイルカラムでは、Ctrl キーやShift キーを押下しながら複数のエレポーネノトを選択することも できます。複数種類のエンポーネントが選択されている場合、スタイル定義ペイノこて定義した値が、選択されているエンポーネントの全インス タンスマ対して適用されること」なります。.



スタイルプロ/ ディグループ

スタイル定義ペイノニて編集することのできるスタイルプロ・ティは、以下のスクリーシショナで示されるように、グループごとこ管理されます。

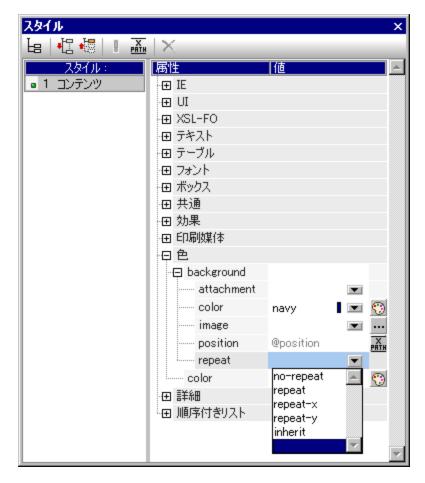


<u>値のあるものを表示、全て展開と、全て縮退 ツールド・ポタン</u>ソールド・ポタンを使用することで、プロゲーの表示を修正することができます(以下のスクノーンショナを参照)。

スタイル値を入力する

スタイルプロ ティの値(スタイル値)は、以下の方法により入力することができます:

- 値カラムに直接入力する:プロパティを選択し、値カラムをダブルクリックします。キーボードにより値を入力し、 Enter キーを押下するか、その他の場所をクリックすることで入力された値が確定されます。
- そのプロパティのコンボボックスにあるドロップダウンリストから値を選択する: コンボボックスにあるドロップダウンの矢印をクリックすると、スタイル値のオプションが表示されます。以下のスクリーンショットでは、(background-) repeat プロパティに対するオプションが表示されています。ドロップダウンリストから目的の値を選択してください。
- 値カラムの右側にあるアイコンを使用する: 2種類のアイコンが用意されており、関連するプロパティに対してのみこれらのアイコンは表示されます。((background-)color プロパティにある)カラーパレットは色の選択を行うために使用され、((background-)image プロパティにある)ダイアログアイコンからは、ファイルの参照を行うことができます。



• XPath 式により、スタイルの値を割り当てることもできます。

スタイル値の修正と削除

スタイル値が正確に入力されなかった場合、値が赤色で表示されます。 プロンディの値の挿入で説明されているとおりに値の修正を行ってください。

スタイル値を削除(つまり、スタイル値を) からいました。その他にも、プロッティの値カラムをダブルクトックして、Delete まけは Backspace キーにより値を削除した後に Enter キーを押下することもできます。

□ 関連項目

- スタイルサイドバー
- CSS スタイルをローカルに定義する
- CSS スタイルをグローバルに定義する
- 外部 CSS スタイルシート
- <u>CSS サポート</u>

8.4.5 XPath を使用したスタイルプロパティ

XPath 条件式を使用することでもスタイルを割り当てることができます。この機能により、スタイルプロ ティの値を XML データや、XPath 条 件式から与えることができます。更に XPath 2.0/3.0 のdoc() 関数を使用することで、アクセス可能な XML ドキュメトにあるノードへの アクセスを行うことが可能しています。スタイルの定義をXML データから取得するだけではなく、XML データのエンテンツ代構造により、異なる スタイルを与えることも可能しています。例えばXPath 2.0/3.0 のif...else ステートメートを使用することで、シーケンス こおける要素の位置 に従って、異なる背景色を選択することが可能になります。これにより、これらの要素がテーブルで表示される時に、偶数番号と奇数番号の背 景色がそれぞれ別の色となります(以下の例を参照)。また、ノードのコンテンソニ従うかたちでスタイル値を変更することもできます。

XPath 条件式を使用することができるスタイルプロパティ

XPath 条件式は、以下のスタイルプレーディー対して入力することができます:

- スタイルサイドバーに表示されている全てのプロパティ
- プロパティサイドバーの共通、イベント、HTML グループ内にあるプロパティ

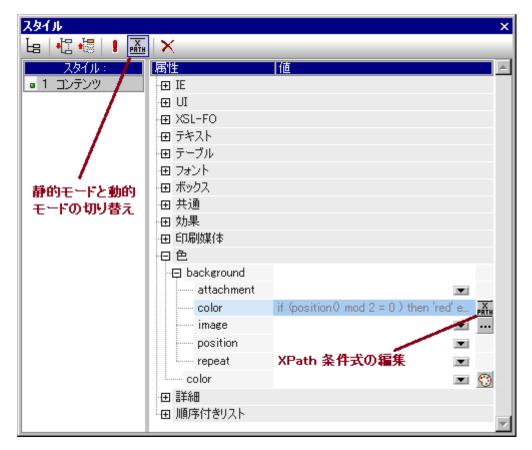
プロペティ値に対する静的モードと動的(XPath)モード

XPath 式の有効化プロ ティコよ 2つのモードが用意されています:

- 静的モード: サイドバーの値カラムにプロパティの値が直接入力されます。例えば、デザインコンポーネントの背 景色に対して red という値をサイドバーに直接入力します。
- 動的モード(または XPath モード): XPath 条件式が入力されます。条件式はランタイムにて評価され、その結果 がプロパティの値として入力されます。例えば、デザインコンポーネントの背景色に /root/colors/color1 という XPath 条件式を入力することができます。ランタイムでは、/root/colors/color/ ノードのコンテンツが背景色プロパ ティの値として使用されます。

静的モードと動的 (XPath) モードの切り替え

XPath 条件式を使用することができるプログライでは、静的モードカデフォルトで選択されています。プログライを動的(XPath)モードにカリ 替えみコよ そのプレ ティを選択してサイドバーのソール・ーにある XPath アイエンをクトゥクします(以下のスクリーシショナを参照)。



静的な値がそのプロ、ティニ対して既に入力されている場合、その値がカアされ、動的モードへの切り替えが行われます。XPath 式 の編集ダイアログ が表示されるので、そのプロ、ティで使用される XPath 条件式を入力します。「OK」をクリックすることで編集を完了します。

プロ ティマ・ルて XPath 条件式が入力されると、そのプロ ティの値カラムにXPath 条件式の編集が必力表示されます(上のアクリーシショナを参照)。このだりをクリックすることで、既に定義された XPath 条件式を編集することができます。静的モードへの切り替えを行うゴよ ツール デー ある XPath アイエンをクリックしてくたさい。定義された XPath 条件式がプロされ、静的モードへの切り替えが行われます。

メモ 静的モードと動的モードのうち、同時にアクティブにすることができるのは1つに限られるという点に注意してください。モードの切り替えを行うと、それまでに入力された値がクリアされるため、再度モードの切り替えを行なってもフィールドは空になります(過去に入力した値 / 条件式を取り戻すには、元□戻すコマナゴマンドを使用してください。SPS を編集した後□再度プロ ティを選択しても、過去□選択されたモード□でプロ ティは開かれます。

XPath 条件式の作成と編集

XPath の定義は、XPath 式 の編集ダイアログ にて作成ならびに編集することができます。2つの方法によりにのダイアログヘアクセスすることができます:

- (サイドバーのツールバーにあえる XPath アイコンをクリックすることで)静的モードから動的モードへの切り替えを行うと、XPath 式の編集ダイアログが表示されます。プロティが既は動的モードはいている時にソールドーのXPath アイコンをクリックすると、静的モードへの切り替えが行われ、XPath 条件式の編集ダイアログは表示されないことに注意してください。
- プロ ティス対して XPath 条件式が既に定義されている場合、値フィールドにある XPath 条件式の編集が文をグトックすることで XPath 式 の編集ダイアログ ヘアクセスすることができます。ダイアログコよ そのプロ ティス対して既に定義されている XPath 条件式が表示され、そのまま編集を行うことができます。

XPath 条件式の入力まけは編集を行すた後は「OK」をクリックして完了してくたさい。

XPath 式により得られる値

XPath 条件式を使用することで、(i) XML ファイルから取得されナ値をプロ ティの値として使用することができるようになり、(ii) 処理されて いるXMLドキュメトのエレテンソや構造を条件に対して使用することで、適切な値を選択することができるようにはいます。XPath 条件式 からは、以下にある2種類のカテゴリーにより値が返されます:

- XML ノードコンテンツ
 - XPath 条件式により、(i) SPS により処理されている XML ドキュメント内のノード、または (ii) アクセス可能 な任意の XML ドキュメント内にあるノードへアクセスすることができます。例えば、Format/@color という条 件式により、コンテキストノードの子要素である Format の color 属性にある値を、この XPath 条件式が定義されたプロパティの値としてセットすることができます。XPath 2.0 の doc() 関数を使用することで、その他の XML ドキュメントにあるノードへもアクセスすることができます。例えば、doc() Styles.xml')//colors/color-3 という XPath 条件式により、Style.xml という XML ドキュメントにある color-3 という要素内の値をプロパティの値として使用することができます。
- XPath 式

XML ドキュメントではなく、XPath 条件式そのものにプロパティの値を入力することもできます。例えば、テ 一ブルの行として出力される要素の背景色を、奇数番目と偶数番目により変更する例を考えてみましょ う。このような処理は if (positikon() mod 2 = 0) then 'red' else 'green' といった XPath 2.0 の条件式により実現 することができます。この条件式の戻り値は、'red' という文字列か、'green' という文字列になり、これらの 文字列(のうちのどちらか)が、この XPath 条件式が定義されたプロパティの値として使用されることにな ります。この例では文字列リテラルとしてプロパティの値が入力されています。文字列リテラルを使う以外 にも、if (position() mod 2 = 0) then doc('Styles.xml')//colors/color-1 else doc('Styles.xml')//colors/color-2 といった条件式を使用することで、XMLドキュメントから値を取得することもできます。'green' という XPath 条件式を定義することで、文字列をそのまま与えることもできますが、その結果は green という静的な値を与え た場合と変わりません。

■ 関連項目

- CSS スタイルと作業
- XPath ダイアログ
- スタイル
- プロパティ
- CSS ザポート

8.4.6 合成スタイル

合成スタイルはCSS テキストスタイルプロ、ティのグループとして表示されており、XML インスタンスドキュメントにおけるノードの属性の値に対 応します。更に、スタイルシートに収められている任意のCSS テキストスタイルプロ ティグループも合成スタイルとして扱われます。合成スタイ ルお以下のデザインコンポーネントにて指定することができます:

- 自動計算
- (コンテンツ)プレースオリダ
- 段落(ブロック)デザイン要素
- テーブルセル

合成スタイルの利点

合成スタイルコお以下のような利点があります:

- スタイルプロパティは XML のデータで、ユーザーにより編集することができます。
- 上に示されるデザインコンポーネントのスタイルプロパティは、SPS にて割り当てられた XML に収められている複数のプロパティとなります。
- SPS デザインの作成段階にて、SPS デザイナーは要素に関連付けられている複数の合成スタイル間を素早く切り替えることができます。

属性にて合成スタイルを入力する

複数のスタイルプロ、ティによく構成される合成スタイルは、ソース XML ドキュメトの要素における属性の値として入力されます。例えば、以下に示される XML ソースドキュメントのdesc-sytle 属性には、デフォルトの合成スタイルが含まれます:

```
<Desc desc-style="font-family:Verdana; font-size:12pt; color:blue">
```

同一要素に対して複数の合成スタイルを指定することもできます。この場合、各合成スタイルをそれぞれの属性に入力する必要があります:

```
<Desc styleBlue="font-family:Verdana; font-size:12pt; color:blue"
    styleRed ="font-family:Verdana; font-size:12pt; color:red">
```

同一の要素にて複数の合成スタイルを指定することができる場合、デザインコンポーネントの合成スタイルプロパティーで値を入力する際に、合成スタイルの切り替えを行うことができます(以下を参照くたきい)。

メモ: XML ドキュメントを妥当なものとするために、合成スタイルへアクセスするために使用される属性は、ソーススキーマにて定義する必要があります。

サポートされている CSS テキストスタイルプログティ

合成スタイルでは、以下のCSS スタイルを使用することができます:

font-family font-size font-weight font-style color background-color text-align text-decoration

合成スタイル値として属性を設定する

デザインコンポーネントの合成スタイルが属性として定義された場合、Authentic View ユーザーによりにの合成スタイルを編集することができるようしています。

以下の操作により、デザインエンポーネントの合成スタイルとして属性をセントすることができます:

1. デザインビューにて、属性を合成スタイルとして割り当てるデザインコンポーネントを選択します。以下のスクリーンショットでは、Desc 要素の(コンテンツ)プレースホルダが選択されています。



2. コンテンソコンポーネトの合成スタイルプロ 守ィに表示されているコンポポックスに、コンテキスト要素の属性が表示されます(上のスクリーンショナの右下にあるプロ 守ィウィンドウを参照(が込い)。デザインコンポーネントの合成スタイルとしてセナする属性を選択してくがさい。コンポポックスコは空のエントリーも表示されている点に注目してくがさい。

XPath 条件式を合成スタイルの値としてセナする

XPath 条件式を合成スタイルプロパティの値としてセットすることもできます。しかしこの場合、合成スタイルの値が (XML ソースドキュメントではなく)SPS 内部に収められるため、Authentic View にて合成スタイルの編集を行うことができなくなります。

XPath 条件式を合成スタイルプロ・ティの値としてセナするコよ プロ・ティサイド・ドーのソール・ドーズあ XPath アイコンをクトックして、表示される XPath ダイアログニて XPath 条件式を入力します。 XPath 条件式は属性値テンプレートとして評価され、戻り値は HTML style 属性の値(その他の出力フォーマナでは、それに準ずる値) しています。

例えば、以下に示されるn1:Person 要素の(エンテンツ)プレースヤルダーで作成されたXPath 条件式を考えてみましょう。

```
if (number(n1:Shares) gt 1000) then 'color:red' else 'color:green'
```

この条件式を使用することで、n1:Person 要素にn1:Shares とり子要素が含まれており、その数値が1000 を超える場合には n1:Person 要素の文字が赤色で出力され、それ以外の場合には緑色で出力されるようごとはす。XPath 条件式により返された値は、 HTML style 属性の値として(その他の出力フォーマナでは、それに準ずる値が、出力ドキュメントに渡されます。

SPS により生成されたXSLT スタイルシートでは、このXPath 条件式は以下にあるような属性値テンプレートとして評価されます:

HTML 出力では、条件の評価結果により、以下のどちらかが生成されます:

```
<span style="color:red">
```

おたは

メモ 属性値テンプレートは、属性の値が XPath 条件式として読み取られる XSLT コンストラクターとなります。 波かっこにより区切られ、属性の値を動的に割り当てることができます。

□ 関連項目

- CSS スタイルと作業
- XPath ダイアログ
- スタイル
- <u>プロ ディ</u>
- <u>CSS サポート</u>

追加機能 321

9 追加機能

このドキュメンテーションで記述された<u>コンテンソの編集、構造、高度な機能</u>、様々な<u>表示方法</u>の他にも、StyleVision には様々な機能が搭載されています。これら機能の詳細については、以下に記されるサブセグションにて記述されます。

- 日付との作業:更に、日付の操作やフォーマナの編集を行うこともできます。
- <u>パースされていないエンティティ URI</u>: XML ドキュメントがベースとしている DTD にパースされていないエンティティを収めることができます。パースされていないエンティティ URI 機能により、イメージやソイパーリングを URI のターゲット URI として使用することができます。
- スクレトの使用. StyleVision (コま Java Script エディターが搭載されており、 Java Script 関数を定義することができます。これらの関数は SPS における任意の場所からイベトハンドラーとして使用することができ、 HTML ドキュメントの出力に対して影響を与えることができます。
- HTML の心ボー: HTML ファイルを Style Vision にんポーナ することができるまか、HTML ファイルから XML、XSD、 そして SPS ファイルを作成することができます。
- XSLT かの新規作成: SPS ファイルを XSLT-for-HTML や XSLT-for-FO テンプレート構造から作成することができるまか、SPS にて XSLT におけるスタイルルールを作成することができます。 SPS エンポーネントを修正して、コンテンソやフォーマトを SPS に加えることができます。
- □ 関連項目
- プロプァナナイバー

9.1 パースされていないエンティティ URI

DTD を使用していて、その中でパースされていないエンティテを宣言した場合、そのエンティティ・関連付けられているURI をSPS 内にあるイメージやソイパーレクのターゲットとして使用することができます。この機能により、同じURI をSPS にて複数回使用することが可能しなります。この機能では、DTD にあるパースされていないエンティティのURI を出力、。

この機能を使用するコよ、以下の様に、DTD、XML ドキュメナ、SPS ドキュメナを適切に編集する必要が決ます:

- 1. DTD では、(i) URI と (ii) (エンティティのリソースタイプに StyleVision が指定された)記述により、パースされて、ないエンティティを宣言する必要がおます。
- 2. XML ドキュメントでは、パースされていないエンティティの名前を与えることにより、エンティティの参照を行う必要がます。
- 3. SPS では、動的ノードの値をパースされていないエンティティとして適切にアクセス することで イメージ とノイパーレク のターゲルとして使用することができます。.

パース去れていないエンティティの宣言と参照と参照

以下にXMLドキュメトの一部を示します。内部 DTD サブセナにはよースされていないエンティティが2つ含まれており、1つには 表記が もう1つには 表記が含まれています。XML コードにある img/@src ならびにlink/@href ノードにて名前を与えることにより、パースされていないエンティティへの参照が行われます。

パース去れていないエンティティを使用する SPS イメージと イパールク

以下の操作により、SPS 内でパースされていないエンティURI を参照するイメージやハイパーリンクを使用することができます:

- 1. メニューオプションの「挿入」から、イメージやハイパーリンクを挿入します。
- 2. 編集ダイアログにて動的タブプロパティを選択し、パースされていないエンティティの名前を含んでいるノードを 選択する XPath 条件式を入力します(以下のスクリーンショットを参照)。 上の XML ドキュメントサンプルの場合、XPath 条件式には //img/@src ならびに //link/@href ノードを入力することになります。



3. ダイアログの下部にある「XSLT 変換に対して分析されていないエンティティとして扱う」のチェックボックス にチェックを入れます。これで選択されたノードのコンテンツがパースされていないエンティティとして読み込まれることになります。指定された名前のエンティティが宣言されている場合、そのエンティティに関連付けられている URI が、リソース(イメージまたはハイパーリンク)の指定に使用されます。

スタイルシートの処理が行われる際には、エンティティ名に関連付けられる名前がエンティティ名の代わりに使用されます。

火モ URI が相対的な URI となる場合、XSLT プロセッサーにより DTD のベース URI により絶対 URI への展開 が行われます。従って、パースされていないエンティティが相対 URI の 'nanonull.gif' により定義されており、 DTD が someFolder 以下に配置されている場合、この URI は file:///c:/someFolder/nanonull.gif となります。

□ 関連項目

- イメージの使用
- ハイパーレクの定義

9.2 XSLT、XSL-FO または FO ファイルからの新規作成

SPS デザインは、HTML の出力を行うさめのXSLT ファイルや、PDF ならびにFO ファイルを出力するさめのXSLFO コマイドを伴う既存のXSLT ファイルをベースに作成することができます。 つまり、1からデザインを作成する代わりに、 既存のXSLT ファイルをデザインのひな形として使用することができます。

XSLTステップ から SPS を作成する

以下の操作により、XSLT、XSLT-for-FO、ませまFO ファイルからSPS ファイルを作成することができます:

- 1. メニューコマンドから「ファイル | 新規作成 | XSLT、XSL-FO または FO ファイルから新規作成…」を選択します。
- 2. ファイルを開くダイアログにて、目的のファイルを選択します。
- 3. 次に表示されるダイアログにて、SPS がベースとして使用するスキーマを選択するよう促されます。目的のスキーマを選択してください。
- 4. XSLT または FO ファイルの構造やフォーマットをベースにした SPS が作成され、デザインビューに表示されます。
- 5. あとは通常の方法で SPS を修正することができます。例えば、スキーマソノーからノードをドラッグし、スタイルや表示方法の修正や新たなスタイルの追加、<u>自動計算</u>と条件テンプレーといた Style Vision の機能を使用することができます。
- 6. SPS を保存して、<u>作業用 XML ファイル</u>を使用した<u>多種の出力の書式</u> のプレビューを行うことができます。更に<u>生成されたファイルの保存</u> コマンドにより、<u>スタイルシートと出力ファイルの生成</u> を行うことができます。

サンプル

XML ファイルの内容を以下に示します。

■ XML file used in charts example: YearlySales.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
      xsi:noNamespaceSchemaLocation="YearlySales.xsd">
       <ChartType>Pie Chart 2D</ChartType>
       <Region id="Americas">
               <Year id="2005">30000</year>
               <Year id="2006">90000</year>
               <Year id="2007">120000
               <Year id="2008">180000
               <Year id="2009">140000</year>
               <Year id="2010">100000
       </Region>
       <Region id="Europe">
               <Year id="2005">50000</year>
               <Year id="2006">60000</year>
               <Year id="2007">80000</year>
               <Year id="2008">100000</year>
               <Year id="2009">95000</year>
               <Year id="2010">80000</year>
       </Region>
       <Region id="Asia">
               <Year id="2005">10000</year>
               <Year id="2006">25000</year>
               <Year id="2007">70000</year>
```

XSLT ファイルの内容を以下に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"</pre>
xmlns:xs="http://www.w3.org/2001/xMLSchema" xmlns:fn="
http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
   <xsl:template match="/">
      <html>
         <head>
            <title>Simple Example for New From XSLT</title>
         </head>
            <xsl:apply-templates/>
         </body>
      </html>
   </xsl:template>
   <xsl:template match="Data">
      <xsl:for-each select="Region">
         <h1 style="color:red">
            <xsl:apply-templates select="@id"/>
         </h1>
      </xsl:for-each>
   </xsl:template>
</xsl:stylesheet>
```

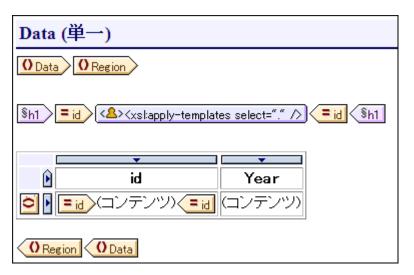
上記のステップ1から4の操作を行うことにより、以下に示されるような SPS がデザインビューに表示されます:



SPS はお2つのテンプレートが作成されている点に注目してください。HTML プレビューは小暦えを行うと、h1 要素のスタイル(color:red;)がSPS へ渡されていることが確認できます。

Americas Europe Asia

デザインビューにて h1 要素を選択し、(スタイルサイドバー」する色プロ ティグループにて) color の値を black へ変更します。その後スキーマソトカら Year 要素をドラッグして、以下のスクトーシショナで示される場所にてテーブルを作成します。 Year ID か最初のカラムに表示されるように、カラムの修正を行います。



ドキュメトのエンテンソ、構造、表示方法プロペティにその他の変更を加え、プレビューコでその出力を確認した後に、<u>生成されたファイルの保存</u>コマンドを使用することでファイルを保存することができます。

□ 関連項目

- SPS ファイル: コンテンソ
- SPS ファイル 構造
- SPS ファイル プレゼンテーション

ユーザー定義 XPath 関数 9.3

SPS ではカスタマイズされた XPath 2.0/3.0 関数を定義することができます。ユーザー定義されたXPath 関数は、XPath 条件式を 使用することができる任意のデザインコンポーネント(例えば自動計や条件、コンポジックスなど)にて再利用することができます。

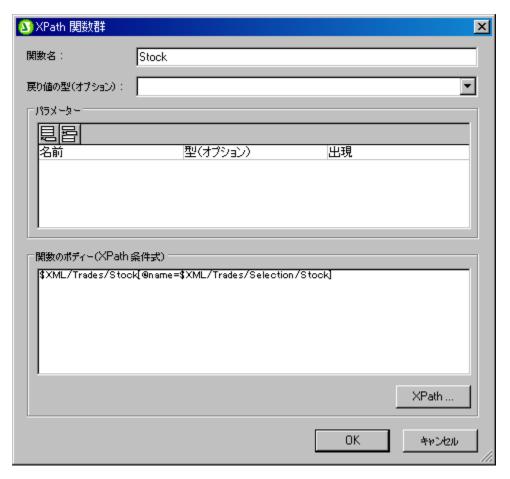
ユーザー定義 XPath 関数の定義と編集

ユーザー定義 XPath 関数は スキーマソノーサイドバーまたはデザインソノーサイドバーにて作成(そして編集)することができます(以下のスク リーンショナを参照)。SPS 内にある全てのユーザー定義 XPath 関数はスキーマソノーならびにデザインソノーサイドバーにある XPath 関 数アイテム以下に表示され、両方のサイドバーからアクセスすることができます。



ユーザー定義 XPath 関数を作成するコよ XPath 関数アイテムにある 🖭 アイコンをクリックします。XPath 関数ダイアログが表示され ます(以下のスクリーンショナを参照)。既に作成された関数の編集を行うコよ サイドバーのXPath 関数リストに表示されているエトリー をダブルクトックします。XPath 関数ダイアログか表示され、関数の定義を編集することができます(以下のスクリーンショナを参照)。

328 追加機能 ユーザー定義 XPath 関数



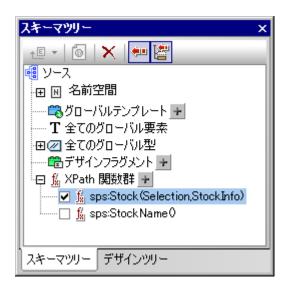
ユーザー定義 XPath 関数が作成されると、その関数をデザインにおける任意の場所で使用することができるようにないます。

ユーザー定義 XPath 関数の名前空間

ユーザー定義 XPath 関数は、http://www.altova.com/StyleVision/user-xpath-functions 名前空間に作成されます。この名前空間にはsps: とうプレフィックスか与えられており、ユーザー定義 XPath 関数を呼び出すには、この名前空間プレフィックスを使用する必要があります(例: sps:MyFunction())。

ユーザー定義 XPath 関数の有効化と無効化

ユーザー定義 XPath 関数エイリーの左にあるチェックボックスにチェックを入れる/外すことで、そのユーザー定義 XPath 関数を、それぞれ 有効化/無効化することができます(以下のスグリーシンコケを参照)。



同じ名前の関数が2つ以上存在するような場合に、この機能を使用することができます。インポートされたSPS モジュール内部に同じ名前の関数が含まれているような時に、同名の関数が表示されることがあります。

ユーザー定義 XPath 関数を呼び出す

ユーザー定義 XPath 関数は、デザイン内にある XPath 条件式から呼び出すことができます。例えば、上記で定義された sps:MyFunction と ウユーザー定義 XPath 関数を、自動計算内部にある以下の様な XPath 条件式にお呼び出すことができます:

sps:MyFunction()/@name.

このXPath 条件式は、以下のように評価されます:

- 1. sps:MyFunction() が評価されます。関数が以下の様に定義されていると仮定しましょう: \$XML/Trades/Stock[@name=\$XML/Trades/Selection/Stock]。関数が評価されると、/Trades/Selection/Stock 要素のコンテンツにマッチした name 属性の値を持った /Trades/Stock 要素が返されます。
- 2. ステップ1の結果が関数呼び出し元の XPath 条件式へ戻されます。/Trades/Stock 要素にある name 属性の値が、自動計算の値として返されます。

関数の削除

関数を削除するコよ、スキーマソノーまけはデザインソノーサイドバーに表示されている XPath 関数を選択して、サイドバーのソールドーであるアイテムの削除アイエンをクトックします。まけよ XPath 関数を右っクトックして、コンテキストメニューからアイテムの削除を選択することもできます。

9.3.1 XPath 関数の定義

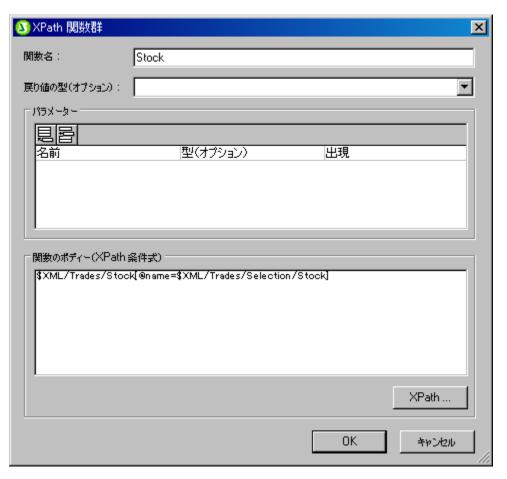
ユーザー定義 XPath 関数には(i) 名前(テキスト文字列)と(ii) 定義(XPath 条件式)の両方が必要になります。

更に、関数に対して1つ以上の引数(パラメータ)を指定することができます。ユーザー定義 XPath 関数には戻り値の型を(戻り値の型 コンボボックスのドロップダウンノストから型を選択することにより、オプランとして指定することできます。データ型を指定することにより、戻り値の値が選択されたデータ型に従ってものなのかをチェックすることができます。選択されたデータ型に戻り値が変換される訳ではおいにに注意してくたさい。型のミスマッチが発生した場合、エラーが返されます。戻り値のデータ型が指定されなかった場合、データ型のチェックが行われることはおりません。

330 追加機能 ユーザー定義 XPath 関数

ユーザー定義 XPath 関数が作成されると、デザイン内の任意の場所で使用することができるようにはます。XSLT スタイルシートでは、(以下に示されるように)xsl:stylesheet 要素の子要素である xsl:function 要素として作成されます。

以下のアグレーシンョナに示されるsps:Stock 関数からよ、Trades/Selection/Stock 要素のエンテンソニマッチするname 属性の値を持った。Trades/Stock 要素が返されます。上に示されるsps:Average 関数からは、3つの入力パラメーター値の平均が返されます。関数の定義では、avg() とう XPath 2.0 の関数が使用されます。戻り値のデータ型にはavg() 関数において入力値のxs:integer データ型の値が評価された際に返されるxs:decimal が指定されています。戻り値のデータ型を指定することで、値がそのデータ型に従ったものであるかのチェックが行われ、そうでは、場合はエラーが返されます。



関数の定義

関数の定義を行うコよスキーマソーまけはデザインソーコあるXPath 関数アイテムのサーアイコンをクトゥクします(上のスクノーシンョナを参照)。既に作成された関数を編集するコよサイドバーに表示されているXPath 関数をダブルクトゥクしてください。関数の名前を入力して、その定義を関数のボディーペインに入力します。必要な場合は、引数の定義も行うことができます(詳細については、パラメーターとノーケンスとハラメーターとノードのセクションを参照ください)。関数の戻り値に対してデータ型を指定することもできます。

XPath 関数を構成するXPath 条件式を記述する際に最も注意すべき点は、XPath 条件式に対してコンテキストノードが与えられていないということです。XPath 条件式により一ドの位置を特定する必要がある場合、以下の方法のどれがによりエンテキストノードを得ることができます。

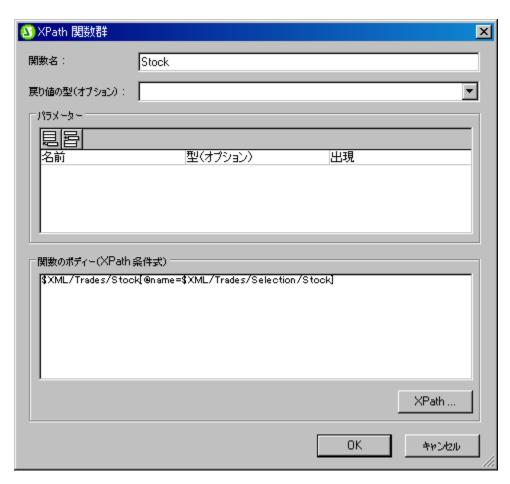
- 1. XPath 条件式をドキュメントのルートから開始します。ドキュメントルートは、\$XML という XPath 条件式により与えられます。例えば、\$XML/Trades/Stock[1] という XPath 条件式により、/Trades 要素以下にある最初の Stock 子要素が選択されます。メインスキーマのドキュメントルートを表す \$XML という変数は、StyleVision にある全ての SPS デザインにてグローバルに定義されています。
- 2. 引数にコンテキストノードを与えることができます。詳細については、<u>パラメーターとノード</u>を参照してくたさい。

以下のような状況で、エラーが返されます:

- 引数が定義されているが、関数のボディーで使用されていない場合。
- 関数により返される値のデータ型が、関数にて定義された戻り値の型にマッチしない場合。
- SPS 内にある関数のどれかにエラーが含まれる場合、(エラーを含む関数が呼び出されない場合でも)デザイン 全体に対して XSLT エラーが生成されます。この場合、ユーザー定義 XPath 関数のリストにあるチェックボックス のチェックを外すことで、関数を無効化することができます。デザインにて無効化された関数は、デザインから生 成された XSLT ドキュメントに含まれません。これにより、エラーを含む XPath 条件式を XSLT に含めず、XSLT エラーを回避することができます。

9.3.2 ノードを検索するための関数の再利用

直前のセクションでは、ノードを特定する XPath 関数を作成する方法について説明しました。以下のスクリーシンョナにて示される sps:Stock 関数により、/Trades/Selection/Stock 要素のエンテンソにマッチする name 属性の値を持った/Trades/Stock 要素 だが 返されること プルます。



XPath 条件式の\$XML/Trades/Stock[@ name=\$XML/Trades/Selection/Stock] にて定義されている位置を、例えば以下の様にモジューリルようことができます:

- * \$XML/Trades/Stock という定義の sps:Stocks() 関数
- * \$XML/Trades/Selection/Stockという定義の sps:SelectedStock() 関数

元からあったXPath 条件式は、以下のように書き直すことができます:

sps:Stocks()[@name=sps:SelectedStock()]

このような方法でXPath 関数を作成することにより、SPS デザイン上の他の場所で作成されたXPath 条件式からも関数を再利用することが可能になり、複雑なXPath 条件式の記述を、大幅に簡易化することができます。

9.3.3 XPath 関数内のパラメーター

ユーザー定義 XPath 関数には任意の数のパラメーターを割り当てることができます。 関数のパラメーターは XPath 関数ダイアログのパラメーターペイノニて定義されます(以下のスクリーンショナを参照)。 これらパラメーターは、ユーザー定義 XPath 関数の定義(関数のボディーペイン)にて使用することができます。

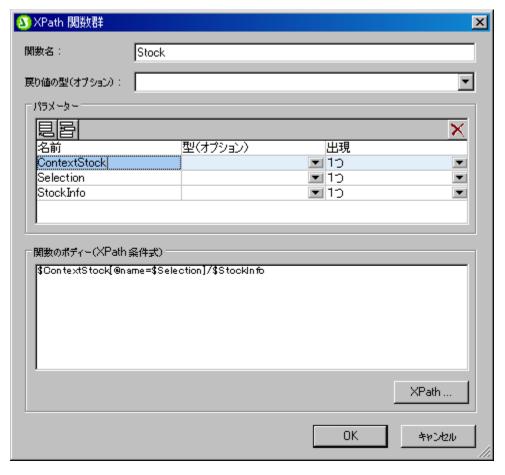
ユーザー定義 XPath 関数のメカニズム

XPath 関数は以下のように動作します:

- 1. 自動計算などで定義された関数呼び出しにて指定された引数の数は、ユーザー定義関数(以下のスクリーンショットにある、ユーザー定義関数のパラメーターペイン)にて定義されたパラメーターの数にマッチしなければなりません。更に、関数呼び出し内の各引数に対して与えられたアイテムの数も、対応するパラメーターにて定義された出現のオプションにマッチする必要があります。パラメーターに対してデータ型の制約が与えられている場合、引数により与えられた値もそのデータ型にマッチする必要があります。
- 2. 関数のパラメーターに対して渡された引数は、XPath 関数(以下のスクリーンショットに示される関数ボディー) にて使用されることになります。XPath 条件式を評価した結果は、オプションとして定義することのできる戻り値の型に対してチェックされます。データ型が期待されたものである場合、関数を呼び出した XPath 条件式にてその結果が使用されることになります。

パラメータの順序

関数が呼び出されると、パラメーターペインにて定義された順序に従って呼び出しに使用される引数が、ラメーターへ渡されます(以下のスクリーンショナを参照ください)。



上のスクーンショナーあるようなユーザー定義 XPath 関数のsps:Stock が定義され、以下のXPath 条件式により呼びされた場合を考えてみました。

sps:Stock(\$XML, Node1, Node2)

これら3つ0万|数(\$XML、Node1、Node2)は、関数のプラメーターにて定義された順字に従い、それぞれ \$contextstock、\$Selection、\$StockInfo へ渡されます。

関数呼び出いて記述された各引数はエンマにより分離されていることで注目してください。これにより、エンマにより区切られた関数呼び出し内の各引数が対応する。ウメーターへ渡されることではます。

/ ラメーターペイン内にある/ ラメーターの順序は、/ ラメーターペインにある追加、挿入、削除アイコンを使用することで変更することができます。

パラメータのデータ型

オプンコンとして、ユーザー定義関数における ウメーターのデータ型を定義することができます。データ型が指定された場合、入力された引数のデータ型が ウメーターのデータ型に対してチェックされ、型がマッチしない場合にはエラーが返されます。この機能により、(関数呼び出しの別数がに得られる)入力データをチェックすることができます。

出現

ユーザー定義 XPath 関数の各・ウメーターは、シーケンスとして考えることができます。・・ウメーターの出現プロ・ティにより、関数呼び出しの引数からその・ウメーター(ネオしてどれだ)カアイテムを送信することができるのかを指定することができます。

関数の定義と関数呼び出しの両方において、パラメーターや引数の分離だけではなく、シーケンス内にあるアイテムの分離にもコンマは使用されます。 そのため、 使用されている文脈から、 コンマが デメーター/引数の分離に使用されているのか、 まけばシーケンスのアイテムを分離するのに使用されているのかを理解する必要があります。

- パラメーター/引数は、かっこにより関数定義内のパラメーターや関数呼び出し内の引数にあるシーケンスを分離することができます。
- シーケンス内にあるかっこは無視されます。

理解を深めるための例とその説明を以下に示します:

- パラメーター/引数内部にあるかっこ: avg() や count() といったいくつかの XPath 2.0 関数は、単一のシーケンスを引数として受け取ります。このシーケンスにて値がコンマにより区切られる場合、または範囲演算子が使用されている場合、シーケンスをかっこで囲むことにより、それらの値がコンマにより区切られた複数のシーケンスではなく、単一のシーケンスであることを明示的に示す必要があります。例えば、avg((count(\$a), \$b, \$c)) という関数呼び出しにて、XPath 2.0 関数の avg() は (count(\$a), \$b, \$c) という単一のシーケンスを引数として受け取ります。シーケンスのアイテムを列挙することで、3つのアイテムからなるシーケンスが構成されるため、引数は単一の引数として、かっこで囲まれた形式で avg() 関数へ渡す必要があります: avg((count(\$a), \$b, \$c))。内側に記述されたかっこが無い場合、3つの引数が avg() 関数に対して渡されることになり、(avg() 関数ではシーケンスからなる1つの引数しか期待されていないため)エラーが返されることになります。
- パラメーター/引数内部にかっこを使用しない: 上の例と同様に、count() 関数も単一のシーケンスを関数の引数として受け取ります。しかし、この例で示された count(\$a) という呼び出しでは、コンマによる区切りは行われておらず、変数/パラメーターの \$a により値の取得が行われます。そのため引数内部にてかっこを使用する必要はなく、count(\$a) という表現は正しいものとなります。
- 関数呼び出し内にあるかっことコンマ: 関数呼び出しでは、(XPath 関数ダイアログのパラメーターペインにて 定義された)対応するパラメーターに対して各引数が使用されるように、かっこを適切に使用する必要がありま す。例えば、MyAverage()というユーザー定義 XPath 関数の定義にて avg(count(\$a), \$b, \$c) という XPath 2.0 条件 式が使用された場合、MyAverage((1,2,3),4,5) という関数呼び出しは妥当なものとなります。シーケンスの (1,2,3) が パラメーター \$a に対して与えられ、シングルトンシーケンスの 4 と 5 が \$b と \$c にそれぞれ与えられます。オプ ションとして、シングルトンシーケンスをかっこで囲むこともできます。MyAverage() から返される値は、この場合 4 となります。

9.3.3.1 パラメータとシーケンス

ここで、ウメーターとノーケンスの関係と、パウメーターとノーケンスがXPath 条件式でどのように使用されるかを理解しておくっきでしょう。以下のようにれらの用語を定義します:

シーケンスは、アトミック値またはノードからなるアイテムにより構成されます。シーケンスのアイテム間にコンマを使用することでシーケンスを構築することができます。

- XPath 2.0 関数はパラメーターを受け取るように定義することができます。例えば、関数の定義にて使用される count(\$a) という XPath 2.0 条件式におけるかっこ内の値 \$a が関数のパラメーターとなります。
- 引数は関数呼び出しにて使用されるアイテムのことです。例えば、count(//Person) という関数呼び出しでは //Person という引数が与えられています。この引数によりノードのシーケンスが count() の引数として与えられることになります。
- substring('styleVisionExamples', 6, 6) という関数の呼び出しでは3つの引数が与えられており、Vision という文字列が返されます。この呼び出しは substring() 関数の定義に対して妥当なものとなります。関数の呼び出しにて複数の引数が与えられた場合、それらはコンマにより区切られることになります。

シーケンスの区切りとして使用されるかっこ

XPath 2.0 条件式を構築する際によコンマにより区切られたアイテムや範囲演算子により構成されるシーケンスを区切るためにかっこが使用され、結果としてかっこにより区切られたシーケンスは1つのパラメーターまたは引数として読み取られるとうことに留意してください。

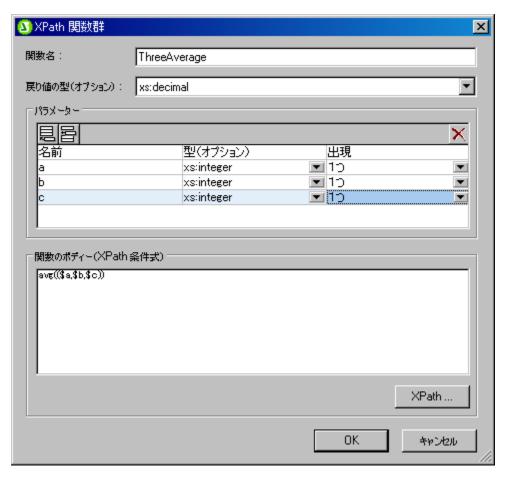
パス条件式は単一のプラメーターまけは単一の別数として読み取られるため、//Person/@salary といずパス(ロケーター)条件式の問じでかって使用する必要は必ずしもありません。パス条件式は単一のプラメーター/別数からなるシーケンスとなります。

上の説明を補足するための例を以下に示します:

- avg((10, 20, 30)): XPath 2.0 の avg 関数は、複数のアイテムが含まれているシーケンスを1つ引数として受け取ります。シーケンスに含まれるアイテムはコンマにより区切られているので、単一のシーケンスを区切るために内部のかっこが必要になります。このかっこが無い場合、3つの引数が与えられることになり、関数呼び出しは不正なものとなります(外型の括弧は関数に対する括弧となります)。
- avg(//Person/@salary): パス条件式により、全 Person 要素のノードにある salary 属性が選択され、その属性値がシーケンスとして返されます。引数が読み取られる前にシーケンスが列挙されることは無いため、括弧をつける必要はありません。単一のパス(ロケーター)条件式が引数となります。パス条件式が評価され、その結果がシーケンスのアイテムとして関数へ与えられます。
- count((10 to 34)): 範囲演算子による列挙が行われます。引数が読み取られる前に、範囲演算子の 'to' により、コンマにより区切られたアイテムのシーケンスが生成されます。そのため、count() 関数呼び出しの引数にはコンマにより区切られた25個のアイテムが含まれることになります。これらのアイテムを単一のシーケンスとして読み取るためには、括弧を使用する必要があります。括弧が使用されない場合、関数呼び出しに25個の引数が与えられることになり、(count() 関数の引数には単一のシーケンスを使用する必要があるため) 関数呼び出しが不正なものとなってしまいます。
- count((10 to 34, 37)): 内側の括弧により、括弧内部のアイテムにより構成される1つのシーケンスが関数呼び出しに使用されます。この例では26個のアイテムからなる単一のシーケンスが与えられます。
- count(//Person): 単一の引数を与える場合、シーケンスの区切りとなる括弧を使用する必要はありません。
 XMLドキュメント内にある //Person ノードを選択するパス条件式が引数として使用され、選択されたノードがシーケンスのアイテムとして渡されます。

XPath 関数にてXPath パラメータを使用する

括弧がユーザー定義 XPath 関数にて使用された場合、(i) 関数呼び出しに使用された引数の数がユーザー定義 XPath 関数のそれとマッチするか、そして(ii) 期待されている型と出現に対して個々の引数が評価されるかを確認する必要があます。



上のスクレーンショナでは、3つの ウメーターが、ウメーターペインにつ 定義されており、これらの ウメーターを関数ボディーペインで使用する ことにより、XPath 関数が定義されています。

パラメーターにて定義された各パラメーターは、それぞれ単一のシーケンスとして考えることができます。出現プロパティでは、それぞれのシーケンスにて許されることはなるアイテムの数が指定されます。上のスケノーシショナに示される定義では、各パラメーターがシングルトンシーケンス(つまパアイテムがつだけ含まれるシーケンス)として定義されています。そのため、関数呼び出しの各引数には、アイテムがつ含まれるシーケンスが使用されることはないます。型プロパティにより、シーケンスのアイテムで対するデータ型が指定されます。

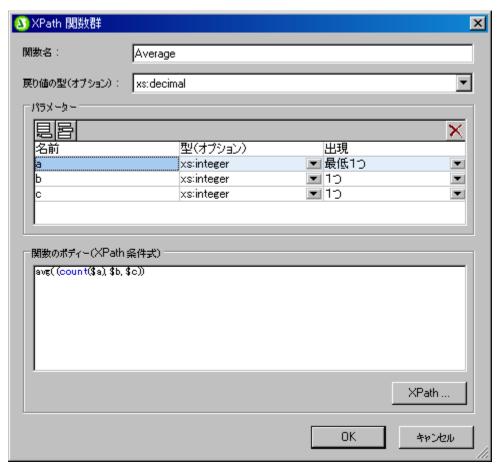
関数ボディーペインにある XPath 関数の定義では、それぞれ1つのアイテムから構成されるシーケンスが ラメーターにより与えられ、それらの 平均か球められます。これらの XPath パラメーターによりシーケンスが構成されるが、これらの、ラメーターを括弧で囲み、単一のシーケンスとして avg() 関数へ引数を渡す必要があります。 ランタイム こて与えられる(3つの) ラメーターに対応する) 関数呼び出しの引数がシングルトンシーケンスでは、場合、エラーが返されます。

上のスクノーンショナに示されるXPath 関数のThreeAverage() を呼び出すためのXPath パラメーターの例を以下に示します。デザインビューにて自動計算を挿入し、以下に示されるXPath 条件式を使用することで、結果を確認することができます。 関数には integer 型の引数が3つ与えられ、それらの平均が返されます。

- sps:ThreeAverage(10,20,30) により 20 が返されます。関数呼び出しには XPath 関数のパラメーターにマッチ する妥当な3つの引数が使用されています。
- sps:ThreeAverage((10),(20),(30))により20が返されます。3つのXPath/ウメータに対応する、3つの妥当な入力引数が存在します。各入力引数は(各シーケンスが単一のシーケンスであるが切に重複しているしかしながら重複はエラーではおい、括弧で囲まれています

- sps:ThreeAverage((10),20,30)により20が返されます。XPath 関数のパラメーターに対応する妥当な引数が3つ与えられています。第1引数が括弧で囲まれており、これは冗長な表現ですがエラーとはなりません。
- sps:ThreeAverage((10,20),(30),(40))によりエラーが返されます。第1引数がシングルトンシーケンスでは無いため、エラーとなります(\$a パラメーターの型プロパティには、「1つ」が指定されています)。
- sps:ThreeAverage((10,20,30))によりエラーが返されます。引数として与えられているのは括弧により囲まれたシーケンスが1つだけで、XPath 関数のパラメーター数にマッチしません。更に、シーケンスもシングルトンシーケンスではないため、不正となります。

/ ウメーターの出現プロ/ 守ィに最低1つがセナされた場合、1つまけは複数のアイテムを含むシーケンスを引数として使用することができます (以下のスクノーンショナを参照)。



上にある定義では、最初の、ラメーターは対して1つ以上のアイテムか含まれるシーケンスが定義され、残り2つの、ラメーターは対してシングルトンシーケンスが定義されています。最初の、ラメーターは対して渡されたアイテムの数を数え、その数と、その他2つの、ラメーターは対して渡された数の平均を返すように関数は定義されています。以下の点に注目してください。

- avg() 関数の呼び出しに使われる引数は括弧で囲まれています。これにより、3つのアイテムからなるシーケンスが、引数として avg() へ渡されます。count() 関数から返される値が最初のシーケンスアイテムに、パラメーター \$b と \$c がそれぞれ2番目と3番目のシーケンスアイテムになります。
- count() 関数呼び出しの引数は単一のシーケンスであることが明らかなので、括弧で囲まれていません。

上のスクノーンショナにて示されるAverage() XPath 関数を呼び出す際に使用する引数の例を以下に示します:

- sps:Average((1,2),3,4) により 3 が返されます。パラメーターに対応する3つの妥当な引数が与えられています。最初の引数は括弧により囲まれており、他の(シングルトン)シーケンスと区切られています。第1引数に対して count() 関数が処理され、2 という値が返され、その値が avg() 関数に渡されるシーケンスにおける最初のアイテムとして使用されます。
- sps:Average(4,4,4)により3が返されます。妥当な引数が3つ与えられています。最初の引数は1つのアイテムにより構成されるシーケンスでも良いため、他の引数と区別するために括弧を使用する必要はありません(対応するパラメーターの出現プロパティを参照ください)。

その他の留意するべき点

以下の点にも留意しておく必要があります:

- XPath 引数の出現に最低1つがセットされている場合、avg((\$a)) といった XPath 条件式を関数ボディーとする MyAverage() といった関数を定義することができます。この関数の引数には、1つ以上のアイテムが含まれるシン グルトンシーケンスが渡されます。関数は、例えば sps:((2,3,4)) という呼び出しにより 3 という値が返されます。3 つのシングルトンシーケンスではなく、単一のシーケンスとして引数を渡すために、引数は括弧で囲む必要があります。
- XPath パラメーターの \$a の出現プロパティに無しまたは1つが指定された場合、avg(\$a, \$b, \$c) といった XPath 条件式により構成される MyAverage() といった関数を定義することができます。この関数には3つのシーケンスが 引数として渡されるだけではなく、最初の引数値を空にすることも可能になります。最初のシーケンスを空にする 場合、空のシーケンスを最初の引数値として明示的に渡す必要があります(そうでない場合エラーが返されます)。sps:MyAverage(30, 20, 10) により関数の呼び出しを行うと、20 という値が返されます。sps:MyAverage((), 20, 10) という関数の呼び出しが行われた場合、(最初のシーケンスが空の入力値としてカウントされるため)15 という値が返されます。それに対して sps:MyAverage(20, 10) という関数の呼び出しでは最初の(空の)シーケンスが明示 的に示されておらず、第3引数が与えられていないと認識され、エラーが返されます。

複雑な例

XPath 条件式を再利用するとう以外にも、ユーザー定義のXPath 関数を使用することで、XPath 2.0 の関数セナでは提供されていない、複雑でカスタマイズされたXPath 関数を構築することができます。例えば、シングルトンシーケンスを引数として受け取るXPath 条件式を使用することで、階乗関数を簡単に作成することができます。\$num とり引数が求める階乗の数だとすると、関数を構成するXPath 条件式は以下のようにないます:

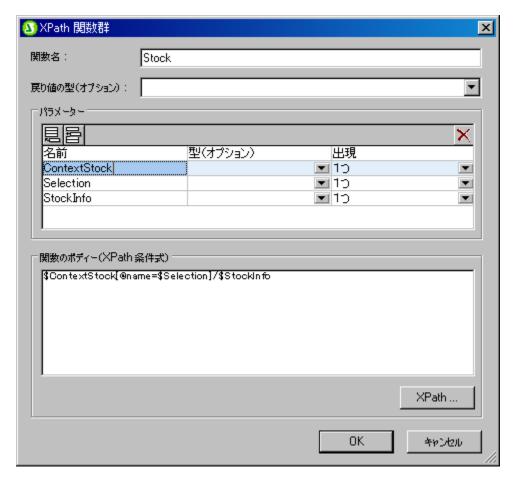
if (\$num < 2) then 1 else \$num * sps:Factorial(\$num - 1)

関数の名前をFactorial() とすると、例えばsps:Factorial(6) という関数呼び出しにより6 の階乗を求めることができます。

9.3.3.2 パラメータとノード

XPath 関数にてノードを選択する ウメーターを使用する場合、関数がデザイン内のどこから呼ばれるかに関わらず、その関数にはコンテキストノードが与えられていないということに注意する必要があります。 関数の定義内(関数ボティーペイン)で使用される XPath 条件式か、XPath 関数を呼び出す際に使用する XPath 条件式にてコンテキストノードを取得することができます。後者の場合、関数呼び出しの引数としてコンテキストノードが与えられることになります。

以下のスクレーンショナに示される、3つの ラメーターとともに定義された。Stock() というユーザー定義 XPath 関数を考えてみましょう



関数ボディーコおける定義は\$ContextStock[@name=\$Selection]/\$StockInfo とおっており、3つのXPath パラメーターが使用されているものの、コンテキストノードの情報は含まれていません。コンテキストノードの情報は、以下の例にあるように、この関数を呼び出す XPath 条件式から与えられるニとしています:

sps:Stock(\$XML/Trades/Stock, \$XML/Trades/Selection/Stock, @name)

関数呼び出しては3つの引数が使用されており、コンテキストノードの情報として使用される値とノードの識別を行うための情報が与えられます。以下のXPath 条件式により関数を呼び出し、同様の結果を得ることもできます:

```
sps:Stock( /Trades/Stock, /Trades/Selection/Stock, @name )
sps:Stock( /Trades/Stock, //Selection/Stock, @name )
```

デザインコンポーネトのXPath 条件式では、コンテキストノードが分かっているため、デザインコンポーネントにて使用される関数の呼び出しでは、ドキュメントルートを表す、\$XML 変数を使用する必要は必ずしもありません。

上に示される関数呼び出しは、定義された3つのXPath、デメーターは対して、それぞれ3つの別数値が与えられていることは自してください。

- · \$ContextStock = \$XML/Trades/Stock (/Trades/Stock 要素)
- * \$Selection = \$XML/Trades/Selection/Stock (/Trades/Selection/Stock 要素)
- \$StockInfo = @name

関数定義におる XPath 条件式は以下のようごなります:

\$ContextStock[@name=\$Selection]/\$StockInfo

引数が渡されることで、関数定義におけるXPath 条件式は以下のようじなります:

\$XML/Trades/Stock[@name=\$XML/Trades/Selection/Stock]/@name

関数に渡されるのはテキスト文字列ではなく、ノードセルであるとら点に留意してくたさい。

これにより、XPath パラメーターを介したコンテキストノードの受け渡しが行われ、その後関数にて目的のソードが特定、返されることはないます。

追加機能 日付の使用 341

9.4 日付の使用

ソースドキュメトに日付の値を受け取るノードが含まれる場合、使用されている XML スキーマニある xs:date や xs:date Time データ型を使用することで、XPath 2.0/3.0 (*下の例を参照*を参照)。 StyleVision では、xs:date まけは xs:date Time データ型に対して以下のような機能がサポーされます。 <u>入力フォーマル</u>機能により<u>日付フォーマルの範囲を広げます</u>。

メモ XPath 1.0 では日付ならびに時間データの操作を行うことができません。しかし、XPath 1.0 でも <u>日付のフォーマルを保証することができます。</u>

XPath 2.0 (まける日付の計算

<u>自動計算</u> における XPath 2.0 条件式では、日付形式のデーダを操作することができます。 XPath 2.0 条件式により行うことのできる例を 以下に示します:

- XPath 2.0 関数の current-date() ならびに current-dateTime() を使用することで、現在の日付と日時をそれぞれ取得することができます。
- 日付の計算を行うことができます。例えば、current-date() DueTime という計算に対して xdt:dayTimeDuration 形式 の値を得ることができ、例えば P24D という値により、プラス24日間という期間が表されます。
- XPath 2.0 関数を使用することで、期間から時間単位を取得することができます。例えば、day-fromduration(xdt:dayTimeDuration('P24D')) という条件式により、24 という数値が返されます。

自動計算における XPath 2.0 条件式の例を以下に示します。この条件式ではある項目の支払期限が過ぎた場合に、4%の年利に対する日割りの利子額を(単利計算で)計算し、その値と元本の和が返されます。

このような計算を行うには、DueDate 要素がxs:date といた日付型により定義されており、その要素のエンテンンがYYYY-MM-DD[±HH:MM] とう形式(時間部分はオプション)で記述されている必要があます。

- 関連項目
- 日付のフォーマナ

9.4.1 日付のフォーマット

XMLドキュメト内に収められる日付は、そのノードのデータ型にて指定された書式に従って保存されます。例えば、xs:date ノードの値には YYYY-MM-DD[±HH:MM] とう書式が使用され、xs:date Time 型のノードではYYYY-MM-DDTHH:MM:SS[±HH:MM] とうフォーマットが使用されます。デフォルトではXML 内部にあるデータがAuthentic View と出力の両方にてそのまま表示されます。しかし、SPS にて値フォーマト機能を使用することで、出力されたデータとは違った表示をAuthentic View にて行うことができるようによっては出力の表示も変更することができます。

値フォーマナトを日付に対して使用することで、以下のデータ型を使オンードや自動計算に対してカスタムフォーマナトを定義することができます:

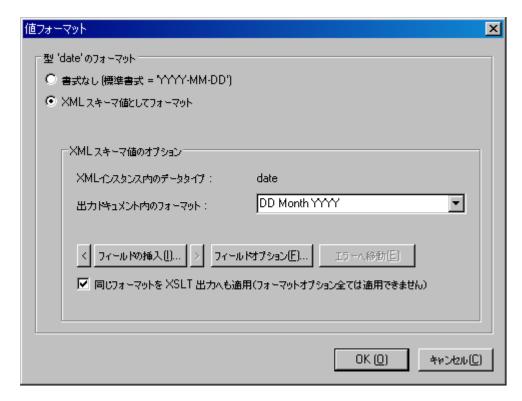
追加機能 342 日付の使用

- xs:date
- xs:dateTime
- xs:duration
- xs:gYear
- xs:qYearMonth
- xs:qMonth
- xs:gMonthDay
- xs:qDay

値フォーマルを使用して日付ノードのフォーマルを行う

以下の操作により、日付ノードにある実際のデータとは異なる日付表示を行うことが可能になります。

- 1. ノードのコンテンツプレースホルダまたは入力フィールドを選択します。コンテンツ、または入力フィールドとし て作成されたノードに対してしか値フォーマットは使用できない点に注意してください。
- 2. プロパティサイドバーにてコンテンツアイテムを選択し、コンテンツプロパティグループを選択します。値フォー マットプロパティの編集ボタン ・・・・ をクリックします。値フォーマットダイアログが表示されます(以下のスクリー ンショットを参照)。



デフォルトでは書式無しラジオボタンが選択されています。

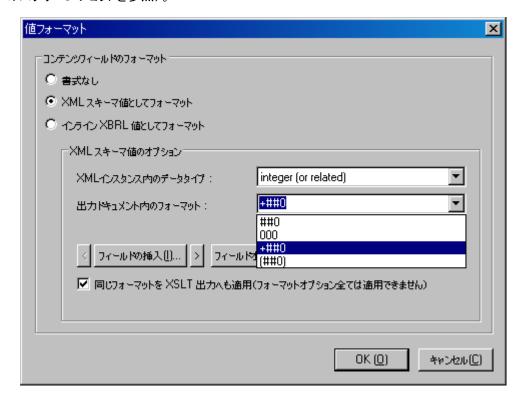
その他のフォーマットを定義するには、フォーマットラジオボタンを選択します。 4. コンボボックスのドロップダウンリストから定義済みの日付フォーマットを選択することができるほか、コン ボボックスの入力フィールドにて独自のフォーマットを定義することができます(以下のスクリーンショットを 参照)。独自のフォーマットを定義するための構文に関する詳細については 値フォーマット構文 のセグションを参 照けざい。

追加機能 日付の使用 343

値フォーマナによる自動計算のフォーマナ

日付フォーマナにより表示された値を、自動計算にて評価することができ、その結果のフォーマナを行うために値フォーマナを使用することができます。以下の操作によりにのような処理を行うことができます:

- 1. デザイン内にある自動計算を選択します。
- 2. プロパティサイドバーにてコンテンツアイテムを選択し、自動計算プロパティグループを選択します。値フォーマットプロパティの編集ボタン をクリックしてください。値フォーマットプロパティのダイアログが表示されます(以下のスクリーンショットを参照)。



デフォルトでは書式無しラジオボタンが選択されています。

- 3. その他のフォーマットを定義するには、フォーマットラジオボタンを選択します。
- 4. コンボボックスのドロップダウンリストから定義済みの日付フォーマットを選択することができるほか、コンボボックスの入力フィールドにて独自のフォーマットを定義することができます(以下のスクリーンショットを参照)。独自のフォーマットを定義するための構文に関する詳細については <u>値フォーマル構文</u> のセグシュンを参照(ナミン)。

値フォーマナを出力へ適用する

定義された値フォーマトは Authentic View. 更に、に対して適用されます。その他にも、しくつかの値フォーマト定義を(全てでは無いですが) HTML 出力に対して適用することができます。 XSLT 出力へも適用チェックボックスにチェックを入れてくたさい。このオプションがチェックされていないが利用できない場合、 Authentic View の出力に対してのみ値フォーマトが適用されます。

- □ 関連項目
- 値フォーマナ(数値データ型のフォーマナ)

9.5 スクリプトの使用

StyleVision では、各 SPS に対して JavaScript 関数を(デザインビューのダブとして用意されている) JavaScript エディターにで義することができます。ここで作成された関数は、HTML ドキュメントのヘッダー部分に保管され、HTML ドキュメントのボディー部から呼び出すことができます。 以下のような状況で、関数を使用することができます:

- 複数のスクリプトを使用して、複雑な結果を求める。この場合、必要とされているそれぞれのスクリプトを個別の 関数として作成し、デザインドキュメントからこれら関数を参照することができます。
- 特定のスクリプトをドキュメント内にある複数の箇所から呼び出す。

JavaScript エディターにて関数を定義する方法については JavaScript 関数の定義サブセクションを参照ください。

GUI では、SPS にて定義された全てのJavaScript 関数がデザインソノーウィドウのスクノトエトリー以下に表示されます(以下のスクノーシンコトを参照)。以下にあるスクノーシンコトではAverate、ImageOut、ImageOver、そして Buttons とり4つの JavaScript 関数が表示されており、SPS にて JavaScript が有効づいています。



JavaScript エディターにて定義された関数は、GUI ではイベトハンドラー呼び出しとして扱われます。デザイバキュメト内にあるエンポーネトを選択することで、プロ ディサイドバーのイベトプロ ディグループにあるイベトハンドラープロ ディから、定義されている全ての関数に対する割り当てを行うことができます。 イベトハンドラーズ はて JavaScript 関数を割り当てる方法については、関数のイベトハンドラーへの割り当てセグションに記述されます。

モジュールSPS 内のスクノプト

SPS モンュールが別のSPS モンュールに追加されると、追加されたモンュール内にあるスクノプトか参照元 SPS でも利用できるようには、参照元 SPS のエンポーネントに対してプログライサイドバーからイベントバンデーとして使用することができるようになります。モンュール SPS の使用に関する詳細については、モンュール SPS のセクションを参照(たさい)。

□ 関連項目

- デザインビュー
- デザインソノー
- モジュールSPS

9.5.1 JavaScript 関数の定義

以下の操作を行うことで、JavaScript 関数を定義することができます:

1. デザインビューにてデザインビューのタブをクリックして、ポップアップから JavaScript を選択し、JavaScript エディターを表示します(以下のスクリーンショットを参照)。



2. JavaScript エディターにて、関数の定義を記述します(以下のスクリーンショットを参照)。

```
1
         function DisplayTime 0
 2
       3
         now = new Date 0;
 4
         hours = now.getHours();
 5
         mins = now.getMinutes();
 6
         secs = now.getSeconds();
         result = hours + ":" + mins + ":" + secs;
 7
 8
         alert (result);
 9
10
111
         function ClearStatus()
12
       13
        Lwindow.status="";
14
```

上のスクリーンショットでは、DisplayTime と ClearStatus という2つの JavaScript 関数が定義されています。これらの関数はアクティブな SPS に対して記述されています。HTML ファイルでは、これらの関数が以下にあるようにヘッダー内部に記述されます。

```
<script language="javascript">
<!-- function DisplayTime()
{
    now = new Date();
    hours = now.getHours();
    mins = now.getMinutes();
    secs = now.getSeconds();
    result = hours + "." + mins + "." + secs;
    alert(result)
}

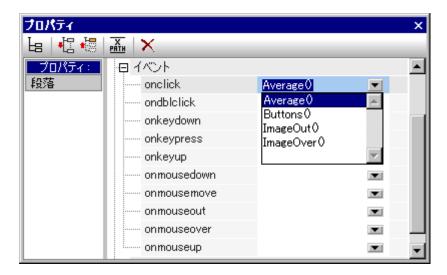
function ClearStatus()
{
    window.status="";
}
-->
</script>
```

これらの関数は HTML ドキュメント内部の任意の場所から呼び出すことができます。StyleVision では、 定義された関数をプロパティサイドバーのイベントプロパティグループにあるイベントハンドラープロパティ へ割り当てることができます。詳細については <u>関数のイベトハンドラーの割り当て</u>のセクションを参照ください。

- □ 関連項目
- デザインビュー
- イベトハイラーへ割り当て

9.5.2 関数をイベントハンドラーとして割り当てる

StyleVision GUI では、SPS エンポーネントがHTML にてレンダルングされた際に発生するイベントハンドラーとして JavaScript 関数を割り当てることができます。これらのイベトハンドラーはHTML 出力で使用されます。グローブル関数をイベトハンドラーとして割り当てることで、(例えば on Click のような)イベントに対してイベントハンドラーをセントすることができます。プロバティサイドバーでは、JavaScript エディターこで定義されたグローブル関数を、イベトプロバティグループにある各イベントのドロップダウンボックスからイベントハンドラーとして利用することができます(以下のスグリーシンョントを参照)。



以下の操作により、関数をイベトハンドラーは割り当てることができます。

- 1. イベントハンドラーを定義する SPS のコンポーネントを選択します。あらゆる種類のノードやコンテンツを(静的か動的かに関わらず)選択することができます。
- 2. プロパティサイドバーにてイベントプロパティグループを選択します。利用することのできるイベントが、属性カラムに表示されます(上のスクリーンショットを参照)。
- 3. 目的のイベントに対して、値カラムにあるコンボボックスの矢印をクリックします。JavaScript エディターにて定義された全ての関数が表示されます。
- 4. ドロップダウンボックスから、そのイベントのイベントハンドラーとして目的の関数を選択します。

イベトハンドラーが定義されたエンポーネントのイベトがHTML 出力にて起動されると、対応する Java Script 関数が実行されます。

□ 関連項目

- デザインビュー
- <u>JavaScript 関数の定義</u>

9.5.3 外部 JavaScript ファイル

SPS は以下にある2つの方法により外部 JavaScript ファイルヘアクセスすることができます:

1. ユーザー定義要素封/オユーザー定義 XML ブロックを作成する これらのデザインオブジェクト 口お外部 JavaScript ファイルへの アクセスを行う SCRIPT 要素を含めることができます。ユーザー定義要素またはユーザー定義 XML ブロックが、デザインの BODY 要素内部に配置されているということを確認してください。これにより HTML 出力の (HEAD 要素ではなく) BODY 要素内部にこれらのオブジェクトが配置されます。

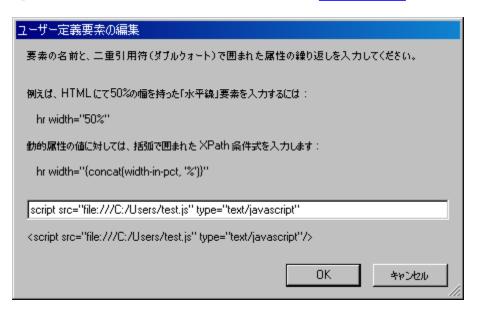
2. <u>JavaScript エディター内にスクレアトを追加する</u> このようにして作成されたスクレアトは、HTML 出力のHEAD 要素内に配置 されます。

ユーザー定義要素とユーザー定義 XML ブロック

<u>ユーザー定義要素</u> と<u>ユーザー定義 XML ブロック</u>を使用することで、外部 JavaScript ファイルヘアクセスすることができます。これらのメカニ ズ*込*を使用することで、外部 JavaScript ファイルへのアクセスを行う SCRIPT 要素を、出力 HTML ドキュメントのBODY 要素内にある 任意の場所へ挿入することができます。

以下の操作により、ユーザー定義要素を挿入することができます。

- 1. JavaScript ファイルへのアクセスを行う SCRIPT 要素を挿入する場所へカーソルを配置します。
- 2. 「挿入」メニューオプションまたはコンテキストメニューから、ユーザー定義要素を挿入するコマイを選択します。



- 3. ダイアログが表示されるため、上のスクリーンショットにあるように、JavaScript ファイルの URL を src 属性の値とする SCRIPT 要素を入力します(例: script type="text/javascript" src="file:///c:/Users/mam/Desktop/test.js")。
- 4. 「OK」をクリックして完了します。

ユーザー定義 XML ブロック を使用することでも同様の結果を得ることができます。ユーザー定義要素に対して記述された上記の方法と違うのよ(i) ユーザー定義 XML ブロック の代わりに ユーザー定義要素 が挿入され、(ii) 開始タグと終了タグを含む SCRIPT 要素全体が挿入されるという点です。

JavaScript エディター

<u>JavaScript エディター</u>を使用することで、HTML 出力のHEAD 要素に外部スクリプトを挿入することができます。JavaScript エディター にて作成された関数定義の外側に、以下にあるスクリプトフラグメントを入力してくたさい。

```
var script = ドキュメナ.createElement('script');
script.type = 'text/javascript';
script.src = 'file:///c:/Users/Desktop/test.js';
var head = ドキュメナ.getElementsByTagName('head')[0];
head.appendChild(script)
```

script.src にあるURL にて指定された外部 JavaScript ファイルが、出力 HTML ドキュメントのHEAD 要素からアクセスされます。

□ 関連項目

- JavaScript 関数の定義
- ユーザー定義要素
- ユーザー定義 XML テキストブロック
- イベトハイラーへ割り当て

9.6 HTML のインポート

StyleVision では、HTML ファイルをベースとしてインポートすることで、以下のファイルを作成することができます:

- インポートされた HTML ファイルのデザインと構造をベースにした SPS ドキュメト。
- HTML ドキュメントコンポーネントがスキーマ要素または属性として作成された XML スキーマ。オプションとして、HTML ドキュメントとは関係の無い要素や属性をユーザー定義スキーマに作成することもできます。
- (i) 作成した XML スキーマの構造をベースにして、(ii) HTML ファイルから得られたコンテンツを含む XML ドキュメント。
- デザインビューのデザインをベースにした XSLT スタイルシート。

HTML-からXML を作成するステップ

以下に示される手順により、インポートされたHTML ファイルをベースにしたXML ファイルの作成を行うことができます:

- 1. HTML インポートによる新たな SPS の作成: HTML ファイルが Style Vision ヘインポートされると、新規 SPS ドキュメトが 作成されます。HTML ドキュメトが HTML マーケアップタグとともにデザインビューに表示されます。ユーザールートと ら名 前のドキュメント 要素とともに、ユーザー定義 XML スキーマがスキーマソノーウィイ・ヴに表示されます。このスキーマが、SPS が、 ースとするスキーマとなります。この時点でデザインビューに表示されている HTML ドキュメントのエンテンソとマーケアップは、静的コンテンソとして SPS に含まれています。
- 2. <u>スキーマとSPS デザインの作成</u>: (i) HTML ドキュメトから(スキーマソノーナインが)スキーマソノー上の適切な場所についポーネントをドラッグするか、オプションとして(ii) 独自のソードをスキーマソノーへ追加することで、スキーマを作成します。 デザインウィンドウでは、スキーマソノーにてノードを構築するから使用された HTML コンテンツが、スキーマノードタグに囲まれて表示されます。 対応するスキーマノードが存在しない HTML コンテンツは、スキーマノード無して表示されます。
- 3. デザイドキュメトにて、必要に応じてノードのフォーマナトや処理ルールの調整、静的コンテンソの追加などを行います。これらの修正は SPS ならびに生成された XSLT においてのみ反映されます。生成されたスキーマや XML ファイルに反映されることはありません。
- 4. スキーマソナーとSPS デザインの構築が完了した後には、以下の項目の生成と保存を行うことができます:
- 作成したスキーマソリーコ対応する XML スキーマ
- SPS デザイン内にあるコンテンソプレースヤルダとともに作成されたスキーマのコンテンソと、スキーマをベースコーた構造を伴う XML データファイル
- SPS (.sps ファイル) おはデザインをベース ユ た X S L T スタイルシート

□ 関連項目

- ユーザー定義スキーマ
- XSLT からの新規作成

9.6.1 HTML のインポートにより新規 SPS を作成

以下の操作により、HTML ドキュメントから新たな SPS ファイルを作成することができます:

- 1. メニューオプションから「ファイル | 新規作成 | HTML ファイルから新規作成 | を選択します。
- 2. 表示されるファイルを開くダイアログにて、インポートする HTML ファイルを選択し、開くボタンをクリックします。
- 3. 相対パスを絶対パスへ変換するか尋ねられます。適切なオプションを選択してください。

350 追加機能 HTML のインポート

新規 SPS ドキュメトが作成されます。デザインビューコよ StyleVision にて使用することのできる定義済み HTML フォーマトのマーク アップととも、ボキュメトカ表示されます(以下のアクノーンショナを参照)。

■A Sample Article ■

This is a simple article, marked up in HTML. It is the example HTML document used to demonstrate how STYLEVISION can generate schema files, XML files, and stylesheets from an HTML document. It contains headlines, paragraphs, two lists, a table, an image, a hyperlink, horizontal rules, and inline italics. A number of these elements have attributes.

p It is important to note that the HTML-to-XML conversion involves two parallel processes: (p)

- · Creation of a schema tree
- Design of a STYLEVISION Power Stylesheet (SPS)

HTML ドキュメントがダインテンプレート内に表示されていることで注目してください。グロー・ジリテンプレートは存在しません。

スキーマソノーサイドバーにて、ユーザールートという名前のドキュメト要素とともに、ユーザー定義スキーマが作成されます(以下のスクノーンショナを参照)。



全てのグローバル要素のJストにグロージル要素が表示されていないことに注目してくたさい。

SPS 構造とデザイン

SPS は単一のテンプレート(メインテンプレート)が含まれており、このテンプレートは一時的な内部 XML ドキュメイのドキュメイノードへ適用されます。スキーマソノーウィドウェで作成されたユーザー定義スキーマの構造に従うかたちで、この XML ドキュメイトは作成されます。この 時点でデザインビューはは、メインテンプレート内部の HTML ドキュメイコンポーネイカ 静的コンポーネイトとして SPS 内に表示されます。これら HTML コンポーネイトは、Authentic View にて読み取り専用の非 XML コンテンソコンポーネイトとして処理されます。

以下にあるいずれかの方法により、ユーザー定義スキーマにて選択したHTML を、要素や属性として作成することができます:

- 1. 選択対象を要素または属性へ変換します。デザインにはノードタグがコンテンツプレースホルダとともに挿入されます。スキーマには要素または属性が作成されます。スキーマでは要素または属性が作成され、XMLドキュメントでは選択対象が XMLドキュメントにて作成されたスキーマノードのテキストコンテンツに変換されます。 XMLドキュメントにて作成されたノードのコンテンツは、SPS を介して得られた出力へ動的に挿入されます。
- 2. 選択対象を要素または属性で囲みます。デザインでは選択対象がノードタグにより囲まれ、コンテンツプレースホルダは挿入されません。つまり、選択対象は静的コンテンツとして SPS デザインに表示されます。スキーマでは要素または属性が作成され、XMLドキュメントでは、空のノードが作成されます。スキーマノードタグ内にある静的テキストは出力されるものの、コンテンツプレースホルダがデザイン内部で明示的に挿入されない限り、このノードに対して動的コンテンツが出力されることはありません。

コンテンソプレースホルダにより、XML ドキュメトから得られたデータが出力にで、表示される場所や(Authentic Viewにて)編集可能な場所が示されます。

- □ 関連項目
- ユーザー定義スキーマ

9.6.2 スキーマと SPS デザインの作成

デザインビューからユーザー定義スキーマへ選択対象をドラッグすることで、スキーマか作成されます。複数のスキーマを一度に作成することはできません。スキーマソノー内のノードへ選択対象をドロップすることで、子まけは兄弟として新たなノードが作成されます。作成されるノードの種類を要素と属性から選択し、選択対象を新たなノードの変換するか、囲むことができます。

選択対象

デザインビュー内にある選択対象は以下のどれかとなります。

- HTML ドキュメント内にあるノード
- ノード内にあるテキスト
- ノード間にあるテキスト文字列
- イメージ
- リンク
- テーブル
- リスト
- 上にあるコンポーネントの組み合わせ

このセケションでは、選択対象に対して行われる一般的な処理によって記述します。テーブルならびにリストの特別な場合によっては<u>要素/</u>属性としてテーブルやノストを作成のセケションを参照くたさい。

選択を行うコよ HTML ドキュメトコンポーネトをクリックするかテキスト文字列を イライトします。 複数のエンポーネトを選択するコよ ドラッグにより複数のエンポーネトを イライトします。 StyleVision にて複数のエンポーネトが イライトされた場合、 それらエンポーネトの親ノード以下にある全ての子ノードが含まれることに注意してください。

スキーマソノーにおける位置

目的のスキーマソノーノードへ選択対象をドラッグすると、マウスカーノルが以下にあるどちらかのシンボルに変わります。

352 追加機能 HTML のインポート

● 新たな子を作成シンボル → にてドロップすると、ドロップされたノードの子ノードとして選択対象が作成されます。

選択対象をドロップする際に、そのソードの兄弟まけは子ノードとして選択対象を作成するか決定することはよいます。

ノード作成方法の切り替え

選択対象をドロップするとコンテキストメニューか表示され、(i)作成されるノードは要素にするのか属性にするるのか、(ii)選択対象をノードに変換するか、ノードの制を選択対象で囲むのか、を選択することではより、以下のスクノーシショナを参照)。



以下の点に留意してください

- 選択対象がノード(要素または属性)に変換される場合、選択対象がコンテンツプレースホルダを含むノードタグに置き換えられます。デザインならびに出力では、選択対象のテキストコンテンツが静的コンテンツから削除されます。出力では、選択対象のテキストが XML ドキュメント内にあるノードの動的コンテンツとして表示されます。
- HTML ノードが XML ノードへ変換されると、HTML ノードタグ内部に XML ノードタグが挿入されます。
- (HTML ノードを含む)選択対象が XML ノードにより囲まれると、選択対象の直前と直後に XML ノードが挿入されます。デザインならびに出力では、選択対象のテキストコンテンツが静的テキストとして保持されます。
- ノードタグは、必要なパス(つまりそのノードを含む祖先ノードタグ)とともに挿入されます。デザイン内のコンテキストノードにより、絶対パスまたは相対パスが作成されます。
- テーブルならびにリストの選択対象からノードを作成する方法については、<u>要素/属性としてテーブルやノストを作成</u>のセクタンを参照付さい。

スキーマ内のノードを追加、削除する

(HTML 選択対象をベース・ユしていない)その他のノードを、ユーザー定義スキーマへ追加することができます。ノードを右クトゾして、コンテキストメニューから目的のコマイを選択してください。スキーマソノーサイドバーのソール・・アイコンからも、コマイを選択することができます。

ノードを削除するコよ、ノードが選択されている状態で、コンテキストメニューまけまソール・デアイコンから削除コマンドを選択します。ノードが削除されることで、既にある。父が不正となってしまう場合もある点に注意してください。

追加機能 HTML のインポート 353

デザインの修正

コンポーネントをドラッグして静的ならびに動的コンポーネントを挿入することで、デザインの構造を修正することができます。StyleVision に用意されているスタイル機能を使用することで、スタイルの修正を行うこともできます。

- □ 関連項目
- ユーザー定義スキーマ

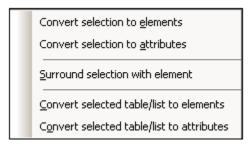
9.6.3 要素/属性としてテーブルやリストを作成

HTMLドキュメント内にあるテーブルならび、コノストをXML スキーマの要素まりは属性へ変換することで、テーブルまりはノストの構造をスキーマ上に保つことができます。

テーブルを要素/属性へ変換する

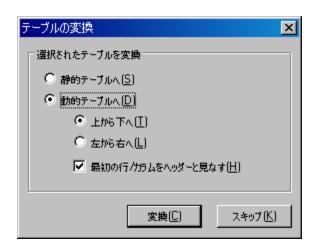
以下の操作により、テーブルをスキーマノードへ変換することができます:

- 1. HTML テーブル内部のテキストをハイライトしてテーブルを選択します。
- 2. スキーマツリー内にあるノードまで選択対象をドラッグして、兄弟または子ノードとして作成します。
- 3. マウスカーソルが兄弟ノードの作成シンボル
 → または子ノードの作成シンボル
 ¬→ になった時にノードをドロップします。
- 4. コンテキストメニューが表示されるので、選択されたテーブル/リストを要素に変換ませば選択されたテーブル/リストを属性に変換コマンドを選択します。



5. 表示される変換ダイアログにて、SPS 内に作成されるのか静的テーブルませょ動的テーブルなのかを選択します(以下のスグノーシショットを参照)。

HTML のインポート 354 追加機能



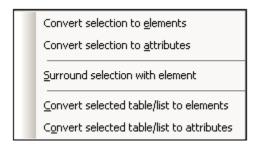
静的テーブルオプションが選択された場合、テーブルの各セルに対してスキーマノードが作成されます。 デザインでは、各ノードにコンテンツプレースホルダが挿入されます。テーブルセル内のデータは、一時的な内部 XML ドキュメント(そして生成された XML ドキュメント)へコピーされます。動的テーブルオプション は、テーブル内にある全ての行の構造が同一である場合に選択することができます。ノードが SPS にて 作成されると、動的テーブルにある行は(全ての行の構造が同一であるため)1つの行により表示されます。テーブルデータはXMLファイルへコピーされます。動的テーブルは上から下方向(垂直方向に行が表示されます)または左から右方向(行のデータがカラムとして表示されます)に展開することができます。最初の行/カラムをヘッダーとして指定すると、カラムヘッダーを含む最初の行が静的テキストとしてデザインに追加され、(ii)スキーマ要素/属性ノードの名前にはヘッダーが使用されます。最初の行/カラムがヘッダーとして指定されなかった場合、ヘッダー行がデザインに含まれることはありません。

6. オプションを選択した後には、変換をクリックして完了します。

リストを要素/属性へ変換する

以下の操作により、リストをスキーマノードへ変換することができます:

- 1. HTML リスト内部のテキストをハイライトしてリストを選択します。
- 2. スキーマツリー内にあるノードまで選択対象をドラッグして、兄弟または子ノードとして作成します。
- 3. マウスカーソルが兄弟ノードの作成シンボル → または子ノードの作成シンボル → になった時にノードをドロッ プします。
- 4. コンテキストメニューカ装示されるので、選択されたテーブル/リストを要素に変換ませば選択されたテーブル/リスト を属性に変換コマイを選択します。



5. 表示される変換ダイアログにて、SPS 内に作成されるリストが静的リストまけは動的リストになるかを選択します(以下のスグリーンション トを参照)。



静的リストオプションが選択された場合、各リストアイテムに対してスキーマノードが作成されます。デザインでは、アイテムリストの静的コンテンツとして HTML リストアイテムのテキストが各ノードに挿入されます。動的リストオプションが選択された場合、各リストアイテムがそれぞれ単一のリストアイテムノードとしてデザインに表示され、各リストアイテム要素にはコンテンツプレースホルダが挿入されます。 6. オプンエを選択した後口は、変換をケルケルで完了します。

□ 関連項目

- テーブルとの作業: SPS にてテーブルを使用する方法について記述されます。
- リストの作成

出力の生成 9.6.4

SPS を作成した後には、「ファイル | 生成されたファイルの保存」コマイにより以下の出力ファイルを生成することができます:

- 生成されたユーザー定義スキーマ:スキーマツリーサイドバーにて作成したスキーマのことです。
- 生成されたユーザー定義 XML データ: 作成したスキーマをベースに、インポートされた HTML から得られたデー タを含む XML ドキュメントのことです。
- ◆ HTML 出力の生成に使用される XSLT スタイルシート。
- HTML 出力。

9.7 ASPX インターフェイスによる Web アプリケーション

DB おはXML データのHTML レポーをSPS により作成する場合、通常 StyleVision では以下のようユポートの作成が行われます:

- 1. ソースデータが DB の場合、完成された SPS が StyleVision にてアクティブになっている状態で、DB から XML ファイルを生成します(ソースデータが XML ファイルの場合、このステップは必要ありません)。
- 2. SPS から、XSLT-for-HTML ファイルを生成します。
- 3. 生成された HTML ファイルをサーバー上に配置します。

ウェブアプリケーションの場合、ソース(DB まけまXML)データが修正されることで HTML ファイルが古くなることがあります。 ウェブサーバー上の HTML ファイルを更新する コよ (i) (DB をベースとしてデータの場合) XML ファイルを再度生成し、(ii) 新たな XML ファイルを XSLT-for-HTML により変換し、(iii) 生成された HTML ファイルをサーバー上へ配置する必要があります。

StyleVision ではHTML ウェブページを素早く更新するナかの方法が提供されます。この機能によりASPX アプリケーションを自動的に 生成することができ、ASPX アプリケーションファイルコ必要な全てのファイル(aspx ファイル XSLT ファイル コードファイルが StyleVision により生成されます。これらのファイルをノース DB ファイルまけは XML ファイル そして XSLT-for-HTML とともにサーバー上 に配置することで、ウェブインターフェースファイルの aspx ファイルが更新される度に、(i) (DB をベースとしたデータの場合) DB から新たな XML ファイルが生成され、(ii) サーバー上にある XSLT-for-HTML ファイルにより XML ファイルが変換され、(iii) 変換の出力がケェブインターフェースページに表示されます。これにより、最新の DB まけは XML データがウェブインターフェースページに表示されます。

ASPX ソリューションのファイルを生成する

DB ベースのSPS まけはXML ベースのSPS が作成された後には、以下の操作によりASPX ソリューションを作成することができます:

- 1. StyleVision にて SPS ファイルがアクティブになっている場合、メニューコマンドの「ファイル | ウェブデザイン | ASPX ウェブアプリケーションを生成」を選択することで、ASPX ファイルを生成することができます。 ASPX アプリケーションファイルが指定されたフォルダーに作成されます。 ASPX アプリケーションが生成されたフォルダーには、少なくとも以下のファイルが含まれます:
 - Readme.doc
 - SPSFilename.aspx
 - SPSFilename.xslt
 - SPSFilename.cs
- 2. DB ファイルまたは XML ファイルを、ASPX アプリケーションが配置されているサーバー上のフォルダーに配置します。.aspx ファイルがアプリケーションのエントリーポイントになります。このファイルが再度読み込まれた場合、表示される DB または XML のデータも更新されます。
- メモ XSLT 変換を正しく行うには、<u>Altova の Altova XML アプリケーション</u> をインストールする必要が決ます。変換を行う際に 問題が発生する場合、Altova XML のセナアップ方オフが記述されている ReadMe.doc ファイルを参照ください。

動作方法

ASPX アプリケーションを生成したフォルダーコよ、その他のファイルとともコ以下のファイルか含まれます:

- Readme.doc
- SPSFilename.aspx
- SPSFilename.xslt
- SPSFilename.cs

SPSFilename.aspx が出力ドキュメトのURL どいます。SPSFilename.aspx により、SPSFilename.cs. に含まれている C# コードが実行され、この C# コードにより、ファイル、ませま DB から取得された) XML コンテンツが読み取られ、SPSFilename.xslt ファイルともに Altova XML へ渡されます(Altova XML は Altova の XSLT 変換エンジンで、Altova ウェブサイトからダウィー・ドすることができます)。

Altova XML により、与えられた XSLT ファイルを使った XML コンテンツの変換が行われます。変換によりHTML ドキュメントが作成され、ウェブアプリケーションを介してブラウザーに表示されます。例えばDB に対する変更などにより XML のエンテンツが変更されると、SPSFilename.aspx の表示(まざはブラウザー内に表示されているページの更新)を行うことで、最新のデータがデータベースや XML ファイルから取得され、ドキュメントのレンダレグが行われます。

9.7.1 サンプル: Windows 7 にて動作する localhost

以下に記述される操作手順により、ローカルホストにASPX アプリケーションをセナアップします。詳細については、ASPX アプリケーションフォルダースある Readme.doc ファイルを参照ください。このフォルダーならパニファイルは SPS ファイルがアクティブな状態でメニューオプションからファイル | ウェブデザイン | ASPX ウェブアプリケーションを生成」コマイドを選択することにより生成されます。

RaptorXML のインストール

最新バージョンのAltovaXMLがインストールされていることを確認してくけさい。AltovaXMLはAltovaの変換エンジンで、(DB より生成された)XMLファイルの変換に使用されます。チャートの生成というた。StyleVision に搭載された機能の、くつかは、AltovaXMLのバージョンによってはサポートされないという点に注意してください。

Internet Information Services (Microsoft Web サーバー) の有効化

Internet Information Service (IIS) が有効化されていない場合、以下の操作を行なってください。ステップ 5 により、IIS が有効になっているが確認することができます。

- 1. Windows のスタートボタンから「コントロールパネル | プログラムと機能 | Windows の機能の有効化または無効化」を選択します。
- 2. インターネットインフォメーションサービスのチェックボックスをクリックします。チェックボックスが部分的に有効となります。
- 3. インターネットインフォメーションサービス以下にある「World Wide Web サービス | アプリケーション開発機能 | ASP.NET」のチェックボックスにチェックを入れます。
- 4. 「OK」をクリックします。処理が完了すると、C.¥inetpub¥wwwroot という名前のフォルダーが作成されます。これがウェブサーバーのルートフォルダーになります。
- 5. ブラウザーから localhost ヘアクセスすることで、IIS のウェルカムスクリーンを確認することができます。

Style Vision にてASPX アプリケーションを生成する

以下の方法によりASPX アプリケーションを生成することができます:

- データベースならびに SPS ファイルが同一のフォルダーにあることが推奨されます。
- 2. SPS ファイルが作成された後に、メニューオプションから「ファイル | ウェブデザイン | ASPX ウェブアプリケーションを生成」を選択します。
- 3. 表示されるダイアログにて、C:¥inetpub¥wwwroot 以下にフォルダーを作成し、そのフォルダーを選択します(例: C: ¥inetpub¥wwwroot¥Test1)。
- 4. フォルダーの選択を確定すると、そのフォルダーに〈FileName〉.aspx、〈FileName〉_AltovaDataBaseExtractor.cs、Web.config ファイルが生成されます。

メモ C:/inetpub/wwwroot フォルダー以下にファイルを保存する場合、管理者として StyleVision を起動する必要があります。StyleVision を終了して、StyleVision のアイコンを右クリックして、管理者として実行を選択してください。

生成されたアプリケーションとASPX を関連付ける

以下の操作により、StyleVision により生成されたアプリケーションが、ASPX によ認識されるようにはます:

- 1. Windows のスタートポタンからコントロールパネル | 管理ツール | インターネットインフォメーションサービス (IIS) マネージャー」を選択します。
- 2. 接続パネルにてフォルダーを展開します(例: Test1)。この時点でのフォルダーアイコンは、標準的な黄色で表 示されます。
- 3. フォルダーのコンテキストメニューから「アプリケーションへの変換」を選択します。ダイアログにて「OK」 をク リックします。フォルダーのアイコンが球形アイコンになります。 4. 接続パネルにあるツリーを展開して、アプリケーションプールを表示し、そのアイテムを選択します。
- 5.接続ペイン内にあるアプリケーションプールアイテムを選択することで表示される DefaultAppPool のコンテキス トメニューから「詳細設定」を選択します。
- 6. ID プロパティからカスタムアカウントを選択し、Windows ユーザー名とパスワードを入力します。
- 7. 32ビットアプリケーションの有効化プロパティの値を True にセットします(これによりデータベースドライバー がアクセスできるようになります)。このステップは64ビットの Windows を使用している場合にだけ必要になりま す。

アプリケーションを実行する

ブラウザーこて localhost/Test1/<FileName>.aspx ヘアクセスします(ASPX アプリケーションが保存されたフォルダー名がTest1 で、生成 元のSPS ファイル名が〈FileName〉.sps の場合)。変換されたHTML がジラウザーに表示されます。 ブラウザーに表示されている ASPX ページを再ロードすることで、データベースやXML内にある最新のデータからページが再度生成されます。

旡 ブラウザーが応答待ち状態になるようであれば、AltovaXML ライセンスファイルが取得されていることを確 認してください。

PXF ファイル: SPS とその関連ファイルのコンテナー 9.8

XSLT 2.0 を使用するSPS デザインはPortable XML Form (PXF) ファイルとして保存することができます。PXF フォーマナトは SPS デザインを(スキーマファイルやソース XML ファイル デザインで使用されるイメージファイル ソース XML を出力フォーマナトへ変換する XSLT ファイルといか 関連するファイルとともつ シケージ化するかな、Altova により開発されました。PXF ファイルフォーマトを使用する ことことは、Authentic View の編集や、Authentic View から出力を生成するためつ必要な全てのファイルを、1つのファイルとて配布す ることが可能になります。

このセクションではPXF ファイルの使用方法について、2つのパートに分けて説明します:

- PXF ファイルの作成
- PXF ファイルの編集
- PXF ファイルのデプロイ

土

PXF 機能は関する以下の点に注意してくたさい

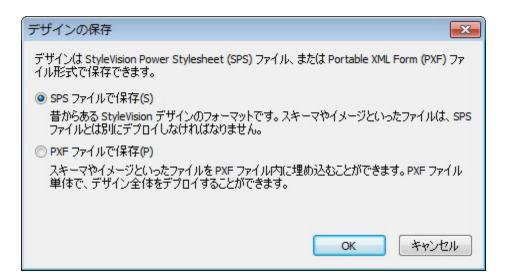
- XSLT 2.0 封は3.0. によりデザインされた SPS だけがサポートされます。
- XBRL ならびに DB XML カラムスキーマソースはサポートされません。これらのスキーマソースが含まれる 場合、保存形式の選択を行うダイアログは表示されません。PXF ファイルを編集するにあたり、これらのスキーマソースを追加することはできません。

□ 関連項目

- 使用方法の概要
- 名前を付けて保存

PXF ファイルの作成 9.8.1

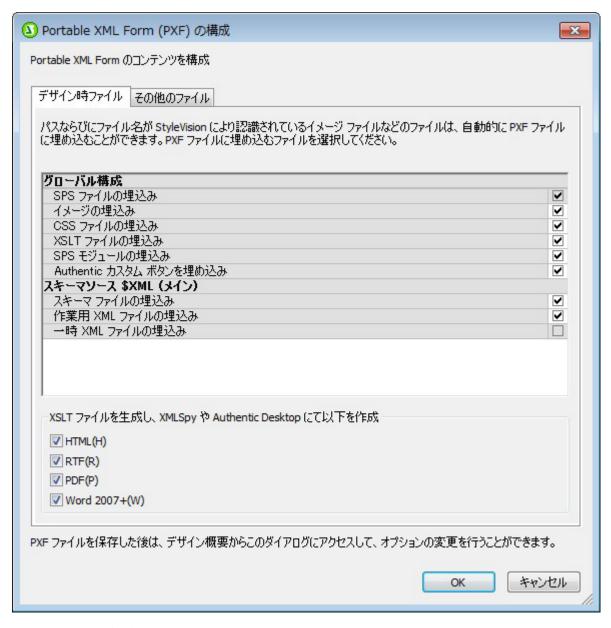
SPS デザインと関連するファイルを含むPXF ファイルを作成するコよ、StyleVision にてSPS デザインを開き、メニューオブンコンから 「ファイル | 名前を付けて保存」を選択します。デザインの保存ダイアログが表示されます(以下のスクリーンショナを参照)。



SPS フォーマットはStyleVision デザインに使用される標準のAltova フォーマナです。このセクションではPXF フォーマナに関する説明を行うさな、SPS フォーマナについては説明しません。SPS としてファイルを保存する場合、ユーザーレファレンスのセクションにある詳細を参照ください。

PXF として保存する

PXF オプションを選択することにより、Windows システム標準の名前を付けて保存ダイアログが表示されます。ファイルの保存は、デザインを保存、コマイと同様の方法に加え、PXF ファイルに追加するファイルを選択するためのステップが加わります。PXF ファイル名を指定すると、PXF の構成ダイアログが表示され、埋め込みを行うファイルを選択することができます(以下のスグリーシンコルを参照)。



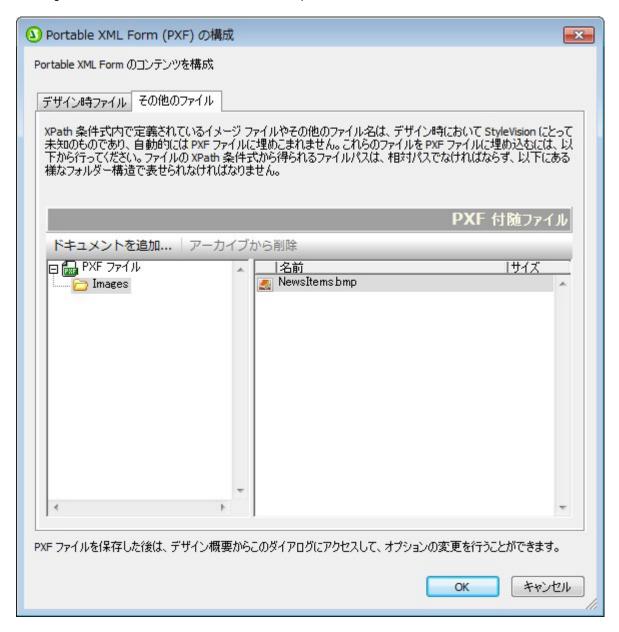
デザイン時ファイルタブのグローハ シレ構成セグションでは、デザイノニ関連するノースファイルの埋め込みを選択/選択解除することができます。更 に、デザインから生成されたXSLT ファイルを埋め込むかを選択することもできます。XSLT ファイルペイノにて、埋め込まれたXSLT ファイル か生成する出力フォーマナを選択してください。XSLT ファイルがPXF ファイルに追加され、そのPXF ファイルがAltova 製品の Authentic View にて開かれた場合、出力フォーマトを生成と閲覧を行うさめのソールド・オダンがAuthentic View にて有効になりま す(以下のスクノーンショナを参照)。



火モ StyleVision の高い、デジンでのみサポーされている(*高から低い順: Enterprise、Professional、Basic*)出力のため のXSLT ファイルは、PXF ファイル内で作成され、PXF ファイルが氏し、ドージョンで開かれる場合、PXF ファイルを保存す ると、低い、デジンでサポートされる出力のためのXSLT ファイルよ、保存されません。 PXF ファイルの保存を継続するかを問う プロンプトが表示、まかは、保存を中断して、サポートされないフォーマルを保管します。

その他のファイルタブでは、デザインコは直接含まれていないファイルを指定することができます(以下のスクリーンショナを参照)。例えば、デ

ザイン内部のXPath 条件式により生成されたURLにて参照されるイメージファイルなどを含めることができます。以下のスクリーシンョナでは、Images フォルダー以下にあるイメージのNewsItems.bmp がPXF ファイルに含まれるように指定されています。



その他のファイルをPXF ファイルに追加するコよドキュメントを追加ボダンをクトグして目的のファイルを選択します。ファイルを開くダイアログが表示され、SPS ファイル的配置されているフォルダーが開かれます。このフォルダー内にあるファイルませまその子孫フォルダー内にあるファイルを選択することができます。ファイルがPXF ファイルへ追加されると、そのファイルと対応するフォルダー構造が表示されます。上のアグレーシショナからは、SPS ファイルが配置されているフォルダーにある Images という名前のフォルダー内に News Items.bmp というファイルがあることが理解できます。

SPS ファイルが配置されているフォルダーの上位にあるフォルダーからファイルが選択されると、エラーが返されます。

SPS デザインでは、その他のファイルコオする参照を相対パスで、更に付属ファイルのペイノに示されているとおりに記述する必要があります。 例えば、上のスクリーンショナにある News Items.bmp には、Images/News Items.bmp とり相対パスで参照を行う必要があります。

- □ 関連項目
- 使用方法の概要
- 名前を付けて保存

9.8.2 PXF ファイルの編集

PXF ファイルはメニューオプションの「ファイル」 開く コマンドにより、StyleVision 内で開き、編集を行うことができます。編集には2種類の 方法があます

- PXF ファイルの構成を編集する。
- ◆ SPS や Authentic XML といった個々のファイル内にあるコンテンツを StyleVision にて編集する(イメージや CSS ファイルといったその他のコンポーネントファイルは外部アプリケーションにて編集することができます)。コ ンポーネントファイルは全て Style Vision にて明示的に更新する必要があります。

PXF の編集を行うエントリーポイント

PXF 構成の編集やPXF ファイルの更新を行うには、デザイン概要サイドバーにあるPXF アイテムを選択します(以下のスクリーシショナ を参照)。

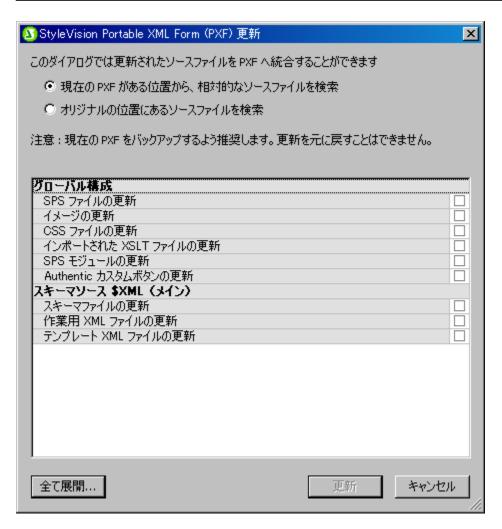


埋め込まれたファイルの構成

デザル概要サイバーはる埋め込まれたファイルの構成リンクをクリックすると、Portable XML Form (PXF)の構成ダイアログ が表示されます。使用することのできるオプションは、PXF ファイルの作成を行う際に使用するものと同じものです。

埋め込まれたファイルの更新

デザル概要サイバーにある埋め込まれたファイルの更新リンクをクナックすると、Portable XML Form (PXF) の更新ダイアログ か表示されます(以下のスクノーンショナを参照)。



ソースファイルの取得を、現在 PXF ファイルが配置されている場所からの相対的な位置により取得するのか、まけまオリジナルの場所に従い 行うのかを指定することができます。更に、更新するファイルのチェックボックスにチェックを入れて、更新ポタンをグリックすることでファイルの更新を行うことができます。新たな PXF ファイルが作成され、既存のPXF ファイルが上書きされます。そのため、更新を行う前には、オリジナルのPXF ファイルをバックアップすることが強く推奨されます。

□ 関連項目

- 使用方法の概要
- 名前を付けて保存

9.8.3 PXF ファイルのデプロイ

PXF ファイルが作成されると、通常のデータファイルと同じような方法で移動、ダウンロード、コピー、そして保存を行うことができます。PXFファイルコは Authentic View にて必要な全てのXML ファイルを含めることができ、出力ファイルを生成することができるさめ、このファイルだけで Authentic ユーザーは出力を生成することが可能になります。

PXF ファイルは Altova 製品のAuthentic View 「こて開く事ができます。XMLSpy 「こて PXF をどのように使用することができるかの使用シナルを以下に示します::

- ▶ メニューオプションの「ファイル | 開く」コマンドにより PXF ファイルを開きます。埋め込まれた XML ファイルが埋 め込まれた SPS ファイルにより Authentic View で表示され、Authentic View にて編集することができます。メ ニューオプションの「ファイル|デザインを保存」コマンドにより、(内部に埋め込まれた XML ファイルが更新さ れた)PXF ファイル内部の変更が保存されます。
- XML ファイルが埋め込まれていない PXF ファイルを「ファイル | 開く」コマンドにより開きます。 XML ファイルが含 まれていない場合、SPS デザインをベースにしたテンプレート XML ファイルが Authentic View にて開かれます。 メニューオプションの「ファイル | デザインを保存」コマンドを選択することで、この XML ファイルは PXF ファイル 内に埋め込まれたファイルとして保存されます。
- Altova 製品の XMLSpy では XML ファイルを PXF ファイルと関連付けることができ、PXF ファイルに埋め込まれ た SPS ファイルを Authentic View の編集に使用することができます。 関連付けは、メニューオプションの 「Authentic | SPS の割り当て」コマンドにより、このような処理を行うことができます。変更が保存されると、こ れら変更点は XML ファイルに対して保存され、PXF ファイルは変更されないままとなります。
- 出力フォーマットの生成に使用される XSLT スタイルシートが PXF ファイル内に埋め込まれた場合、Authentic View ユーザーはそのフォーマットにて出力を生成することができるようになります。ドキュメントの出力は出力生 成ツールバーボタンから行うことができます(以下のスクリーンショットを参照)。Authentic View にて表示される 出力生成ツールバーのボタンは、その出力に対応する XSLT ファイルが、PXF ファイル内部に含まれる場合だ け有効となります。例えば、HTML と PDF に対する XSLT スタイルシートだけが PXF ファイルに含まれている場 合、HTML ならびに PDF 出力のツールバーボタンだけが選択可能となり、その他の(RTF ならびに DocX)出力 ボタンは選択不可能状態で表示されます。



旡 PXF ファイルがウェブサーバー上に配置されており、Authentic Browser プラグインにより使用される場 合、サーバーにてそのファイルがブロックされていないことを確認する必要があります。PXF (.pxf) ファイル に対して(例えば IIS 管理パネルから)以下のような MIME タイプを追加することで対応できます: application/x-zip-compressed.

- □ 関連項目
- 使用方法の概要
- 名前を付けて保存

10 処理の自動化

StyleVision にお生成されたXSLT ファイルと出力ファイルを StyleVision の機能と組み合わせることにより、パフフルな自動化を行うことが可能しています。このセクションでは自動化機能について説明します。

Style Vision のファイル生成機能

StyleVision によりSPS デザインが作成されると、使用している StyleVision のGUI から XSLT ならびに出力ファイルを生成することができるようによります。メニューコマンドの「ファイル」生成されたファイルの保存」により、以下のファイルを生成することができます:

- HTML 出力するナーめのXSLT ファイル。
- HTML 出力

上のノストから分かるように、StyleVision にて保存することのできるファイルコは2種類あります:

- 1. SPS のデザインにより生成される XSLT ファイル
- 2. 最終的な出力ファイル(例: HTML)

最終的な HTML 出力ファイルを生成するためのプロセスは1つのステップにより実行され、XSLT スタイルシートにより XML ドキュメントが出力フォーマットへ変換されます。

Style Vision Server とRaptorXML: GUI を伴われ ファイルの生成

XSLT スタイルシートと目的の出力ファイルを生成するづよ、StyleVision のGUI(「ファイル 生成されたファイルの保存」コマンド)の他にも、以下の2つの方法が用意されています:

- 1. GUI を開かずこStyleVisionのファイル生成機能を呼び出す StyleVision Server は、多種の出力を作成することができます。
- 2. RaptorXML は、Altova のXML(+XBRL) バリデーター、とXSLT とXQuery エンジンを含むスタイプロンの Altova アプリケーションです。RaptorXML 内のXSLT エンジンは、XSLT スタイルシートを使用して XML ドキュメートを処理し書式を出力するか XML の変換に使用することができます。XSLT ファイルを最初に作成すると、RaptorXML により使用されることができます。(RaptorXML は、SPS を入力・ラメーターとして受け取らない場合があります)RaptorXML の使用は以下の点で有利です。(i) 大きなファイルの速い変換の結果によりスピードが向上しました。(ii) コマイドライン インターフェイス コルえ、RaptorXML は、COM、Java、と、NET のためのインターフェイスを提供し、このため、これらの環境内からの呼び出しが簡単しないました。変換のための RaptorXML の使用方法は、RaptorXML のサブセクションで説明されています。
- 3. Altova FlowForce Server ワークフロー内でAltova Style Vision Server を使用して、複数の変換を事前に設定されたトリガーに近く例えば毎日定刻に、実行することができます。この点については、<u>FlowForce Server を使用した自動化</u>で詳しく説明されています。

10.1 コマンドライン インターフェイス

StyleVision 機能を以下の2つの方法で呼び出すことができます:

- StyleVision 実行可能ファイルを呼び出す。これにより StyleVision のXSLT ファイル生成機能にアクセスすることができます。 XSLT ファイルは SPS ファイルから生成されます。
- <u>StyleVision Server</u> を使用して出力ファイル(HTML、など) を生成することができます。関連したファイル(XML、XSD、イメージ、ファイル などの)出力ファイルが、パッケージされた SPS ファイルである PXF ファイルから生成されます。 PXF ファイルは、StyleVision から生成されます。

コマイラインの使用方法

コマイラインを使用するコ22種類の方法があります:

- コマンドラインにコマンドを直接入力し、直ちに実行します。例えばコマンドプロンプトにて Style Vision また はStyle Vision Server にコマンドを入力し、Enter キーを押下することで、コマンドが実行されます。
- 複数のコマドをバッチファイルに入力し、以下の例にあるようなバッチ処理を行います:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

バッチファイルが処理されるとコマンドが実行され、ファイルが生成されます。

スケジュール設定により実行される Style Vision

W indows のスケジューリタスクソールを使用することで、指定されたスケジュールコ従うかたちで Style Vision コマイド実行することができます。コマイドを直接入力、おけば実行するタスクとして、デチファイルを指定することができます。作成方法に関しては 自動処理の方法のセクションを参照くたさい。

10.1.1 StyleVision

コマドラインの構文は以下のようになります:

```
StyleVision [<SPS File>] [<options>]
```

where

StyleVisionStyleVision アプリケーションフォルダーご存在する StyleVision を呼び出します。<SPS File>SPS ファイルを指定する<options>下に以入されるオプションの1 つまたは複数のオプション

コマドカ実行されるとStyle Vision はGUI か開かれないなど)警告なしに作動し、必要とされる出力ファイルを生成し閉じられます。エラーまけは警告が発生すると、GUI か開かれ、メッセージボックスに対応するメッセージが表示されます。

メモ Style Vision 内で SPS が正確 コードされる Jま、SPS が使用する XSD と作業用 XML ファイルが SPS 内で指定されている場所に存在する必要があます。

オプション

オプションの入力の順字に決まりはありません。 FO、RTF、PDF、W ord 2007+ 出力に関連するオプションは、 Enterprise エディショ ン、ませよ Enterprise とProfessional エディションのみで使用することができます。これらのオプタンは 下の *Enterprise エディショ* ン、 おは Enterprise とProfessional エディションリスト内で説明されて、ます。

XSLT ファイル出力

-OutXSLT=<file> XSLT-for-HTML を指定されたファイルに書き込む

-OutXSLRTF=<file> XSLT-for-RTF を指定されたファイルに書き込む(Enterprise と

Professional エディション)

-OutXSLFO=<file> XSLT-for-FO を指定されたファイルに書き込む (Enterprise エディショ

XSLT-for-W ord 2007+ を指定されナファイルに書き込む

OutXSLWord2007=<file> (Enterprise エディンコンのみ)

サンプル

StyleVision "QuickStart.sps" -outxslt="QuickStartHTML.xslt" StyleVision "C:\Test\QuickStart.sps" -outxslt="C:\Test\QuickStartHTML.xslt"

以下の点に注意してくたさい:

- ハンスは絶対的、または、相対的でバックスラッシュを使用します。
- ファイル名、おけよノスペースか含まれる場合、ノス全体は引用符で囲まれる必要がおります。例: "c: \м▽ Files\MyXML.xml" or "c:\MyFiles\My XML.xml".
- コマンド、パスフォルダー、およびファイル名は大文字と小文字を区別します。

10.1.2 Style Vision Server

StyleVision Server を、Windows、Linux とMac OS システム上で、コマイドライン インターフェイス(CLI) 上で使用することがで き、XML ファイルを出力 HTML、PDF、RTF、とDOCX ドキュメントに変換します。 StyleVision Server CLI のgenerate コマ ンドは、XML ファイルとPXF ファイルを2つの引数として取り、希望する出力の書式をパラメーターとして取ります。変換のためのXSLT スタイルシートは、入力として提出されている PXF ファイルから取得されます。

RaptorXML Server のCLI ではく、StyleVision ServerのCLI を使用する利点は、StyleVision Server は、PXF ファイルを 入力とて取ることができる点です(RaptorXML はXSLT ファイルを入力としてとはす)。 Style Vision Server はAltova FlowForce ワーケフローの一部として使用されることが奨励されます。 FlowForce ワーケフローは 変換ジョブを設定済みのトリガーに従い 開始します: FlowForce ジョブがトリガーされるとや・ナワーク内で、複数のファイルは自動的に変換されます。詳細に関しては、 FlowForce Server を使用した自動化セクションを参照してくたさい。

StyleVision Server CLI に関する詳しい説明は、次を参照してください StyleVision Server ドキュメナ。

出力ファイル

StyleVision Server は、1つませば複数のファイルを指定されたPXF ファイルから生成することができます:

コマンドライン インターフェイス

• PXF内で指定されたXML とXSLT-for-HTML ファイル まけよ 代替のXML ファイルを使用したHTML(.html)ファイル

370 処理の自動化 RaptorXML の使用

10.2 RaptorXML の使用

Altova RaptorXML は、最新の標準と並列計算の標準のための最適化のために構築されたAltova の第三世代の超高速な XML と XBRL プロセッサーです。 XBRL processing is available only in RaptorXML+XBRL Server. 高度なクロスプラナフォームに対応し、エンジノは今日のマルチコアコンピューターに XML と XBRL データを処理するために高速な処理機能を提供します。

RaptorXML には以下の2つのエディションかあります:

- ネトワーク上でアクセスすることができ、複数のファイルを1度に変換することのできる RaptorXML Server エディション:
- ネトワーク上でアクセスすることができ、複数のファイルを1度に変換することができ、XBRL 検証をサポーする RaptorXML+XBRL Server エディンコン。

RaptorXML は関する詳しい説明は、次を参照してくたさい Altova Web サイト。

一般的な使用例

StyleVision ユーザーにと最も一般的な RaptorXML の使用例は、XSLT 変換機能です。この機能は以下を行うさめに通常使用されます:

- 1. 「ファイル 生成されたファイルの保存」コマドを用いて、XSLT スタイルシートはSPS から生成されます。SPS ファイルから XSLT スタイルシートを生成するためにRaptorXML を使用することはできないことに注意してください。
- 2. RaptorXML を使用して、XML ドキュメントを変換するかがこ生成された XSLT スタイルシートが使用されます。 RaptorXML を使用して HTML 出力を生成することができます。

RaptorXML の利点

RaptorXML を使用する利点を以下に示します:

- RaptorXML はよ 高速な検証とXSLT 変換が搭載されており、大きなファイルを処理するために役に立ちます。
- コマンドライン、COM、Java、NET インターフェースによる簡単な使用。
- バッチファイルや Windows のタスクスケジューラーにより、<u>処理の自動化やスケジュール処理</u>などを行うことができます。

XML ソースドキュメトから子出力ドキュメト(HTML などの生成を自動化するかとRaptorXML を使用する方法に関しては、次のセグシンを参照してください、自動処理の方法。

RaptorXML のCOM、Java、と.NET インターフェイスの使用を含むRaptorXML 口関する詳細は、次を参照してくたさい. RaptorXML ユーザードキュメト。

10.2.1 PDF 出力

XMLドキュメントからPDF 出力を生成するコは2つのステップを踏む必要がおります::

- 1. (RaptorXML などの) XSLT 変換エンジンにより XML ドキュメントの変換を行います。Altova XSLT エンジンといった XSLT 変換エンジンを使用することで変換を行うことができ、FO ドキュメントを生成することができます。
- 2. (Apache FOP などの) FOP といった FO プロセッサーにより FO ドキュメントを処理することで PDF 出力が 生成されます。XSLT 変換から得られた FO を FO プロセッサーへ渡すように StyleVision をセットアップ することで、PDF 生成の結果を StyleVision の PDF プレビューウィンドウに表示、または(「ファイル」生成さ れたファイルの保存」コマイによりファイルとして保存することができます。

処理の自動化 RaptorXML の使用 371

RaptorXML &PDF

RaptorXML は XSLT 変換に使用されるだけで、FO プロセッサーに対して FO 出力を渡すようなパラメーターは存在しません。そのため、XSLT 変換の(最初の)ステップにより FO ドキュメントが生成されるだけになります。

FO ドキュメントを FO プロセッサーに渡すことで FO から PDF への処理が行われます。このステップを実行するために使用される命令は使用するプロセッサーにより異なります。例えば、Apache FOP プロセッサーの場合、以下のコマンドにより入力 FO ドキュメントと出力 PDF ドキュメントの名前と場所が指定されます:

fop -fo input.fo -PDF output.pdf

FOP には他のパラメーターも用意されています。詳細については FOP ユーザーリファレンス を参照ください。

FOP **EXSLT**

FOP オプショイこより、入力 XML ファイル、入力 XSLT ファイル そして出力 PDF ファイルを同時に指定することができます:

fop -xml input.xml -xslt input.xslt -PDF output.pdf

この場合、FOP は、内蔵のXSLT エンジンを使用し、XML からFO への変換が最初に行われます。その後生成されたFO ドキュメトをFOP へ渡すことで、FO からPDF への変換処理が行われます。

しかしなから、FOP 内蔵のエンジンは、Style Vision とRaptor XML によりサポートされるすべてのXSLT 機能をサポートしない可能性があります。この結果、Style Vision により生成された XSLT スタイルシート がXML 変換により FOP の内蔵の XSLT を使用して指定されている場合、エラーが発生する可能性があります。Raptor XML + XBRL の XSLT エンジンを使用して、FO への変換を行い、FOファイルを FOP へ渡して PDF の生成を行うようにしてがさい。

PDF のが近理

XSLT 変換の最初のステップのためにRaptorXML を使用して、PDF を生成する素早、簡単な方法です。 2番目のステップである FOP を使用した、FO 処理は、2つのコマンドを組み合わせる、ジッチファイルを作成します。例:

raptorxmlserver xslt --input=Test.xml --output=Test.fo Test.xslt
fop -fo Test.fo -pdf Test.pdf

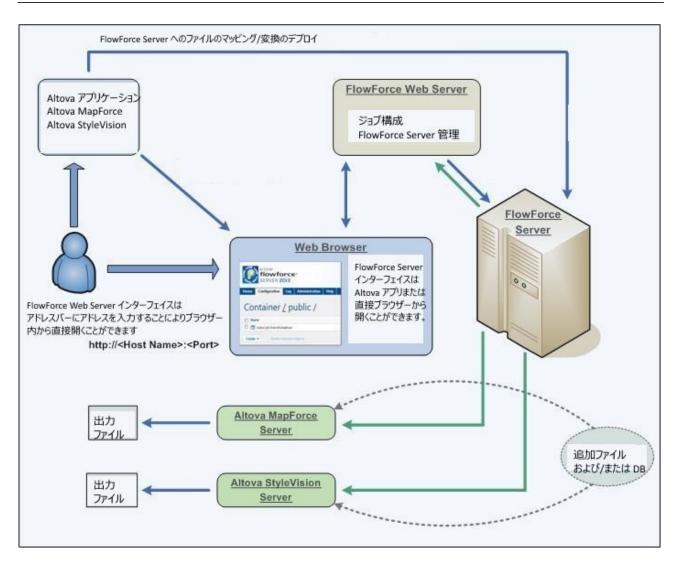
最初のコマンドにより RaptorXML が呼び出され、test.fo が出力として生成されます。2番目のコマドでは、test.fo が FOP プロセッサーへ渡され、PDF ファイルの test.pdf が作成されます。バッチ処理と、バッチファイルを使った処理の自動化については、次を参照してくたさい処理の自動化。

10.3 FlowForce Server を使用した自動化

W indows、Linux、とMac OS システム上で使用することのできる Altova の Flow Force Server を使用して、ネ・ナワーク上で変換を自動化することができます。 このプロセス お以下のように作動します:

- 1. (ファイル FlowForce ヘデプロイする コマドを使用して) StyleVision から、PXF ファイルは.transformation ファイルとて FlowForce Server にデプロイ されます。.transformation ファイルは、SPS (内でデザインされまとおり に変換を実行するかが2必要な全てのファイルと情報を含んで、ます。下の図では、デプロイは接続ラインにお表示されて、ます。)
- 2. .transformation ファイルがFlowForce Server にデプロイされた後、.transformation ファイルを使用して、ジョブ 定義内で指定されているトリガーには、変換を生成するジョブをFlowForce 内でジョブを作成することができます。(トリガー は、例えば、毎日定時にナリガーするように設定することができます。)(下のダイアグラムで表示されているように StyleVision、まけは、HTTP アドレスによりアクセスすることのできる Flow Force ジョブを FlowForce W eb Server インターフェイスで作成することができます。 FlowForce ジョブの作成に関する詳細は、次を参照してください FlowForce ドキュメント
- 3. 実行時にFlowForce Server は 変換命令と関連するファイルを変換を行う Style Vision Server に なします (下の図を 参照してくたさい)。

FlowForce ワークフロー内のStyleVision Server のや区割りは、下の図に表示されています(Altova MapForce マピングを実行の対対にAltova MapForce Server に転送するFlowForce ジョブも作成することができるが、ワークフロー内でのMapForce Server の役割は表示されています)。



FlowForce ジョブによい呼び出されない場合、StyleVision をコマイドラインから呼び出すことができます。使用方法は、StyleVision Server ドキュメト で説明されてます。

374 処理の自動化 自動処理の方法

10.4 自動処理の方法

バッチファイル(ファイル拡張子 .bat と共に保存されるファイル)コマイ・ラインから実行されるコマイ・のシーケンスを含んでいます。バッチファイルが実行されると、バッチファイル内の各コマイ・は、最初から始まが順番に処理されます。バッチファイルは次のシチュエーションで役に立ちます。

- 一連のコマイを自動的に実行する場合(下を参照)。
- コマイが前のコマイにより生成される入力を必要とする一連の処理コマイを作成する場合。(例えば、変換の出力として生成される XML ファイルが次の変換の入力として使用される場合)。
- 特定の時間にタスクのシーケンスが実行されることを予定する場合。

コマンドのシーケンスを持つバッチファイル

実行されるコマイのシーケンスは以下のように入力されます:

```
@ECHO OFF
CLS
StyleVision TestEN.sps -outxslt=HTML-EN.xslt
StyleVision TestDE.sps -outxslt=HTML-DE.xslt
StyleVision TestES.sps -outxslt=HTML-ES.xslt
```

バッチファイルが処理されると、コマンドは実行され、ファイルが生成されます。上記のバッチファイルは、Style Vision を使用して、3つの XSLT ファイルをSPS ファイルから生成します。

11 メニューコマンドとレファレンス

このセグションでは StyleVision ツール デ、デザイン ビューのシンボル、そしてメニュー コマンドについて説明します。セグションは以下のように構成されます:

- デザインビューで使用されているシンボルの説明。
- ◆ XPath 式 の編集ダイアログの説明。
- すべての<u>ツールドーとそれらのアイコン</u>、そしてソールドーの表示をカスタマイズする方法。
- すべてのメニューコマンド。

ユーザー・ノファレンスのセグションでは個々のロマンドに関する記述がなされている一方で、関連するセグションでは StyleVision の機能で使用されているメカニズムの詳細によって説明されます。メカニズムは以下のグループには構成されています:

- SPS ファイルコンテンツ
- SPS ファイル構造
- SPS ファイルの高度な機能
- SPS ファイル表示方法
- SPS ファイル追加機能

- 照関連項目

- ユ<u>ザーインターフェイス</u>
- チュートリアルのケリックスタート

11.1 デザイン ビューのシンボル

SPS のデザイノコは通常、複数種類のエレポーネトが含まれ、各コレポーネトはそれぞれ異なるシンボルでデザイン内に表示されます。これらのシンボルは以下のようにグループ分けされます。:

- XMLドキュメト内のノード
- XMLドキュメント コンテンツ
- データ入力デンドス
- 定義済みのフォーマット
- XPath オブジェクト
- URI オブジェケト

これらコンポーネントノス対して以下の操作を行うことができます:

- ドラッグ アンド ドロップにより移動する
- (i) 編集メニューのコマイ、まけま(ii) 標準的な W indows のショートカナにより切り取り、コピー、貼り付け、まけば削除する
- フォーマットを適用する
- 右クリックすることで、コンテキスト メニューを表示する

XMLドキュメント内のノード

XML ドキュメト内にある要素や属性は SPS デザインドキュメト内でタグとして表示されます。各ノードには開始タグと終了タグか管まれます。開始タグまけは終了タグをダブルクトックすることでそのソードが縮退され、ノードが縮退されると、ノード内部に含まれているすべてのエンテンツが隠されます。縮退されたノードをダブルクトックすることで、ノードが展開され、内部に含まれているコンテンツが表示されます。

ノード には以下の種類が存在します:

・ドキュメントノード



\$XML により示されるドキュメント ノードは XML ドキュメント全体を表します。スキーマ ソースが XML ドキュメントに関連付けられている場合には緑色の \$XML タグが表示され、スキーマ ソースがデータベースに関連付けられている場合には黄色の \$DB タグが表示されます。上のスクリーンショットでは、OrgChart を含むドキュメント ノードが左側に、右側には縮退されたドキュメント ノードが示されています。.

要素ノード



挿入された場所に祖先ノードが存在しない場合、すべての祖先ノードとともに要素ノードが挿入されます。 上のスクリーンショットでは、展開 (左側) ならびに縮退 (右側) された Name という要素ノードが示されます。

属性ノード



挿入された場所に祖先ノードが存在しない場合、すべての祖先ノードとともに**属性ノード**が挿入されます。 属性ノードには = という頭文字が表示されます。上のスクリーンショットでは、href 属性ノードが展開され たかたち (左側) ならびに縮退されたかたち (右側) で示されます。

ノードはデザイン内にノードテンプレートとして含まれます。デザインに含むことのできる異なる種類のテンプレートに関しては、次のセグションを参照してください、テンプレートとデザインフラグメント・

XML ドキュメント コンテンツ

XML ドキュメント コンテンソは2 種類のプレースヤルダこより表示されます:

- (contents)
- (rest-of-contents)

contents プレースオリダは単一ノードのエンテングを表し、ノード内に含まれるすべてのテキスト エンテンツが出力されます。ノードが属性ノードまけはテキストのみの要素ノードの場合、ノードの値が出力されます。ノードが混合コンテンツや要素エンテンツが引力で構成される要素ノードの場合、子孫ノードのテキスト コンテンツが出力されます。XSLT の用語に置き換えると、contents プレースオリダは、select 属性の値によりノードが指定された。xsl:apply-templates 要素と等し、振る舞いを行います。

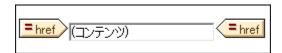
メモ contents プレースホルダが要素ノードに適用されると、その要素の属性に含まれる値は出力されません。属性ノードの値を出力するには、属性を(メインまたはグローバル)テンプレートに明示的に含める必要があります。

rest-of-contents プレース村/ダにより、カレント ノード以下にあるその他の子要素に対してテンプレートが適用されます。この場合、各子要素に対して適用されるテンプレートはグロー シレテンプレート(その要素に対して定義されている場合)、おけは要素のデフォルト テンプレート(テキストのみの要素がはおテキストだ」を出力し、子要素のテンプレートを適用)とないます。例えば、title、author、isbn、pubdate という子要素を含んだらbook という要素を考えてみましょう。book の定義によりtitle 子要素だけが出力されると指定された場合、この定義が処理されることで、その他の子要素(author、isbn、pubdate)「出力されないこと」ではます。しかし、book の定義内のtitle 要素直後にその他のコンテンツ プレースポルダが含まれている場合、その他の子要素(author、isbn、pubdate)、グロー・シルテンプレート(その要素に対する定義が存在する場合)おけは要素のデフォルト・テンプレートも適用されることがます。

データ入力デバイス

デザイノニてデータ入力デッドスを配置することで、Authentic View ユーザーによる XML ドキュメントの正確な編集と、妥当なデータの入力がサポートされます。ノードロコ以下のデータ入力デッドスを割り当てることができます。

スカフィール(単一行と複数行)



コンボボックス



チェックボックス



ラジオボタン



展開されたタグはダブルクトックすることで縮退することができますし、縮退されているタグも、ダブルクトックにより展開することができます。データ 入力ディイスの細かな使用方法については、データ入力ディイスを参照くたさい。

定義済みのフォーマト

定義済みフォーマナは薄い紫色のタグで表示され、ダブルクトックにより展開/縮退することができます。





上のスケノーンショナでは、定義済みのフォーマナ p (para) が展開された状態 (左側) と縮退された状態 (右側) で示されています。定義済みのフォーマナを適用するコよ 目的のアイテムを (そのアイテムをクトックするかテキストをマーキングすることで) ハイライトして、 定義済みのフォーマナを挿入します。。

XPath オブジェクト

StyleVision ではXPath 条件式を使用するためこ2 つのめカニズムが用意されています。

条件テンプレート



条件タグは青色により表示されます。開始タグにはセルが含まれており、左端のセルには疑問符が含まれています。その他のセルには (i) 個々の when 条件に対応した 1 からはじまる数値か、(ii) その他の条件を表すオプションのアスタリスクが表示されます。条件分岐はクリックにより選択でき、選択された条件分岐の番号が開始タグにてハイライトされ、その分岐に対応するテンプレートが条件の開始タグと終了タグの間に表示されます。選択された条件分岐の XPath 条件式もデザイン ツリーでハイライトされます。条件のタグを縮退/展開することはできません。

自動計算

Office Summary: =(自動計算)

自動計算はデザイン ビューにて =(自動計算) というオブジェクトにて表示されます (上のスクリーンショット を参照)。選択された自動計算の XPath 条件式がデザイン ツリーにてハイライトされ、プロパティ サイドバ 一からプログライサイドバー自動計算の編集をおごなうダイアログヘアクセスすることができます。

URI オブジェクト

デザイン内部へ挿入することのできるURI をベースコンナナブジェクトには、以下の3 種類がかます:

- イメージ
 - StyleVision からアクセスできるイメージが SPS デザインへ挿入されると、イメージがデザイン ビューに表 示されます。イメージへアクセスできない場合、SPS はイメージのプレースホルダによりマークされます。
- ブックマーク(アンカー)



ブックマークは A という文字の入った黄色のタグにより表示されます(上のスクリーンショットを参照)。メ ニューから [挿入 | ブックマークの挿入] を選択することで空のブックマークが追加されるほか、コンテンツが選択された状態で [タグで囲む | ブックマーク] のメニュー オプションを選択することで、ブックマーク を挿入することができます。アンカー タグは展開 (左側のスクリーンショット) ならびに縮退 (右側のスクリーンショット) することができます。

リンク



ブックマークは A という文字の入った黄色のタグにより表示されます(上のスクリーンショットを参照)。メ ニューから [挿入 | ブックマークの挿入] を選択することで空のブックマークが追加されるほか、コンテンツが選択された状態で [タグで囲む | ブックマーク] のメニュー オプションを選択することで、ブックマーク を挿入することができます。アンカー タグは展開 (左側のスクリーンショット) ならびに縮退 (右側のスクリ ーンショット) することができます。

- 関連項目
- ツールドー
- デザインサイドバー
- コンテンソ編集処理

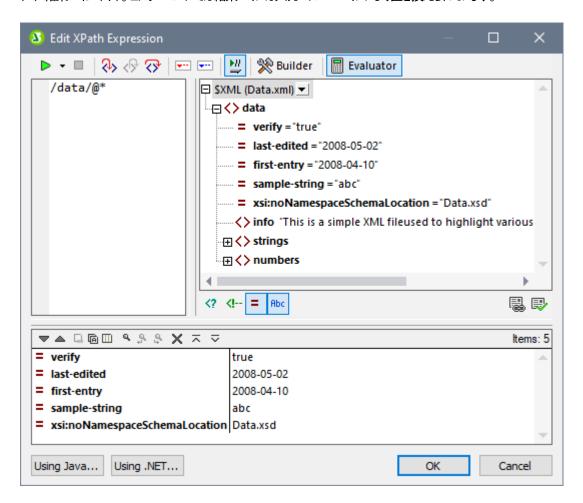
11.2 XPath 式 の編集ダイアログ

XPath 式 の編集ダイアログ (下のスクリーンショナ) はXPath 式 の編集を作成、テストするために使用されます。 条件付きの処理 または スタイル と プロン ディの値のために式を入力する場合など XPath 式が入力されるデザインビュー内のすべての箇所で使用することができます。

ダイアログは SPS (XSLT 1.0 のためのXPath 1.0、XSLT 2.0 のためのXPath 2.0、XSLT 3.0 のためのXPath 3.1) の XSLT バージョン を自動的にサポートします。 XPath バージョンを切り替える コよ SPS の XSLT バージョンを切り替えます。

ダイアログのレイアウト

XPath 式 の編集には以下のペインか含まれています (下のスクリーンショナを参照): (i) 式ペイン (左上)(ii) ソースペイン (右上) (iii) 結果ペイン (下)。 ビルダーモードでは結果ペインは入力ヘルルトーペインにおっ置き換えられています。

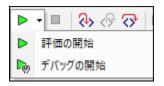


評価モードとデバッグモード

XPath 式 の編集ダイアログ には2つのモードが存在します:

- XPath 式が割り当てられた作業 XML ファイルコ対して評価される<u>評価モード。</u>式ペイン に式は入力され、結果は結果ペイン に表示されます。結果内のソードをクトックしてダイアログのソースペイン内のソードに移動することができます。
- 割り当てられた作業 XML ファイルコ適用し XPath 式をデッジグすることのできるデッシグモード。ブレークポイントとレースポイントを設定し、評価をステップ・イステップで行うことができます。各ステップで変数のエレテンクを確認することができ、評価の追加アスペクトをチェックすることのできるカスタムのフォッチ式を設定することができます。で

ウイドウのソール ドーの左角にある「評価/デバッグの開始」ドロップダウンメニュー内から適切なコマンドを選択し、2つのモード間を切り替えることができます(下のスクリーンショナを参照)



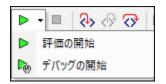
2つのモードの使用方法はこのセクションのサブセクション内で説明されています。

XPath/XQuery 式ビルダー

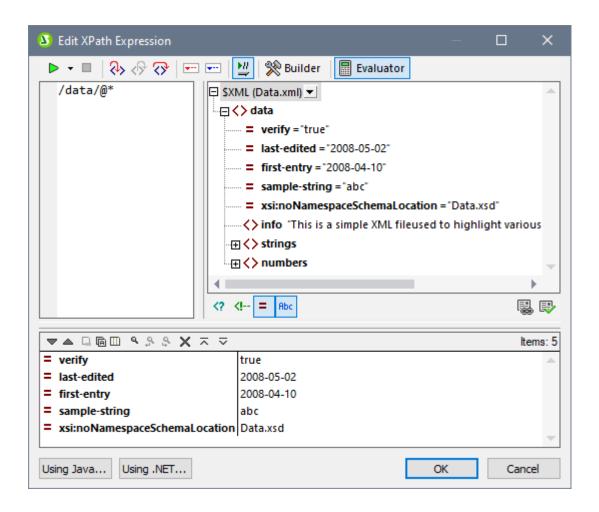
両方のモード内で、<u>式ビルダー</u>を使用して構文的に正しい式を作成することができます。メインソール デージンの「ビルダーモード」 ボタンを使用して式ビルダーを切り替えます。

11.2.1 エバリュエーター

「評価/デバッグの開始」ドロップダウンメニュー内で「評価の開始」を選択し評価モードを選択します(下のスクリーンショナを参照)。



評価モード内でエバリュエーター ボダンをクトックします (*下のスクリーンショナトを参照*)。エバリュエーターコよ次のペインが存在します *(下の スクリーンショナトを参照)*。(i) 式ペイン *(左上)*(ii)ソースペイン *(右上)*(iii) 結果ペイン *(下)*。



XPath 式とその評価

XPath 式は式ペイン内に入力されます。結果ペイン内に評価の結果は表示されます(上のスクリーンショナ)。

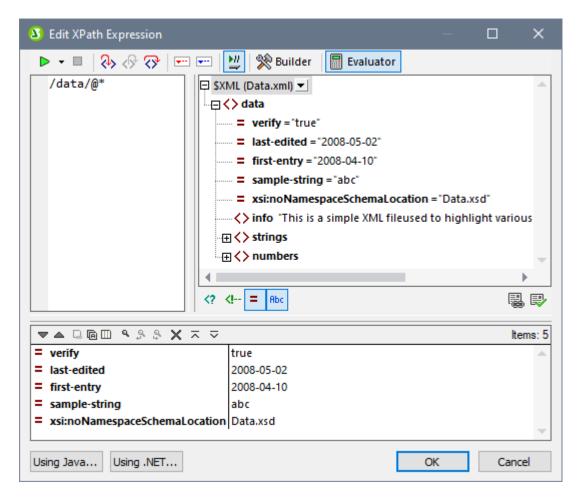
以下の点に注意してくたさい

- XML ファイルス対して式を評価するコよ そのファイルは <u>SPS のノースの一つのファイルと作業</u> として割り当てられる必要があます
- (入力中に評価」iアイコンをソール デーから選択すると), 結果は式の入力中に表示されます。 おはよ (ツール デーの左上の)「評価/ディッグの開始」(F5) をクトックすると表示されます。
- ソースソノー内のソードのXPath ロケーター式を入力するコおノースペイン内のソードをダブルクトックします。
- ソースペイン内では、以下の表示を切り替えることができます: (i) 処理命令 (ii) コメント (iii) 属性 (iv) 要素。ソースペインの下のがなっを使用してこれを行ってくたさい。
- コンテキストノードは式が作成されるデザインコンポーネントを指します。式のコンテキストノードとして他のノードを設定するコお以下を行います。(i) ソースペイン内のノードを選択します(ii) (ソースペインの下にある)評価コンテキストをかかりします。このコンテキストノードを式のナめに保存するコは、(ソースペインの下にある)「評価コンテキストを記憶する」コマンドボタンを切り潜えてください。式のナめに実際のコンテキストノードは現在のデザインコンポーネントのコンテキストノードにない、ランタイムで使用されるコンテキストノードになります。
- XPath 式内でJava と.NET プログラング言語の関数を使用することができます。ダイアログの下の「Java の使用」と
 「.NET の使用」が対しまXPath式内のJava と.NET 拡張関数の使用法に関する説明をする情報ボックスを表示します。
 詳細に関してはこのドキュメントの拡張関数セクションを参照してください。

- 式フィールド内のデキストのサイズを大きくする/小さくするコよ、式フィールをクリックし、Ctrl を押して、スクロールホイールを使用します。結果ペイン内にもこの機能が適用されることに注意してくだい。
- ノードのロケーター 次式を手動で挿入する代わりに、以下を行うことができます: (i) ロケーター 次を挿入する XPath 式内の場所にカーノルをポイントします (ii) ソースソノー内でターゲナとするノード内にカーノルをポイントします。式内の選択されたノードのロケーター 次を入力します。ロケーター 次はドキュメントのレートノードが始まる絶対パスです。

結果ペイン

結果ペインは下のスクリーンショナト内で表示されています。結果ペイン固有ののソールドーが存在することに注意してください。



結果ペイン/コお以下の機能が存在します:

- 結果リストは2つの列によい構成されています: (i) ノード名、まけはデータ型(ii) ノードのコンテンツ。
- XPath 式が要素おは属性などのノードを返す場合ノードのエンテンソ全体が一ドの値として表示されるかを選択することがてきます。これを行うには、完全な結果を表示するをオンセリル替えてください。
- 式におり生成されるリテラルとは異なり、結果に(テキストノードを含む)ノードが含まれる場合、結果ペイン内でそのノードをクリックすると、ソースソノー内のXMLドキュメント内の関連するノードが、イライトされます。
- 結果サブラインの別、まけよ、値列の両方をエピーすることができます。全ての別をエピーするコよ、サブラインを右クトックし「全ての列を含みコピーする」をオンコリル替えます。(まけは、結果ペインのソール・一内のアイエンを使用してコマンドをオンとオココリル潜えることもできます。)コピーするサブラインを右クトックし、(そのサブラインのおめは)「サブラインをコピーする」まけは「全てをコピーする」(全てのサブラインのためは)を選択します。

結果ペインのツールバー

結果パインのソール・一コはナビゲート、検索、およびコピー機能を提供するアイコンが含まれています。これらのアイコンは、左から順番に以下で説明されています。対応するコマンドは結果リストアイテムのエンテキストメニュー内で使用することができます。

アイコン	動作
次、前	結果リストからそれぞれ次のおよび前のアイテムを選択します。
クリップボードに選択されたテ キストラインをコピーする	ク <u>)</u> クトップボードに選択された結果アイテムの値カラムをコピーします。全てのカラムをコピーするコマンドに切り替えてくがさい(下を参照)。
全てのメッセージをクリップボ ード/ニコピーする	空の値を含む値カラムのすべての結果アイテムをクトップボードにコピーします。各アイテムは異なるラインにコピーされます。
全てのカラムを含みコピーす る	(i) 全てのカラム まけは(ii) 値カラムのエピー間を切り替えます。カラムの区切りは単一のスペースです。
検索する	結果リスト内で特別文字を含む文字列を検索する検索ダイアログを開きます。
前を検索する	検索ダイアログ内に最後に入力された用語の前の発生を検索します。
次を検索する	検索ダイアログ内に最後に入力された用語の次の発生を検索します。
クリア	結果のJストをクリアする
複数のラインの結果を単一のラインに折りたさむ	結果アイテムの値カラムか複数ラインのテキストを含む場合、(新しいラインまけば文字を含む場合)、複数ラインと単一ラインの表示を切り替えることができます。
完全な結果を表示する	ノードの値としてノードのコンテンツ全体を表示します。

11.2.2 デバッガ―

デ・・・・ガーにより XML ファイルと作業 のエンテキスト内で XPath 式をデ・・・グすることができます。「評価/デ・・・・グの開始」ドロップダウンメニュー内で「デ・・・・グの開始」にアクセスします (下のスクリーンショナー)。これによりデ・・・グラモードに設定されます。 式を作成するかのヘルプのオータの)ビルダーと(式をデ・・・グするかの)エ・・リュエーター間を切り替えることができます。 デ・・・グを開始するかっぱに評価/デ・・・グの開始」 (F5) をクトグします。



式の挿入後(デバッグモードにモードが切り替えられていることを確認して)「評価/デバッグの開始」(F5) をケックしてデバッグを開始することができます。.

□ デバッグモードをセットアップするためのボタン

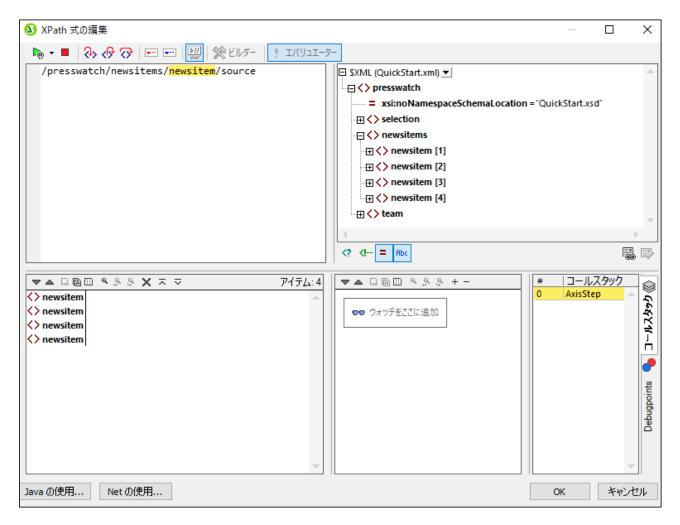
₽	評価/デバッグの開始(F5)	デッジガーを開始します。
----------	----------------	--------------

%	ビルダーに切り替える	式の作成を手助けするエンテキストを考慮する入力ヘッレトを提供する式ビルダーモードに切り替えます。
<u> </u>	入力中に評価する	式の挿入中に式の評価を切り替えます。

デバッグモードのレイアウト

デッグモード内では、結果ペイノマンのペインが追加されます(下のスクリーンショントを参照):

- ペイン内に個別のタブを持つコールスタックとデバッグポイントペイン
- ウォッチ式と変数が同時に同じペイノニ表示される変数とウォッチ式ペイン



デッグガーモードには以下の機能が搭載されています。

XPath 式がどのように評価さえるかをステップごとは確認するかがにXPath 評価プロセスにステップインすることができます。ステップイン (F11) ツール デーボタンを使用して、これを行います。各評価ステップで、現在評価されている式の一部が黄色で、イライ

トされており、そのステップの評価の結果は結果ペインに表示されます(上のスクリーンショナ)。例えば、上のスクリーンショナでは、book 要素のすべてのsection 子孫要素が選択されています。

- 評価を一時停止する箇所でブレーケポイトを設定し、これらの箇所で結果をチェックします。ブレーケポイトでのみ一時停止し評価をステップスルーすることができます。「デバッグの開始」(F5)ツールドーボタンをこのために使用します。ステップイン(F11)を使用したすべてのステップで一時停止するより、基準によってのステップで一時停止するより、またいできます。
- トレースポイントとしてマークされるステップでの結果を確認するようはトレースポイントを設定します。(ブレークポイント以外)評価は 一時停止しませんが、結果ペイン内のリスト内にトレースポイントの結果は表示されます。
- ウォッチ式を使用して、評価のドキュメントデータまけはアスペケトなどの)情報をチェックすることができます。この機能はブレーケポイントで役に立ちます。
- 変数とファッチ式ペイン内に表示されるスコープ内の値を含む変数。
- 評価ステップのプロセッサーコールはコールスタックとディッグボイントへインのコールスタックダブ内で表示されます。
- ブレークポイトとトレースポイトが設定されると、コールスタックとデバッグポイトペインのデバッグポイトタブ内で表示されます。

これらの機能の情報に関しては、下の詳細を参照してください。

デバッガーの実行

XPath 式のデバッグのオーかの大まかなステップは以下の通りです:

- 1. XPath 式を式ペイノに入力します。
- 2. ブレークポイト、まけおレースポイトを設定します。ブレークポイトは評価を停止するポイトを指します。トレースポイトは記録される評価内のポイトを指します。トレースポイトは従って評価結果のトレース可能ないを提供します。
- 3. 「デバッガーの開始」をケトックすると、式内にブレーケポイントがマークされていない限り、評価が最後のステップの前の段階まで行われます。「デバッガーの開始」を繰り返しケトックしてブレーケポイントを進め評価の最後まで移動します。
- 4. ステップイン/アナ/オー・、一機能を使用して評価をステップ、イステップ(ゴテ)ます。

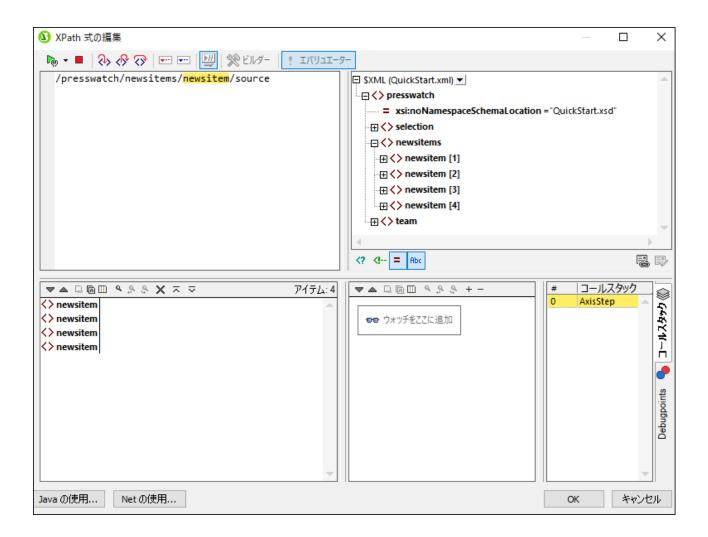
□ デバックのためのボタン

Þ	デバッガーの開始 (F5)	デバッガーを開始します。評価はブレーケポイントのためご停止する以外最後まで 行われます。
	デバッガーの停止 (Shift+F5)	評価を停止しデバッガーを停止します。
₹>	ステップイン (F11)	評価を継続し、ステップごと言評価を行います。
⟨ Ş	ステップアナト(Shift+F11)	現在の評価ステップをステップアナトし、親ステップに移動します。
₹\$	ステップオー・バー(Ctrl+F11)	子孫ステップにステップオー・ドーします。
	ブレークポイントの挿入/削除(F9)	カーノルがポイントする箇所の式ステップでブレーケポイントを挿入作除します。
•	トレースポイント <i>の</i> 挿入/削除 (Shift+F9)	カーソルがポイントする箇所の式ステップでトレースポイントを挿入/削除します。

ステップイン、アナト、およびオーバーの評価ステップ

ステップイン機能によい評価をステップ、イステップで行うことができます。このコマドの各クトックによい評価の次のステップに移動することができます。現在のステップは式内で、イライトされ表示されます(下のスクリーンショナを参照)。ステップアケ 機能によい現在のステップとして高いレベルのステップに移動することができまが、ステップオーン 一機能は対し、レベルのステップにステップオーン・一し、同じレベルの次のステップに移動することができます。ステップ機能を下のスクリーンショナト内で表示されている式を使用して、3つのステップが対しを押してそれぞれの仕組みを試すことができます。

下のスクノーンショナーはロケーターステップ newsitem に到達すると一時停止される際の評価を表示しています。この時点では結果は4つの newsitem ノードを表示しています。



ブレーケポイント

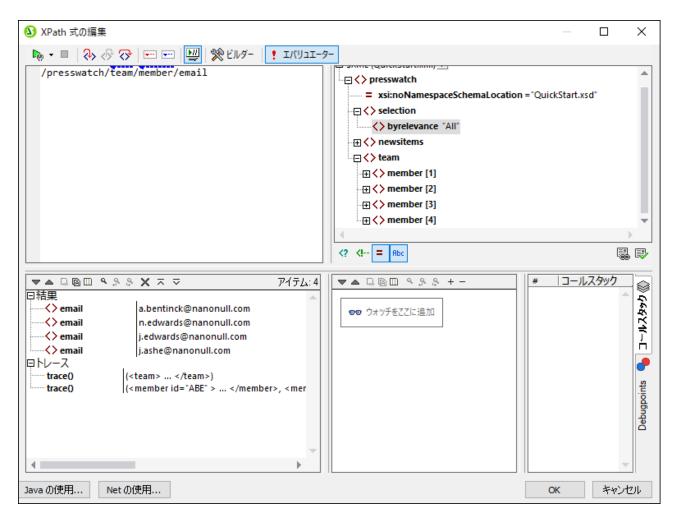
ブルーケポイトは「デバッガーの開始」にお開始された後にデバッガーを停止するかかポイトです。ブルーケポイトでデバッガーが停止すると、結果をチェックして、ステップイン機能を使用して評価の次のステップの結果を表示することができます。ブルーケポイトを設定するには、式内のブルーケポイトを設定する箇所にカーノルをポイトし、「ブルーケポイントの挿入/削除」(F9)ツールバーボタンをケナックします。ブルーケポイトは赤い点線のオーバーラインでマーケされています。ブルーケポイトを削除するには、選択し「ブルーケポイント/削除」(F9)をケナックします。

下のデッグポイトも参照してくたさい。

トレースポイント

トレースポイトは結果が記録されるポイトです。これらの結果は 結果 タブのトレースツリーで表示されて、ます (下のスクリーンショナを 参照)。これにより式の特定の部分のすべての評価を確認することができます。例えば、下のスクリーンショナ内では、トレースポイトが team ノードとmember ノード上で設定されています。これらのトレースポイトでの結果はトレースツノー内で表示されています。

トレースポイントを設定するコよ、トレースポイントを設定するポイントにカーソルをポイントし、ツール・・・・ボダン「トレースポイントの挿入/削除」(Shift+F9)をクナックします。トレースポイントは青い点線のオート・ラインでマークされています(*下のスクリーンショ外を参照*)。トレースポイントの挿入/削除」(F9)をクナックします。

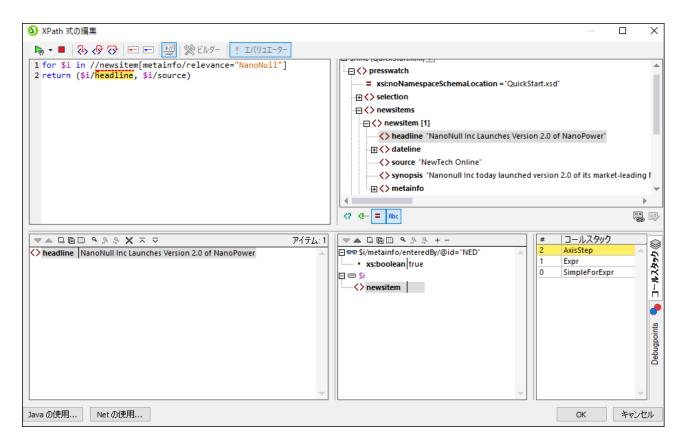


メモ ブレークポイントとトレースポイントが式の一部に設定されている場合、赤色と青色の交互の点線によりオードーラインが作成されます。

下のデッグポイトも参照してくたさい。

変数、ウォッチ式、コールスタック

変数とウォッチ式は変数とウォッチ式ペイン(のスクリーンショナの中央のペイン)内に表示されます。



変数

式内で宣言されており、現在の評価ステップ内のスコープに存在する編集は相対する現在の値と共に表示されています。例えば、上のスクリーンショナでは、headline 上のブレークポイトで処理が一時停止されています。\$i 変数はこの評価ステップではスコープ内です。ですから、上のスクレーンショナ内で\$i が最初のソードである現在の値と共に表示されています。

ウォッチ式

ウォッチ式は評価の開始前、ませば評価の一時停止中に挿入することのできる式です。以下の目的のために使用することができます。

- 特定の条件をテストする場合。上のスクノーンショナ内のサンプルでは、ウォッチ式
 \$i/metainfo/enteredBy/@id="NED" がNED のID を持つチームメンバーによりにの新規のアイテムが入力されているかをテストするために使用されています。結果 true は最初のニュースアイテムがこの条件が満たしているかを示しています。
- 特定のエンテキスト内のデーダを検索する場合。例えば、Company 要素エンテキストでは、ターゲットドキュメント内で企業のカスタマーコードを検索するファッチ式 @id を入力することができます。
- 追加データを生成する場合。例えば、新規のアイテムの総数を示す適切な文字列が生成されます。

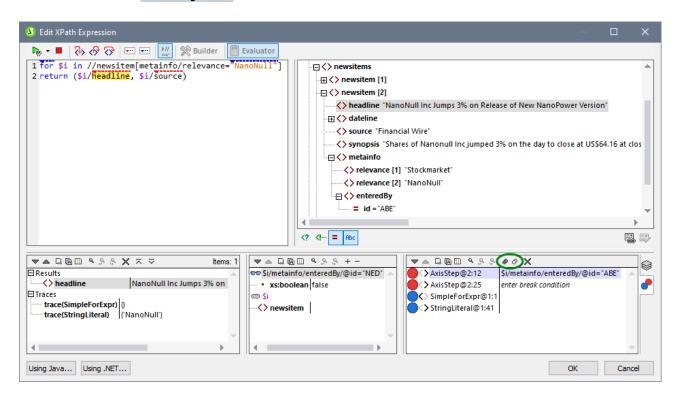
ウォッチ式を挿入するコまペインのソール・一内の、上のスクリーンショナト内で赤枠で囲まれている)「ウォッチエントリの追加」をクトックし、式を挿入し、終了すると「Enter」をクトックしてください。 ウォッチ式を削除するコま ツール・一内で「選択されたウォッチエントリの削除」をクトックしてください。 ディッグ中に式が例えば、変数のこっか範囲以外など、何らかの理由で正確に評価されない場合、ウォッチ式は赤色で表示されます。

<u>コールス</u>タック

コールスタックとデバッグポイントペインのコールスタックタブはデバック内のそのポイントまでのプロセッサーの呼び出しを表示します(上のスクリーンショナ内の右下のペイン)。現在のプロセッサーの呼び出しは黄色で、イライトされています。現在の評価ステップに直接導いた呼び出しのみが表示されることに注意してください。

デバッグポイント

コールスタックとデッシグポイントペインのデッジグポイントタブは式内で設定された。赤い、丸印で表示されるブレークポイントと(青い、丸印で表示される)トレースポイントを表示しています(下のスクリーンショントの右下のペイン)。各ダイアログポイントはラインと文字番号と共にストされています。 例えば、AxisStep@2:12 は式ペイン内の式のライン 2 と文字 12 にデッシグポイントが存在することを意味します。



以下の機能に注意してください

- ブレーケポイトのためにデッグポイトペイン内の ブレーケポイントの条件をダブルクトックしてブレーケの条件を入力し、条件のために式を入力し Enter を押しますます。条件がtrue と評価された場合のみそのブレーケポイトは有効化されます。例えば、上のスクナーショナではブレーケ条件 \$i/metainfo/enteredBy/@id="ABE" はid ABE を持つチームメンドーにより入力されたニュースアイテムのそれぞれのヘッドライン上のブレーケポイトを有効化します。スクナーショナトはこのブレーケポイトで一時中止されている評価を示しています(子のブレーケポイントでのファッチ式がfalse を返すことご注意してください。)
- 対応するソール デーボタンをクリックして全てのディッグポイントを有効化/無効化することができます。全てのディッグポイントの有効化と全てのディッグポイントの無効化ディッグポイント(上のスクリーンショナトので緑で囲まれているがタン)。ディッグポイントが無効化されていると、有効化されるまで全ての評価のためにディッグポイントは無効化されています。
- 対応するコンテキストメニューのために個別のブレークポイントを有効化/無効化することができます。

ペイン内のソール・デコマンド

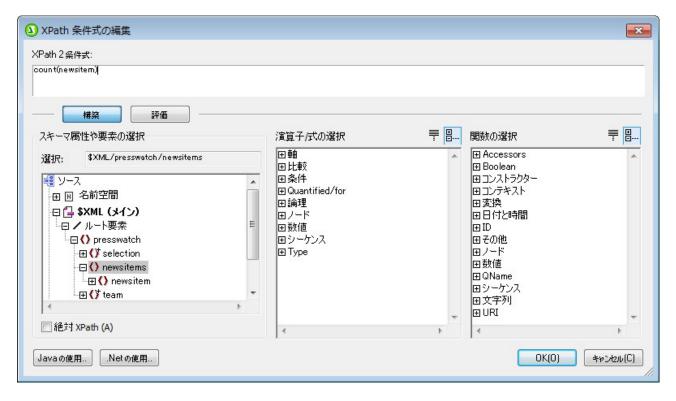
デバッグモード内の式の編集ダイアログのペインコよけビゲート、検索、およびロピーの機能を提供するがタンか含まれています(*上のスクリーンショナ*)。これらのがタンは左側から下のテーブル内で説明されています。相対するコマンドもリストされるアイテムのエンテキストメニュー内で使用することができます。

アイコン	動作
次、前	結果リストからそれぞれ次のおよび前のアイテムを選択します。

クリップボードに選択されたテ キストラインをコピーする	ク)シプボードに選択された結果アイテムの値カラムをエピーします。全てのカラムをエピーするコよ、全てのカラムを含みエピーするコマンドに切り替えてくたさい <i>(下を参照)。</i>
全てのメッセージをクリップボ ード/ニコピーする	空の値を含む値カラムのすべての結果アイテムをクトップボードにコピーします。各アイテムは異なるラインにコピーされます。
全てのカラムを含みコピーす る	(i) 全てのカラム まけは(ii) 値カラムのエピー間を切り替えます。カラムの区切りは単一のスペースです。
検索する	結果リスト内で特別文字を含む文字列を検索する検索ダイアログを開きます。
前を検索する	検索ダイアログトリニ最後に入力された用語の前の発生を検索します。
次を検索する	検索ダイアログ内に最後に入力された用語の次の発生を検索します。
クア	結果のノストをクリアする
複数のラインの結果を単一のラインに折りたさむ	結果アイテムの値カラムが複数ラインのテキストを含む場合、(新しいラインまだは文字を含む場合)、複数ラインと単一ラインの表示を切り替えることができます。
完全な結果を表示する	ノードの値としてノードのコンテンツ全体を表示します。

11.2.3 式ビルダー

XPath 条件式の編集ダイアログ内にある[構築] ポタンかやリックされると、XPath 条件式を構築するかがの入力へいいーペインが表示されます。これら入力へいいー内にあるエイ・リーをダブルグリックすることで、テキスト ボックスのカーノル位置にそのアイテムが挿入されます。



入力ヘルーペインコお以下の3 種類があります:

- XPath 式内に要素と属性を挿入するためのスキーマソノー。相対的な XPath チェックボックスかチェックされていると作成されている XPath 式内のデザイン内のノードである)選択されているノードへのロケーション、以はエンテキストノードに対して相対的に挿入されます。 エンテキストノードはスキーマソノーペインの下に表示されています。 絶対 XPath 式はボキュメントルートから始まり、Relative 相対的な XPath チェックボックスかチェックされていない場合選択されナノードのために使用されます。
- 演算子と式のための入力へルトーペイン。以下か含まれます: (i) 軸 (ancestor::, parent::, など) (ii) 演算子 (例えば eq とdiv) および(iii) 式 (for # in # return #, など)。ペインのアイテムはアルファベナ 順、おとは関数カテゴリ 別にグループ分けすることができます。ペインのタイトルドー内のドロップダウンメニューから 階層的 ませま フラナケ を選択してオプションを選択します。
- アルファベナ順、おけば数別にグループ分けされたアクティブな XPath バージョンの関数を持つ入力ヘルルー。ペインのタイトル バー内のドロップダウンメニューから 階層的 まけは フラナト を選択してオブションを選択します。 The Names/Types オブションに

 レ関数の別数を選択することができ、名前、まけばデータ型として表示することを選択できます。

ビルダーの機能

- 両方のペイン内のアイテムコ関する詳細を表示するコよマウスをアイテムの上にかざします。
- 各関数は、自身の引数、引数のデータ型、関数の出力のデータ型などである署名と共にリストされています。
- 署名は、名前、おは、関数の別数と出力のデータ型を利用してリストされています。ペインのタイトルドーのドロップダウンメニューから名前 おは 型 を選択します。
- ペイン(演算子、条件式、まけま、関数) 内のアイテムをダブルクトックして、式内のカーノルの箇所にアイテムを挿入します。関数は プレースオルダ(罪 シンボル)により示されている引数を使用して挿入されます。
- (i) XPath 条件式の編集フィールド内でテキストが選択されており、(ii) プレースヤックを含む関数の条件式が挿入のためにダブッルクトックされている場合、選択されていたテキストがプレースオックの代わりに挿入されます。

条件式内に関数を挿入後、関数名の上にマンスをポイントすると、関数の署名と関数の説明をしたテキストが表示されます。同じ名前を持つ関数のために異なる署名が存在する場合、これ以よオーバーロード要素として下の部分に示されます。カーノルを関数のかに内にかざし、Ctrl+Shift+ Space キーを押すと、その関数名の多種のオーバーロードの署名を確認することができます。

XPath 式の作成

XPath 式 の編集ダイアログは次の方法でXPath 式の作成を手助けします。

- コンテキストノードとスキーマツリー
 - ソースペイン内の選択 テキストボックス はすく にこしテキストノードを表示します。 式はこのコンテキストノード内の場所に挿入され、コンテキストとしてこのノードと共に評価されます。
- スキーマツノーからのノードの挿入

ソースペイン内で、スキーマ全体が表示されます。スキーマソー内のソードをダブルクトックし、XPath 式に挿入します。If the Relative XPath チェックボックスがチェックされていると、選択されているノードはコンテキストノードに相対的なロケーション・父式と共に挿入されます。

● <u>名前空間の情報</u>

スキーマ属性や要素ペインにあるスキーマッリーには名前空間アイテムが含まれます。このアイテムを展開すると、スタイルシート内部で宣言されたすべての名前空間が表示されます。この情報は XPath 条件式にて使用する名前空間の接頭辞を確認するために使用することができます。

• XPath 軸、演算子、式の挿入

演算子/式の選択ペインでは、SPS で選択されている XPath の軸 (ancester:: や parent:: など)、演算子 (eq や div など)、または式 (for # in # return # など) が表示されます。表示方式は、アルファベット順と、機能によりアイテムがグループ分けされた階層構造から選択することができます。軸、演算子、または 式を XPath 条件式に挿入するには、目的のアイテムをダブルクリックしてください。

XPath 関数の挿入

関数の選択ペイノは、XPath 関数をアルファベナ順に、おけま、機能別のグループ別にリストして、ます。(ペインの上のアイコンをケックして、2つの異なる表示方法を切り替えてください)。各 関数は自身の署名と共にリストされています。関数にコン以上の署名が存在する場合、その関数は署名の数だけリストされます。署名内の別数は、ガンマで区切られており、各引数には、発生各ごとご発生インジケーターを有します(? はゼロのシーケンス、おけま、指定された型の1つのアイテムを示し、* iは、ゼロのシーケンス、おけま、指定された型の1つ以上のアイテムを示します)。引数は、名前、おけまデータ型ごとに表示されます。ペインの上のThe arguments can be displayed as names or as datatypes; select Names or Types in the title bar of the pane.各関数は、その関数の戻り型を指定します。例: => date ? は、期待されるデータ型は、無しのシーケンス、まけま、date アイテムであることを示しています。関数の上にマウスをかざすと、関数の短い説明が表示されます。

• Java ならびに NET 拡張を XPath 条件式に使用することで、これらプログラミング言語に搭載された関数 ヘアクセスすることができます。ダイアログ下部にある [Java] ならびに [.NET] ポタンをクリックすると、 XPath 条件式内部で Java ならびに NET 拡張関数を使用するための方法が示されます。この機能の詳細については、このドキュメンテーションの 拡張関数 を参照(たさい)

直接のテキストエントリ中のインテリジェントな編集

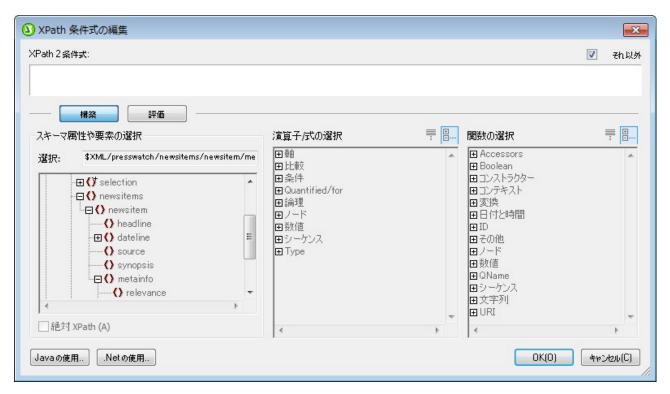
条件式テキスト ボックスへ条件式を直接入力する場合、その時点で利用することのできるオプションがポップアップで表示されます(以下のスクージションを参照)。



これらは要素 XPath 関数とXPath 軸を含みます。「上」と「下」キーを使用してオプションを上下に移動します。オプションを選択して、 式に挿入する場合 Enter を押します。

それ以外の場合チェックボックス

条件テンプレートの条件分岐にそれ以外の条件を追加することができる場合、[それ以外] 切り替えが表示されます(以下のスクリーンショットの赤い矢印を参照)。それ以外の分岐が存在しない条件テンプレートにチェック ボックスは表示されるほか、すでにそれ以外の条件分岐が含まれる条件テンプレートを編集する際にも [それ以外] チェック ボックスは表示されます。[それ以外] チェック ボックスにチェックを入れ、[OK] ボタンをクリックすることで、条件テンプレートにそれ以外の条件分岐が追加されます。[それ以外] チェック ボックスにチェックを入れると、構築モードにあるすべての入力ヘルパーが無効になります。これは、それ以外の条件分岐に与えられる条件はなく、XPath 条件式を構築する必要がないためです。



[それ以外]条件分岐を使用する詳細によいてよ「次を参照してくけさい 条件テンプレート。

11.3 ツールバー

ツールドーでは様々な StyleVision コマイが用意されています。ツールドーは以下のようは構成されます:

- 書式
- テーブル
- デザイン要素の挿入
- デザインフィルター
- 標準

各ツールドーのアイコンは、対応するコマンドの簡単な説明とともに、このセクション以下にあるサブセクションに記されます。

ツールドの位置

ツール・ー・はフロート状態にして画面上の自由な位置に配置することができるまか、GUI のファイ・ウ内部に配置することができます。ツール バーは通常 GUI の上端にあるメニューバー以下に配置されますが、ウィイ・ウク両端ませば下部に配置することもできます。

以下の操作により、ツール・領域にソール・を配置することができます。

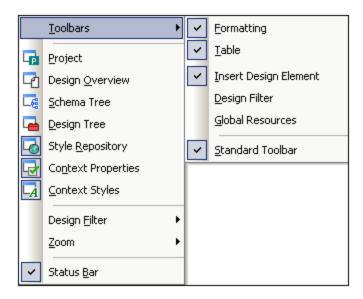
- 1. ツールバーのハンドル(ツールバーがツールバー領域に配置されている場合)またはそのタイトル バー(ツールバーがフロート状態の場合)をクリックします。
- 2. ツールバーを目的の場所までドラッグし、ドロップします。目的の場所にツールバーが存在しない場合、ウィンドウの端にツールバーを移動させることで、新たなツールバーが表示されます。

ツール・をプロート状態にするコよッツール・領域の外にソール・をドラッグし、ウィバウの端以外の画面上の任意の位置でドロップします。

ツールドを表示/隠す

以下にある3 つの方法で、個々のソール デーの表示を切り替えることができます:

• [表示 | ツールバー] メニューにて、ツールバーから対応するツールバーを選択または選択解除します。.



• ツールバーの領域を右クリックして、コンテキストメニューから個々のツールバーの表示を切り替えます (以下のスクリーンショットを参照)。

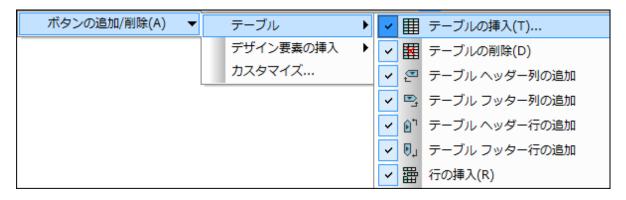


(ツール| カスタマイズ)メニューにて表示されるカスタマイズ ダイアログ のソール データブこて、ツール デのチェック ボックスを . クナックすることで、個々のソールドーの表示を切り替えます。完了した後には「閉じる」 ボタンによりダイアログを閉じるこ とができます。

ツールドボタンを追加/削除

ツール・ーニある個々のだタイお追加まける消除することができます。以下の操作により、ツール・ーからがタンを追加まける消除することがで きます:

- 1. 追加または削除するボタンに関連付けられたツールバーにて、(ツールバーがツールバー領域内に配置されて 1. 追加または削除するパダンに関連付けられたソールハーにで、(ソールハーかソールハー領域内に配置されている場合) [カスタマイズ] ボタンまたは(ツールバーがフロート状態の場合) [ツールバーのオプション] ボタンをクリックします。[カスタマイズ] ボタンはツールバーの右端または下端に表示されている矢印のことです。[ツールバー オプション] ボタンは、フロート状態のツールバー右側に表示されます。
 2. 表示される [ボタンの追加/削除] メニュー上にマウス カーソルを移動させます。 そのツールバー領域に表示されているツールバーの名前と、[カスタマイズ] メニュー アイテムが表示されます (以下のスクリーンショットを
- 参照)。



- 3. 追加または削除するツールバーボタンを含んでいるツールバーにマウスを移動させます(上のスクリーン ショットを参照)。
- 表示されるメニューにて、追加または削除するツールバー ボタンの名前をクリックします。
- 5. [カスタマイズ] アイテムをクリックすると、カスタマイズダイアログが表示されます。

ボダンのノスト以下にある[ツールバーをリセット] アイテムこより、そのソールドをStyleVision. をインストーリした状態に戻すことが できます。リセットされると、そのソール・・ースあるすべてのがタンか装示されます。

廴 ツールバーに含まれるボタンと、そのツールバーとの関連付けを解除することはできません。上に示され たツールバーのボタンを表示/隠す動作は GUI に対してのみ適用されます。

三 関連項目

ユーザーインターフェイス

11.3.1 書式

[書式] ツール・コンデザインビューで表示され、一般的なインラインならびブロックのフォーマナプロ ティを、SPS のデザインにて選択されたアイテムへ割り当てるナックロマンドが含まれています。



定義済みの HTML フォーマット

ドロップダウン リストから選択された HTML フォーマットがデザイン ビューの選択対象に適用されます。例えば div を選択することで、HTML のブロック(div) 要素がデザイン ビューにて現在選択されている要素の周りに適用され ます

テキスト プロパティ

配置

配置プロケイ(左寄せ、中央揃え、右寄せ、両端揃え)を、デザインビューの選択対象に直接適用します。

リスト

カーソルの挿入ポイントにリストを挿入することができるほか、SPS の選択対象をリストへ変換することができます。

ハイパーレンク

カーソルの挿入ポイントにハイパーリンクを挿入します。このコマンドの詳細な使用方法については、<u>ハイパーレク</u>を参照ください。

□ 関連項目

- ・ツールド
- 書式とスタイル
- 入力フォーマ小

11.3.2 テーブル

[テーブル]ツール・はデザインビューで表示され、静的ならびに動的なテーブルの構造やフォーマナを指定するコマナが収められています。テーブルツール・内にあるこれらのコマナドを以下のスクレーンショナに示します。

~		テーブルの挿入(T)
~	M	テーブルの削除(D)
~	•	テーブル ヘッダー列の追加
~	•	テーブル フッター列の追加
~	₽ [†]	テーブル ヘッダー行の追加
~	P	テーブル フッター行の追加
~	噩	行の挿入(R)
~	思	行の追加(W)
~	器	行の削除(O)
~	贈	列の挿入(C)
~	闡	列の追加(N)
~	膃	列の削除(L)
¥	<u>⊞</u> join	右のセルを結合(G)
~	년 join	左のセルを結合(F)
~	join	上のセルを結合(A)
~	join join	下のセルを結合(E)
~	€][÷ SPLIT	セルを水平方向に分割(H)
~	I SPLIT	セルを垂直方向に分割(V)
~	$\overline{\uparrow}$	上揃え
~	*	中央揃え
~	±	下揃え
~	F.	テーブルマークアップの表示(M)
~		セル境界の表示(B)
~		テーブルプロパティ(P)
		ツールバーのリセット(R)

行と列の操作

SPS のテーブル(静的ならび、動的) に対して行ませる別の挿入、追加、削除を行うことができます。 行ませる別は現在のカーノル位置の直前に挿入、ませる直後に追加されます。 カーノルが配置されている行/列を削除することもできます。 これらの操作は[行/列の挿入]、[行/列の追加]、[行/列の削除] ボダイこより行われます。 [テーブル ヘッダー行/列の追加] ませま [テーブルフッター行/列の追加] により、テーブルのヘッダーやフッターを追加することもできます。

セルの操作

現在カーノルが配置されている SPS テーブルセルを、隣接する4 つのセルの1 つに結合することができます。結合操作は HTML のテーブルコンドルス・カンドルトルの操作のために使用されます。更に [セル

を水平方向に分割] おけま[セルを垂直方向に分割] ボタイより、カーノルが配置されている. SPS テーブルのセルを、それぞれ水平おけま垂直方向に分割することもできます。SPS テーブルのセル内部にあるコンテンソは、上揃え、中央揃え、下揃えすることができ、「セルの境界を表示] ボタンをクトソクすることで、セルの境界線を表示おけお隠すことができます。

テーブルの操作、プロパティ、表示

静的まけば動的なテーブル内部にカーソルが配置された状態で<u>テーブルの削除</u>をクシンフすることで、そのテーブルが削除されます。テーブルのマーケアンプ表示は、[テーブル マークアップの表示] ポタイことりい替わります。[テーブルのプロパティ] コマイにより、そのテーブルのプロ ティを定義するさかのダイアログが表示されます。

- □ 関連項目
- ツールドー
- <u>テーブル</u>
- テーブルメニュー

11.3.3 デザイン要素の挿入

デザイン要素の挿入 ツール・・ツール・・コは SPS デザイン内部へデザイン要素を挿入するコマドや、関連するコマドに対応したアイエカ 含まれて、ます。ツール・・アイエカら挿入することのできるデザイン要素を以下のスクリーンショナに示します。ツール・・・コは3 種類のアイテムが収められて、ます:

- 1. デザク要素:コンテキストノードン影響を受けます(ツールドーコを引きの要素です)
- 2. レイアウト要素:ノードコンテキストとは独立して使用できます
- 3. グリナドに関連する操作:デザインの補助に使用されます

→ D	テンプレートの挿入
✓ ♣	ユーザー定義テンプレートの挿入
→	コンテンツの挿入
→	リッチエディットの挿入
✓ <u>I</u>	入力フィールドの挿入
✓ I ^ □	複数行入力フィールドの挿入
~	コンボ ボックスの挿入
▽ □	チェック ボックスの挿入
v 0	ラジオ ボタンの挿入
~	挿入時にデータ ソースを確認
~	自動計算の挿入
✓ §	段落の挿入
✓ <u> </u>	リストの挿入
✓ Ⅲ	テーブルの挿入
→	パーコードの挿入
→	チャートの挿入
✓	イメージの挿入
~	レイアウト コンテナーの挿入
✓ T	レイアウトボックスの挿入
~ \	ラインの挿入
~	グリッドの表示
~	グリッドへのスナップ有効
	ツールバーのリセット(R)

デザイン要素

デザル要素は「挿入」メニューからも利用することができる、コンテキストノードに影響を受けるノードです。以下の操作により、ツール・・ア イエムからデザイン要素を挿入することができます:

- 挿入する要素のツールバーアイコンを選択します。
 要素が挿入されるデザイン上の場所をクリックします。選択された要素のためのデザイン要素の挿入ダイアログボステンスを表す。 す。コンテキストノードとは、デザイン要素を挿入するために指定したカーソル位置の外側にあるノードのことで
- 3. 現在選択されているコンテキストノード内部にデザイン要素を挿入するには、[OK] をクリックしてください。他 のコンテキストノードを選択するには、スキーマッリーにて目的のコンテキストノードを選択し、[OK] をクリックしてください。
- 4. 自動計算といったデザイン要素の場合、自動計算を定義するといった操作を更に行う必要があります。そうで ない場合(例えばユーザー定義テンプレートの挿入など)、デザイン要素の挿入ダイアログがスキップされます。

この場合、XPath 式の編集ダイアログ といた、別のダイアログが表示されます。必要な情報を入力し、ダイアログの[OK] ボタンをクリックしてください。

挿入されたデザインの種類により、ステップ3 ませるステップ4 でデザイン要素が挿入されます。

レイアウト要素

デザイン要素を挿入ツール ヾーコよ (i) レイアナ コンテナー、(ii) レイアナ ボクス そして (iii) ラインを挿入するための、3 つのレイアナ 要素コマンドが収められています。レイアナ ボクスとラインはレイアナ コンテナー内部だけに挿入することができることに注意してください。

レイアウト コンテナーを挿入するづま [レイアウト コンテナーの挿入] アイコを選択し、レイアウト コンテナーを挿入するデザイン上の場所をクトグしてください。レイアウト コンテナーの大きさごな て尋ねられ、選択によりレイアウト コンテナーが挿入されます。レイアウト ボックスの挿入] アイコをクトグし、レイアウト ボックスの挿入] アイコをクトグし、レイアウト ボックスを挿入するしょ [レイアウト ボックスの挿入] アイコをクトグし、レイアウト ボックスを挿入するしくアウト コンテナー上の場所をクトグしてください。レイアウト ボックスが挿入されます。レイアウト ボックス内部をクトグすると、キーボードによる文字の入力を出じめることができます。ラインを挿入するづま [ラインの挿入] アイコをクトグして、レイアウト コンテナー内にあるライの開始点までカーノルを移動してください。クトグによりラインの開始点がセトされ、目的の終了点までカーノルをドラッグしてください。終了点でカーノルをドロップすることで、指定されて開始点が必必了点までのラインが挿入されます。

レイアナ コンテナーやレイアナ ボックスの大きさを変更するコよ レイアナ コンテナーまり はイアナ ボックスの右端まり は下端までカーノ ルを移動させ、境界線をドラックしてくたさい。レイアナ ボックスを移動するコよ レイアナ ボックスの上端まり はまましま たばまでカーノルを移動させ、カーノルの表示が十字に変わったころでドラックによルイアナ ボックスを新たむ場所へ移動してくたさい。

グルドは関連する操作

[グリッドの表示] コマイドにより、グリイドの表示/非表示を切り替えることができます。[グリッドへのスナップ] コマイドが有効になっている場合、レイアナ コンテナー内部で作成されたエイアナ ボックスやラインが、グリイド線やグリッド線の交差点に対してスナップされます。 グリイドのプロ ティレオプションダイアログ ([ツール | オプション]) のデザイン タブ こてセナ することができます。

□ 関連項目

- ・ツールド
- 挿入メニュー
- レイアナコンテナー

11.3.4 デザインフィルター

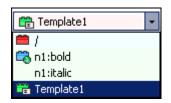
デザインフィルターツール・一(アのスクリーンションナ) には、デザインに表示されるテンプレートのフィルタルノブを行うさせのロマンドが収められています。ツール・コニある各アイコンについて以下で説明します。.



アイコン	コマンド	説明
ij	選択されたテンプレート だけを表示	選択されたテンプレートだけを表示します。テンプレート内にカーノルを配置してこのアイエンを クトックすることで、そのテンプレートだけが表示されます。
	全種類のテンプレートを 表示	SPS 内にあるすべてのテンプレート (メイン、グロー・) 认 名前付き、レイアナト) が表示されます。

アイコン	コマンド	説明
	インポートされたテンプレ ートを表示	インポーされたテンプレートの表示を有効/無効/1します。
	メイン テンプレートを表 示 <i>/</i> 隠す	メイン テンプレートの表示を有効/無効にします。
	グロー・バル テンプレー ト を表示/隠す	グロー・ジルテンプレートの表示を有効/無効にします。
	デザイン フラグメントを 表示/隠す	デザイン フラグメント の表示を有効/無効にします。

デザインフィルターコンボボックスコはSPS にある全テンプレートのリストが表示されます(以下のスクリーンショナを参照)。



コンボボックスからテンプレートを選択することで、そのテンプレートかデザインにて選択されます。コンボボックスによりデザイン内にある目的のテンプレートへ素早くアクセスすることができ、複数のテンプレートかデザイン内部に存在する場合、その中でもテンプレートの一部が表示されていないような状況で利用することができます。

- □ 関連項目
- <u>ツールド</u>
- デザインソノー

11.3.5 標準

標準ツール・一口は重要なファイルが編集機能に関するコマドが含まれています。簡単な説明ととしてれるのアイコンを以下に示します。コマド名をクトックすることで、リフォンス・セグションにあるコマドの詳細な説明を確認することができます。

アイコン	コマンド	ショーナカット	説明
₽ ₽	XML スキー マDTD/XML から 新規作成	Ctrl+N	スキーマをベースした SPS ドキュメトを新わっ作成します。ドロップ ダウをクトクすることで、DB やHTML ドキュメトをベースした SPS や、空のSPS を作成することができます。
A	<u>開〈</u>	Ctrl+O	既存のSPS ドキュメントを開きます。
	デザインを保存	Ctrl+S	アクティブな SPS ドキュメントを保存します。

アイコン	コマンド	ショーナカット	説明
	すべて保存	Ctrl+Shift+S	開かれているすべてのSPS ドキュメントを保存します。
	<u>印刷</u>	Ctrl+P	作業用 XML ファイルの Authentic View を印刷します。
<u> </u>	印刷プレビュー		作業用 XML ファイルの Authentic View の印刷プレニューを表示します。
*	切り取り	Shift+Del	選択対象を切り取り、クトップボードにコピーします。
	<u> </u>	Ctrl+C	選択対象をクソプボードにコピーします。
	貼り付け	Ctrl+P	クトプボードの内容をカーノル位置へ貼り付けます。
×	<u>削除</u>	Del	選択対象を削除します。
2	元に戻す	Alt+ Backspace	編集された変更を元に戻します。元に戻せる回数に制限はありません。
2	<u>や値し</u>	Ctrl+Y	元に戻した操作をや値します。
#	<u>検索</u>	Ctrl+F	Authentic View や出力ビューにあるテキストの検索を行います。
4	次を検索	F3	検索されたテキストが次に出現する箇所を検索します。
XSLT 1.0	XSLT 1.0		XSLT 1.0 がスタイルシート言語とて設定されます。
XSLT 20	XSLT 2.0		XSLT 2.0 がスタイルシート言語として設定されます。
XSLT 3.0	XSLT 3.0		XSLT 3.0 がスタイパント言語とて設定されます。
ABC	スペルチェック		SPSドキュメトにてスペルチェックを行います。

■ 関連項目

• <u>ツールド</u>

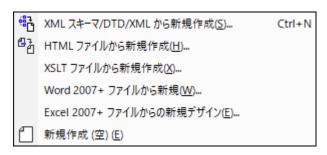
11.4 ファイル メニュー

ファイルメニューコよ SPS や関連するファイルを操作するためのコマイが収められています。以下のコマイを利用することができます:

- 新規作成様々なソースからSPS を新たった成します。
- 開く、再ロード、閉じる すべて閉じる ファイルを開く、アクティブなファイルを閉じる、ませは再ロードします。
- <u>デザインの保存、名前をつけて保存、すべて保存</u> アクティブな SPS や、開かれているすべての SPS ファイルを保存するナーダのコマドです。
- MobileTogether デザインファイルとしてエクスポートするを使用して、アクティブな SPS ファイルから MobileTogether デザインを生成することができます。
- <u>生成されナンアイルの保存</u> SPS により生成することのできる出力ファイルを保存します。
- Web デザイン ASPX アプリケーションを動作させるのに必要なすべてのファイルを、指定されたフォルダーに生成します。
- プロ 守ィ 出力ドキュメントのエンコーディング、ブラウザーの CSS 互換モード、Authentic View における相対イメージ パスの解決方法、RTF (Enterprise ならびに Professional エディションのみ) や Word 2007+ (Enterprise エディションのみ) 出力にてイメージを埋め込むのか、またはリンクするのかの指定を行います。
- 印刷プレビュー、印刷出力ビューインで有効しないます。プレビューン表示されている内容を印刷するなめのコマンドです。
- 最後に開かれナンアイル、終了 それぞれ最後に開いナンアイルを開く、プログラムを終了するナーダのコマンドです。

11.4.1 新規作成

[新規作成] コマバトへマブスカーノルを配置すると、SPS ドキュメントを新たった成するためのサブメニューが表示されます(以下のスクリーシショントを参照):

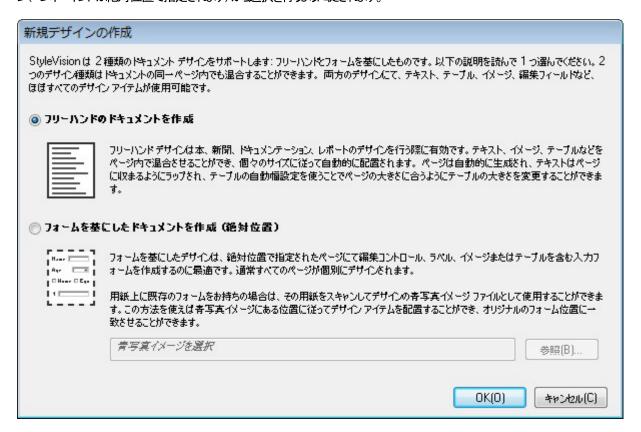


- XML スキーマや DTD、XML ファイルから生成された XML スキーマをベースにした SPS ファイルを新たに作成します ([XML スキーマ/DTD/XML から新規作成])。選択されたスキーマが デザル概要サイバーへ 追加され、(スキーマソノーサイバー(する) スキーマツノースキーマの構造が表示されます。デザルビューでは空のデンプレートと生にSPSが作成されます。 新規 SPS はURL やグローバレリノースを介したファイルがも作成することができます (下記参照)。
- HTML ファイルからノードごとに作成したユーザー定義のスキーマをベースに、新規 SPS を作成します([HTML ファイルから新規作成])。ユーザー定義のスキーマがデザク概要サイドバーとスキーマソノーサイドバーへ追加されます。単一のドキュメト要素(ルト要素)を持ったソノーがスキーマツノーで表示され、HTML ファイルがデザクビューに表示されます。
- XSLT-for-HTML または XSLT-for-FO ファイル、FO ファイルから SPS を作成することができます。XSLT のテンプレート構造ならびにスタイルが SPS にて作成されます。SPS コンポーネントを修正し、コンテンツやフォーマットを SPS に追加することができます。詳細については「XSLT かの新規作成」を参照付さい。
- デザインの静的テキストとしてMS Word ドキュメントのコンテンンを含む新規 SPS。
- デザインの静的テキストとしてMS Excel ドキュメトのコンテンンを含む新規 SPS。
- 空の SPS を新たに作成します ([新規作成 (空)])。 デザイン概要サイドバーやスキーマ ツリーにスキーマ は追加されず、 デザインビューに 空のメイン テンプレート が作成されます。

デザインの種類を選択する

(XSD とXML) といない一スファイルを選択すると、新規デザインの作成ダイアログが表示されます。

新規デザインを作成ダイアログボックスか装示され、(i) フレー ンドのドキュメント デザイン、まけま(ii) フォームをベース ユナボキュメント デザイン (コンポーネント が絶対位置で指定されます) から選択を行うように促されます。

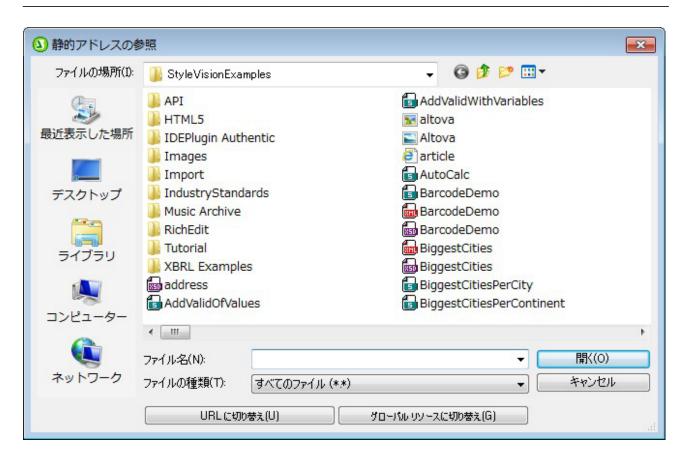


フノー・とドのドキュメトデザインでは、ドキュメトコンテンンが出力メディアオブジェクト(印刷物や画面)にファトするようコンイアナされます。ドキュメトコンテンソ内のアイテムは相対的にしか配置することができず、絶対位置を指定することはできません。このオプションは、レポートや記事、書籍などのデザインに適しています。

フォームをベースユナギキュメントロは単一のレイアナコンテナーが作成され、その中でデザインコンポーネントを絶対位置で指定することができます。レイアナコンテナーコは、絶対位置でレイアナボックスを配置することができ、各レイアナボックス内にギキュメントコンテンクを配置することができ、BにあるフォームのデザインをSPS にて再現したい場合、オバジナルのフォームイメージを青写真イメージとして使用することができます。 青写真のイメージをレイアナコンテナーの背景として含めることができ、フォームのデザインに役立てることができます。

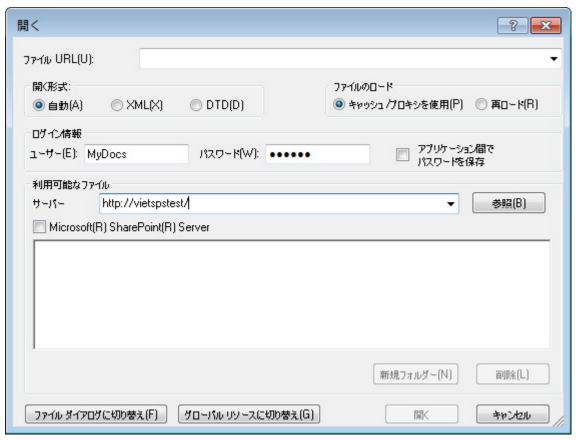
URL とグローバルノソースを介してファイルを選択する

ファイルを開くならび、ファイルを保存ダイアログによっては、URL やグロー・ゾレリソースにより目的のファイルを選択することができます(以下のスクリーシショナを参照)。[URL にスイッチ]、ませま「グロー・ゾレノソースにスイッチ] を選択することで、別の選択プロセスが表示されます。

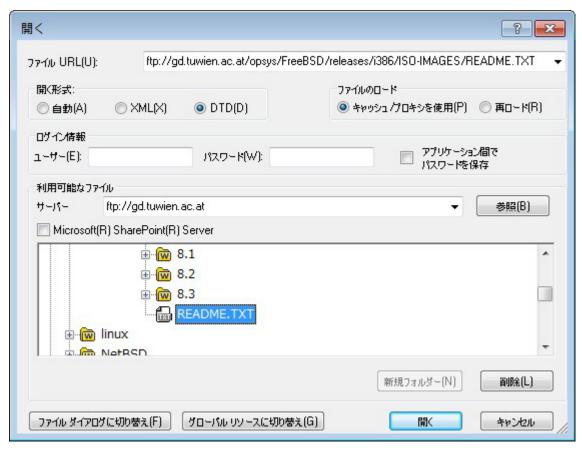


URL によりファイルを選択する 以下の操作によりURLを介したファイルの選択を行うことができます:

1. [URL にスイッチ] コマバをクトックします。URL モードのファイルを開くダイアログが表示されます(以下のスクリーンショナを参照)。



- 2. サード URL フィールドニアクセスする URL を入力します (上のスクリーンショナを参照)。サードが Microsoft(R) Share Point(S) Server の場合、 Microsoft(R) Share Point(R) Server チェック ボックスニチェックを入れてくけさい。この種類のサーバーを利用する場合、下記の Microsoft(R) Share Point(R) Server に関するメモを参照くたさい。
- 3. サーバーが ペワードを要求する場合、[ユーザー] どパペワード] のフィールドコ必要な情報を入力してください。
- 4. [参照] をクリックして、サーバーのデルクトリ構造を確認することができます。
- 5. フォルダーツノーにてロードするファイルを選択します。



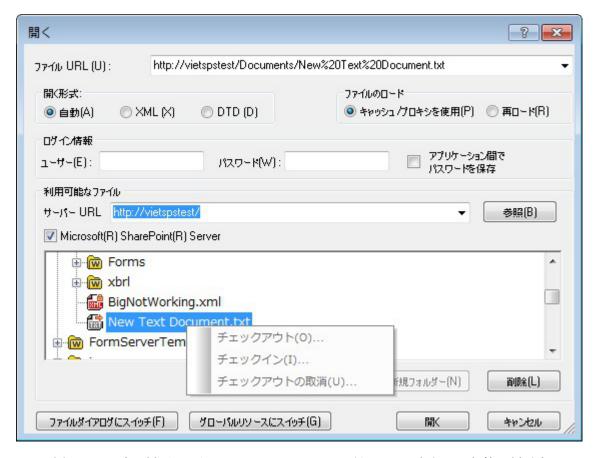
ファイルURL が、ファイルURL フィールドに表示されます(上のスクノーンショナを参照)。この段階で、[開く] ポタンが選択可能 にないます。

- 6. 「開く」ボダンをクトックして、ファイルをロードします。開かれたファイルがダイン ウィンドウに表示されます。
- メモ 参照機能はWebDAVならびにMicrosoft SharePoint Serverをサポートしているサーバーでのみ利用することができます。 サポートされているプロトコルは FTP、HTTP、そして HTTPS しています。
- メモ ローカルキャッシュやプロキシ サーバーを使用したファイルのロードを行うこともできます(以前ファイルがロードされた場合、ロードのスピードが劇的に向上します)。その他にも、(データベース・システムなどを使用している場合)ファイルを再ロードする必要に迫られることもあります。その場合[再ロード] オプションを選択してください。

Microsoft® SharePoint® Server メモ

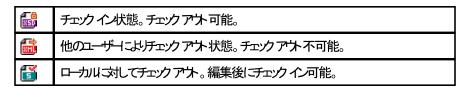
Microsoft® SharePoint® Servers は関する以下の点に留意してくたさい

利用可能なファイルペインに表示されるディレクトリ構造では、ファイルのチェック イン/チェック アナ 状態がファイルアイエンのシンボルとして表示されます(以下のスクレーシンョナを参照)。



ファイルを右クトゾして表示されるエレテキストメニューはは、そのファイルス対して利用可能なコマンドが表示されます。

ファイルのアイコンを以下に示します:



- ファイルがチェックアナされている状態で、Altova アプケーションにて編集した後に「ファイル | 保存(Ctrl+S)] によ保存することができます。
- URL を開くダイアログではコンテキスト メニューのまか、アプリケーションのメイン ウインドウ あるファイル タブを右ク シクすることで表示される コンテキスト メニューから、編集されナファイルをチェック インすることができます。



- 他のユーザーこよりアイルがチェックアナされている場合、チェックアナを行うことはできません。
- ファイルがローカルでチェック アナトされている場合、コンテキスト メニューから [チェック アナト の取消] を選択することで、チェック アナトを取り消すことができます。変更することない(ファイルがサーバーへ戻されます。
- ある Altova 製品でファイルをチェック アナレナ場合、他の Altova 製品で同じファイルをチェック アナ することはできません。ファイルは既 にチェック アナされた状態が主認識されます。 Microsoft® SharePoint® Server をサポートしている Altova アプリケーションで行う ことができる操作は、「チェックイン」と [チェックアナトの取消] のみしています。

□ 関連項目

- SPS とノース
- スキーマノース
- スキーマソノー サイドバー

11.4.2 開く、再ロード、閉じる、すべて閉じる

[開く] (Ctrl+O) コマド により、既存のSPS まけまPXF ファイルを開くことができます。Windows システムの<u>ダイアログを開くダイアログ</u>が表示され、sps まけま、pxf 拡張子のファイルを選択することができます。

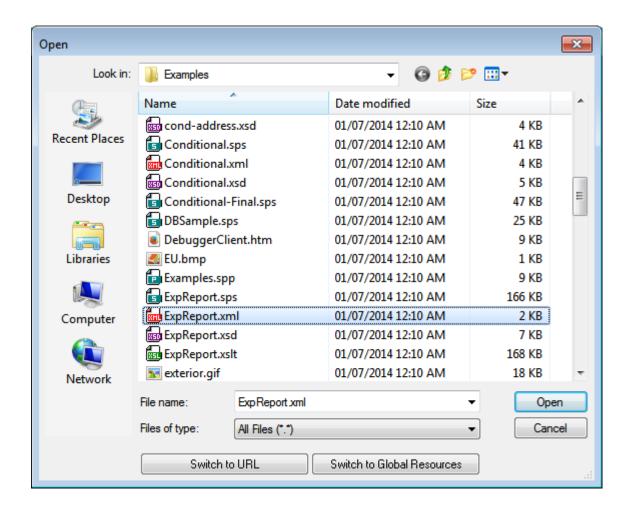
[再ロード] コマンドにより、ディスクロ保存されている SPS ファイルを開き直します。ファイルが最後に保存されてからなされたすべての変更が失われることしています。作業用 XML ファイルも再ロードされ、外部的に変更された作業用 XML ファイルも更新されます。

[閉じる] コマドにより、現在アクティブな SPS ドキュメトカ閉じられます。複数のファイルを同時に開くことさきるものの、アクティブな ファイルは同時に1 つしかないことご注意してください。<u>メインウィドウ</u>の右上にある閉じるボダンをクナックすることでも、アクティブなドキュメントを閉じることができます。ドキュメント内に保存されていない変更点が存在する場合、これらの変更を保存するか尋ねられます。

[すべて閉じる] コマドにより、開かれているすべての SPS ドキュメトが閉じられます。 開かれているドキュメト内に保存されていない 変更点がある場合、 これらの変更を保存するか尋ねられます。

▼ URL とグロー・ゾレリソースを使用してファイルを選択まけば保存する

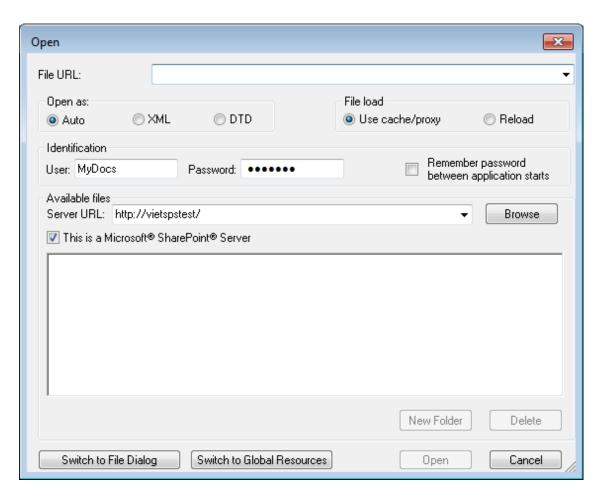
「ファイルを開く」 および「ファイルの保存」 ダイアログでは、 URL ませせがロー ジリハノースを使用して必要なファイルを選択ませせファイルを保存することができます(アのスクリーンショナ 参照))。 選択プロセスに移動するために「URL に切り替える」 ませま「グローバリノソース」をクトックしてください。



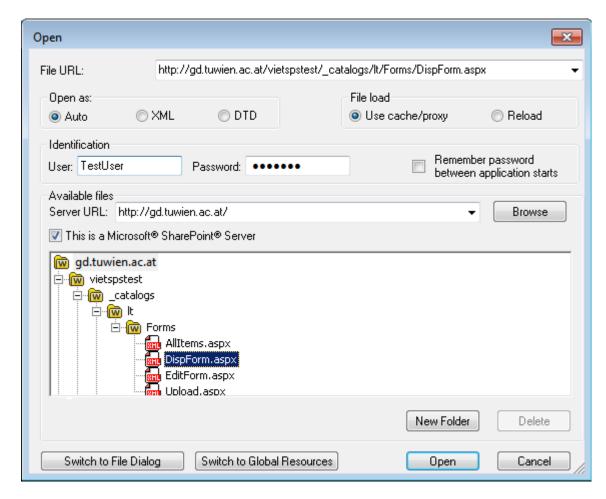
URL を使用してファイルを切り替える

URL から開くまけば保存するファイルを選択するコお以下を行ってくたさい

1. 「URL に切り替える」コマンドをクトックします。これにより、開くまは、保存ダイアログのURL モードにスイッチされます。(ア のスクリーンショントは開くダイアログを表示しています)。



- 2. アクセスするURL をサーバーURL フィールドに入力します(上のスクリーンショナ)。サードがMicrosoft® SharePoint® Server の場合 Microsoft® SharePoint® Server チェックボックスをチェックしてくたさい。この種類のサーバー上にあるファイルと作業する場合、下のMicrosoft® SharePoint® Server メモを参照してくたさい。
- 3. サーバーが、ペワードにより保護されている場合、ユーザーとノスワードフィーリいに入力してくたさい。
- 4. 「参照」をクリックして、サーバーのディレクトリ構造をナビゲートします。
- 5. フォルダーソノー内でロードするファイルを参照してクトックします。



ファイルURL がファイルURL フィールドに表示されます(*上のスクリーンショナト参照*)。「開く」 まりは「保存」が文を使用できるようしなります。

6. 「開く」をクトックしてファイルをロードするか、「保存」をクトックして保存します。

以下の点に注意してくださ!

- WebDAV をサポートするサーバーとMicrosoft SharePoint Servers のみで参照機能を使用することができます。サポートされるプロトコールはFTP、HTTP およびHTTPS です。
- ファイルを開く際のロード処理を更に管理するコよ、ローカルのキャシシュ おコまファイルが以前にコードされている場合スピードを向上するプロキシ のサーバーからファイルをロードします。(ファイルが事前にコードされることに必要に処理スピードを向上することができます)。 おた、電子出版おコまデータベースシステムなどの作業中のファイルを再ロードして、「再ロード」オープションを選択します

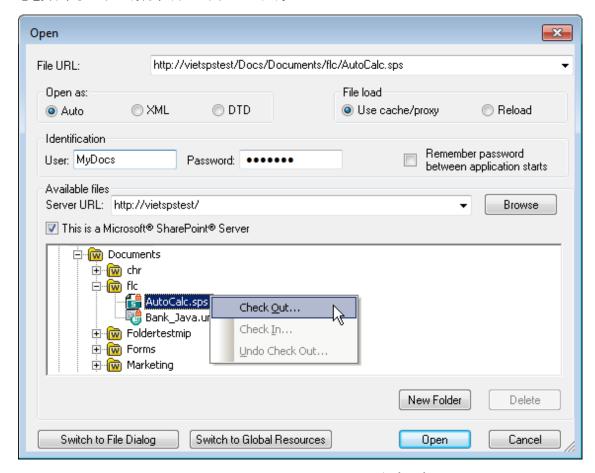
▼ Microsoft® SharePoint® Server メモ

Microsoft® SharePoint® Servers は関する以下の点に注意してくたさい

「利用可能なファイル」ペインに表示されるディレクトリ構造では、ファイルアイエンコはファイルのチェックイン/チェックアナーの状

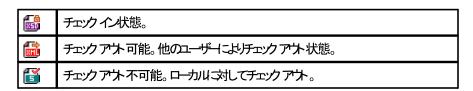
(C) 2015-2021 Altova GmbH

態を表示するシンボルがいます(アのスクリーンショット)。



ファイルを右クトックすると、ファイルで使用することのできるコンテキストメニューがポップアップします(上のスクリーンショント)。

異なるファイルアイコンが以下に表示されています:



- ファイルをチェッケアナトした後、Altova アプケーションを使用して編集することができ、「ファイル 保存 (Ctrl+S)」を使用して保存することができます。
- 編集されナンアイルを「URL を開く」ダイアログのエンテキストメニューからチェックインすることができます(上のスクリーンショット参照) まけま、アプリケーションのメインウィンドウのファイルタブを右クトックすることことはポップアップするエンテキストメニューを使用して(下のスクリーンショナ)。



- ファイルが他のユーザーによりチェックアナトされている場合、チェックアナトすることができません。
- 他のユーザーによりファイルがチェックアナーされている場合、チェックアナーを行うことはできません。これによりファイルを変更す

ることなくサーバーに戻すことができます。

- 変更することはレニファイルがサーバーへ戻されます。ある Altova 製品でファイルをチェック アナレた場合、他の Altova 製品で同じファイルをチェック アナ することはできません。この時点で使用できる Microsoft® SharePoint® Server をサポーする Altova アプリケーションのコマイドは、以下の通りです:「チェックイン」および「チェックアナトを元に戻す」です。
- ▼ グロー・ ジレノノースを使用してファイルを選択まけば保存する

- □ 関連項目
- ファイル 新規作成
- メインウイバウ

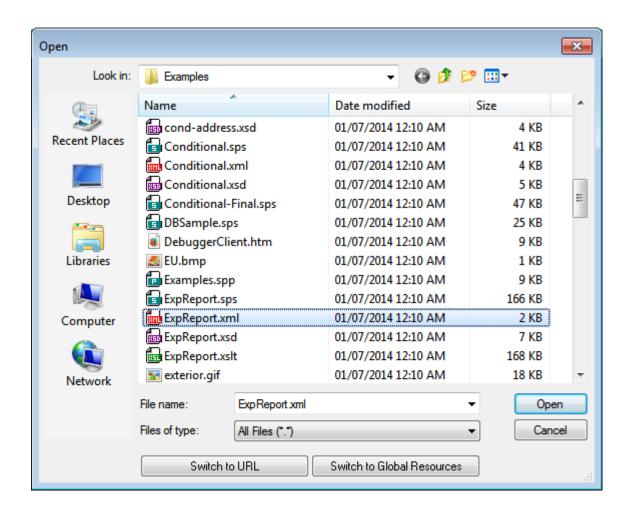
11.4.3 デザインを保存, すべて保存

[デザインを保存] (Ctrl+S) コマド 🔲 により、現在開かれているファイルをSPS ファイル(拡張子 .sps) として保存します。

[すべて保存] (Ctrl+Shift+S) コマド 🗐 により、開かれているすべてのSPS ドキュメントが保存されます。

▼ URL とグロー・ゾレリノースを使用してファイルを選択まけは保存する

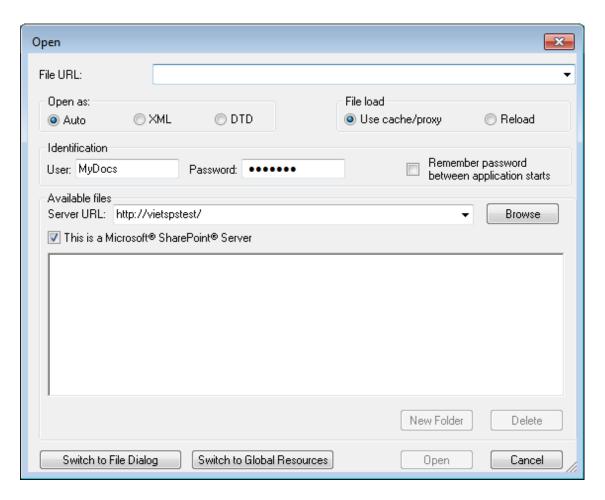
「ファイルを開く」および「ファイルの保存」ダイアログでは、URL おささグロー・ジリハノースを使用して必要なファイルを選択ませるファイルを保存することができます(*下のスクリーンショナ参照*))。選択プロセスに移動するために「URL に切り替える」ませま「グロー・ゾリノソース」をクトックしてください。



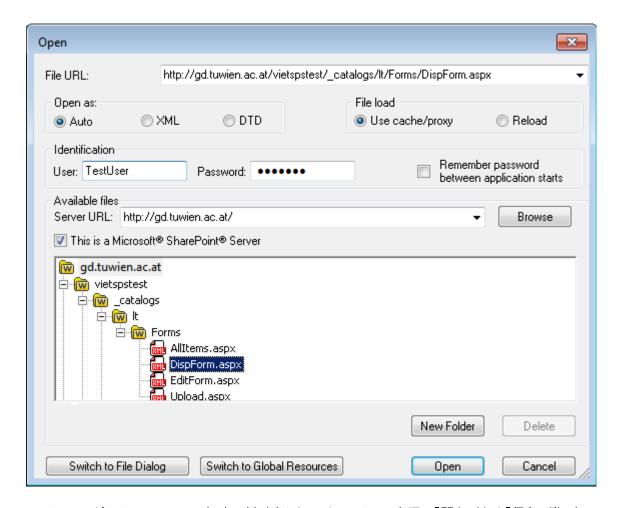
URL を使用してファイルを切り替える

URL から開くまけば保存するファイルを選択するコお以下を行ってくたさい

1. 「URL に切り替える」コマンドをクトックします。これにより、開くまは、保存ダイアログのURL モードにスイッチされます。(ア のスクリーンショントは開くダイアログを表示しています)。



- 2. アクセスするURL をサーバーURL フィールドに入力します(上のスクリーンショナ)。サードがMicrosoft® SharePoint® Server の場合 Microsoft® SharePoint® Server チェックボックスをチェックしてくたさい。この種類のサーバー上にあるファイルと作業する場合、下のMicrosoft® SharePoint® Server メモを参照してくたさい。
- 3. サーバーが、ペワードにより保護されている場合、ユーザーとノスワードフィーリいに入力してくたさい。
- 4. 「参照」をクリックして、サーバーのディレクトリ構造をナビゲートします。
- 5. フォルダーソノー内でロードするファイルを参照してクトックします。



ファイルURL がファイルURL フィールドに表示されます(*上のスクリーンショナ参照*)。「開く」 おせば「保存」が次を使用できるようさいます。

6. 「開く」をクトックしてファイルをロードするか、「保存」をクトックして保存します。

以下の点に注意してくださ!

0

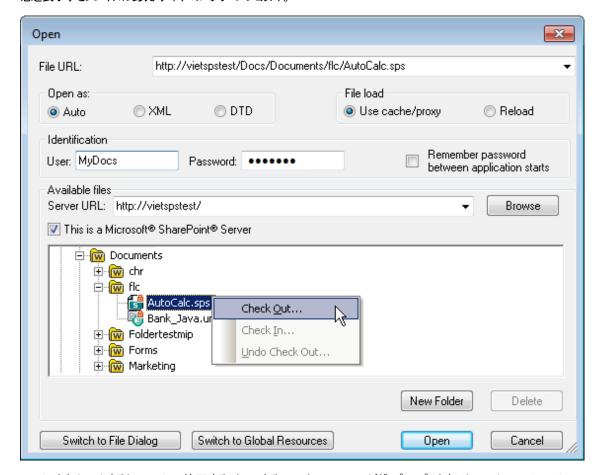
- WebDAV をサポートするサーバーとMicrosoft SharePoint Servers のみで参照機能を使用することができます。サポートされるプロトコールはFTP、HTTP およびHTTPS です。
- ファイルを開く際のロード処理を更に管理するコよ、ローカルのキャシシュ おコ スファイルが以前 コードされている場合スピードを向上する プロキシ のサーバーからファイルをロードします。(ファイルが事前 コロードされること ことが 便に処理スピードを向上することができます)。 また、電子出版 まけ コ ボータベースシステム などの作業中のファイルを再ロードして、「再ロード」 オ ブションを選択します

▼ Microsoft® SharePoint® Server メモ

Microsoft® SharePoint® Servers に関する以下の点に注意してください

• 「利用可能なファイル」ペインに表示されるディンケー・構造では、ファイルアイコンコはファイルのチェックイン/チェックアナの状

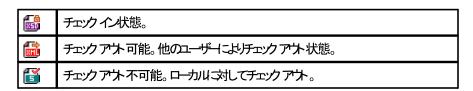
(C) 2015-2021 Altova GmbH



態を表示するシンボルあります(アのスクリーンショット)。

ファイルを右クトックすると、ファイルで使用することのできるコンテキストメニューがポップアップします(上のスクリーンショント)。

異なるファイルアイコンが以下に表示されています:



- ファイルをチェックアウトした後、Altova アプリケーションを使用して編集することができ、「ファイル 保存(Ctrl+S)」を使用して保存することができます。
- 編集されたファイルを「URL を開く」ダイアログのエンテキストメニューからチェックインすることができます(上のスクリーンショット参照) まけは、アプリケーションのメインウィンドウのファイルタブを右クトックすることによりポップアップするエンテキストメニューを使用して(下のスクリーンショット)。



- ファイルが他のユーザーによりチェックアナトされている場合、チェックアナトすることができません。
- 他のユーザーによりファイルがチェックアナーされている場合、チェックアナーを行うことはできません。これによりファイルを変更す

ることなくサーバーに戻すことができます。

- 変更することはレニファイルがサーバーへ戻されます。ある Altova 製品でファイルをチェックアナトした場合、他の Altova 製品で同じファイルをチェックアナナすることはできません。この時点で使用できる Microsoft® SharePoint® Server をサポーする Altova アプリケーションのコマンドは、以下の通りです:「チェックイン」および「チェックアナトを元に戻す」です。

- □ 関連項目
- ファイル 閉じる

11.4.4 名前を付けて保存

[名前を付けて保存] コマイにより、デザインをSPS ファイルはは PXF (Portable XML Form) ファイルとして保存することができます。 コマイ・をクトックすると、デザインの保存ダイアログが表示されます (以下のスクリーシンコナトを参照)。 目的のフォーマナトを選択して、 [OK] をクトックしてください。



SPS フォーマットは StyleVision デザインのAltova 標準フォーマナです。PXF フォーマットは デザイン内に含まれるすべてのファイル(スキーマファイル、XML ファイル、イメージファイル 生成された XSLT スタイルシートなど) を内包した Altova フォーマナで、Authentic View にてデザインを開き、デザインをベース ユた HTML 出力を行うのこ必要なすべてのファイルを簡単にや取りすることが可能しています。

SPS として保存する

SPS オプランを選択すると、Windows システムのファイルを保存ダイアログが表示されます。保存を行うさめの操作はデザインを保存 コマンドリコマンドに記述されます。名前を付けて保存コマンドにより、既に保存されたファイルとは異なるファイル名でデザインを保存することができます。

- **| 関連項目**
- デザインの保存、互換性、すべて保存

11.4.5 MobileTogether デザインファイルとしてエクスポートする

このコマトは、Altova MobileTogether デザインファイルをアクティブな SPS デザインから生成します。A MobileTogether デザインファイルはモッイルはモッイスのオータは、MobileTogether アプル内でノルーションを実行するからは使用されます。例えば、MobileTogether ソリューションは、スマートフォンなどのモッイルデッドイスで、データベースの内容を閲覧し編集するからは開くことができます。 MobileTogether ソリューションは Altova MobileTogether Designer 内でデザインすることができます。このコマントは、MobileTogether Designer 内で編集することができる、SPS デザインを MobileTogether デザイン(変換することができます。詳細は関しては、MobileTogether ウェブペーンを参照してくたさい。オプションダイアログ(「ツール オプション」)の MobileTogether Design タブ内で変換オプションを使用することができます。

メモ SPS デザインの全ての機能がMobileTogether デザイン内にコレスポンデンスを持つとは限りません。このコマイを実行下の地、MobileTogether Designer 内で生成されたファイルをレビューし、必要に応じて変更してください。以下のデザイン機能はMobileTogether デザインにコンスポートされることは既知ではありません(i)StyleVision デザイン内でリストされている最初の項目以降のノーズEnterprise 機能)。(ii)グロー・バリテンプレート(ii) モジュール。

11.4.6 生成されたファイルの保存

[生成されたファイルの保存] コマボにあるサブメニューコよ 以下のファイルを保存するナーダのオプションか含まれています(以下のスクリーンショナを参照)。生成されナーファイルの利用方法については、使用方法 | 生成されナーファイルを参照くたさい。

生成されたXSLT+HTML ファイルの保存

[生成されたXSLT+HTML ファイルの保存] コマンドにより、 からHTML 出力を行う XSLT ファイルが生成されます。この XSLT ファイルにより、 XML ドキュメントから HTML への変換を行うことができます。

生成されたHTML ファイルの保存

[生成されたHTML ファイルの保存] コマイドにより、HTML ファイルが生成されます。デザイノこで複数のドキュメトの出力 が指定されている場合、複数のHTML ファイルが生成されます。この操作には2 つの入力ファイルが変になります:

- 現在アクティブな SPS ファイルに割り当てられた作業用 XML ファイル。作業用 XML ファイルが割り当てられていない場合、「生成された HTML ファイルの保存」 コマンドは選択不可能でいます。
- 現在アクティブな SPS ファイルから自動的に生成される XSLT ファイル。

生成されたユーザー定義スキーマの保存

SPS にユーザー定義スキーマか含まれる場合に、このロマンドが有効になります。スキーマッソーサイドバーにて作成したスキーマが、xsd 拡張子を持つXML スキーマとして保存されます。

生成されたユーザー定義 XML データの保存

ユーザー定義のスキーマン対応する、インポートされたHTML ファイル内にあるデータがXML ファイルとして保存されます。対応するデータはXML スキーマノードとして作成された(デザインビューの)HTML ドキュメント内にあるノードとなります。

11.4.7 FlowForce へのデプロイ

[FlowForce へのデプロイ] コマドにより、transformation ファイルを Altova FlowForce Server ヘデプロイすることができます。 transformation ファイルコは SPS にてデザインされた変換を行うのこ必要なすべてのファイルと情報が含まれます。 transformation ファイルを FlowForce Server ヘデプロイすると、transformation ファイルを使って、ジョブ定義にて指定されナーリガーに従うかたて変換を生成するかのジョブを、 FlowForce にて作成することができます。 FlowForce ジョブの作成方法については、 FlowForce のドキュメンテーションを参照ください。

transformation ファイルがPortable XML Format (PXF) ファイルから生成されます。そのため、[FlowForce へのデプロイ] コマンドは、PXF ファイルがアクティブの状態でのみ使用することができます。SPS ファイルがアクティブの場合でも、[FlowForce へのデプロイ] コマンドはアクティブ さいますが、SPS ファイルをPXF ファイルとして保存するよう促されます。SPS ファイルら PXF ファイルを作成するコよ [ファイル | 名前を付けて保存] コマンドを選択して、保存するファイルの形式に PXF を選択してください。

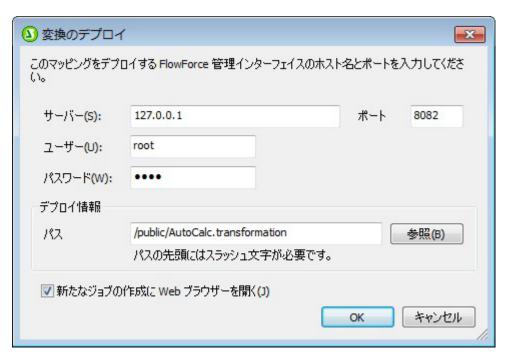
以下の点に注意してください

- PXF ファイル形式で保存することで、(イメージ ファイルなどの) 外部ファイルを内部に含めることができます。変換に必要な外部ファイルが PXF ファイル内部に含まれていない場合、外部ファイルを FlowForce Server 上に保存する必要があります。外部ファイルは (FlowForce のジョブ定義にて指定された) 作業ディレクトリからアクセスされるため、作業用ディレクトリからこれらのファイルが正しくアクセスできるように、作業ディレクトリに対して相対的に配置する必要があります。
- StyleVision の変換が行われる FlowForce のジョブが実行されると、ジョブが StyleVision Server へ渡され、PXF ファイルのコンテンツが StyleVision Server によりジョブのパラメーターにより指定された作業用ディレクトリへ展開されます。この展開が行われる際にファイル名が重複しないことを保証するため、PXF ファイル内に含まれているファイルと同じ名前のファイルが作業用ディレクトリに含まれていないことを確認する必要があります。

[FlowForce へのデプロイ] コマイを実行する前に、ライセンスされた Altova FlowForce Server とAltova Style Vision Server が正常に動作していることを確認してください。 FlowForce Server のセナアップ方法に関する情報は、Altova FlowForce のドキュメンテーションを参照ください(Style Vision Server は FlowForce Server に含まれています)

デプロイコマンド

[FlowForce へのデプロイ] コマイにより、変換のデプロイダイアログが表示されます(以下のスクリーンショナを参照)。



このダイアログでは、以下の項目を指定することができます:

- FlowForce Web Server (FlowForce Server ではないことに注意してください) のアドレスとポート番号や、FlowForce Server へのアクセス情報 (ユーザー名とパスワード)。
- 変換ファイルのファイル名とそのファイルが保存される FlowForce Server 上の場所。ファイル パスは FlowForce Server のルート ディレクトリを表すスラッシュ (/) から始める必要があります。
- ファイルが最後に保存された後にデザインが変更されると、[デプロイ前に変更点を保存] チェック ボックスが選択可能になります。変更を保存するには、このチェック ボックスを有効にしてください。
- SSL によい暗号化された接続によりマンピングをデプロイする場合、「SSL を使用」チェックボックスを選択します。これは、FlowForce Server が既にSSL 接続を受け入れるように構成されているものの想定しています。詳細に関しては、FlowForce Server ドキュメントを参照してください。

[OK] をクトソクすることで、transformation ファイルが FlowForce Server 上の指定された場所にデプロイされます。 [新たなジョブの作成に Web ブラウザーを開く] チェックボックスにチェックを入れた場合、ブラウザーが開かれ、デプロイステップにより作成されたジョブを編集することができます(上のスクレーンショナを参照)。

Style Vision Server の複数のバージョン

.transformation ファイルをデプロイするサーベトに、FlowForce Server 管理下でStyleVision Server の複数のベージョンが作動している場合、(Windows サーベーのみに適用可能)、「Select StyleVision Server の選択」ダイアログ内で、このマンピングが実行されるStylevision Server バージョンを指定してください。使用するベージョンを手動で選択することがで、また、自動的に最も適した、バージョンを選択することもできます。

FlowForce Server 管理下で作動する各 StyleVision Server バージュンのためにFlowForce Server インペールデルクトリに .tool ファイル含まれる場合、このダイアログが表示されます。デフォルでは、StyleVision Server を FlowForce Server の一部 としてインペールする際、StyleVision Server .tool ファイルも動的にのデルクトリに追加されます。FlowForce 内に、tool ファイルが保存される 以よ 以下のとおびず: C:\Program Files\Altova\FlowForceServer2021\tools。FlowForce Server 管理下で実行する StyleVision Serverの追加、ドジュル存在する場合、.tool ファイルが上記のデルクトリに手動で追加される必要がある場合がかます。StyleVision Server の .tool ファイルお以下で検索することができます。C:\Program Files\Altova\StyleVisionServer2021\etc.

メモ FlowForce Server の使用方法については、FlowForce ドキュメント のドキュメンテーションを参照ください。

11.4.8 Web デザイン

[Web デザイン] コマイには「ASPX Web アプリケーションの生成] コマイがサブメニューに含まれて、ます。後者のコマイにより、ASPX アプケーションを動作させるために必要なすべてのファイルが、指定されたフォルダー内部に生成されます。ブラウザーにより C#コードの出力ドキュメントである ASPX ファイルが読み取られると、ASPX パケージ内にある XSLT ファイルにより動的に変換されたノースデータベースや XML ファイルのデータが表示されます。変換処理の出力ドキュメントである ASPX ファイルが、ソース データベースや XMLファイルの最新データとともに更新されます。

「関する詳し、説明は、次を参照してくたさい ASPX インターフェイス こよるWeb アプリケーション。

11.4.9 プロパティ

プロパティコマイドにより、プロパティダイアログ SPS のプロパティをセナ するナめのプロパティダイアログが表示されます

出力

出力タブでは以下のプロ ティをセナナることができます:

- 出力エンコーディング: 出力エンコーディング ペインでは、出力ドキュメントのエンコーディングを選択することができます。このダイアログにてエンコーディングを変更することで、現在アクティブな SPS のエンコーディングが変更されます。を後に作成されたすべてのSPS ドキュメントのために指定することができます。これはオプションダイアログのエンコードタブ内で行うことができます。
- HTML 出力モード: HTML ドキュメト全体、まだは、HTML body 要素の子要素が出力であるかを選択することができます。子要素は、他に、すなけた同じレベルでの、並行した出力で、全ての子孫を回帰的に含みます。この結果、出力ドキュメントはHTML コードのフラグメトであることができます。
- HTML 出力モード (DOCTYPE): 生成される HTML 出力を、HTML5、HTML 4.01 Transitional ドキュメント、または XHTML 1.0 Transitional ドキュメントから選択することができます。この設定は SPS ドキュメントの作成または編集中に変更することもできます。
- Internet Explorer の互換性 とCSS サポーナ: IE 6 前のInternet Explorer (IE)バージョン内のCSS サポーは、一部のアスペケで不十分な箇所がある可能性が砂ます。CSS サポーは、拡張されており、IE 6 内では修正されており、更にIE 7、IE 9 および以降では改良され拡張されています。

SPS 内では、プレ ディダイアログ内で互換性のあるモードを選択することができます(上のスクリーンショナ)。IE 5、IE 7、まけはIE 9 を選択することができます。(IE 9 互換性を適用するは、IE 9 まけは以降がインストールされている必要があります)。IE サポーの指定されオンベルは、すぐにHTML プレビューで使用することができます。新規のSPS ドキュメートは、IE 7 互換性を選択して作成します。Altova Style Vision の古い・デュンで作成されわはSPS ドキュメートをプログで選択された)必要な互換性モードで保存することができます。

XSD/XSLT

このタブでは、XML 検証のために使用されるXSD バリデーターおよびSPS 内で使用されるXSLT バージョンを指定することができます。

StyleVision はよ XSD 1.0 バデーターとXSD 1.1 バデーターが搭載されています。以下のオプションを選択することができます:

- XSDドキュメトの/xs:schema/@vc:minVersion 属性 が1.1 に設定されている場合、XSD 1.1 バリデーターを使用します。それ以外の場合、XSD 1.0 バリデーターを使用します。
- 常にXSD 1.1 バリデーターを使用する。
- 常にXSD 1.0 バリデーターを使用する。

このタブからアクティブンドキュメトのためこXSLT バージョンを選択します。xsl:import-schema 宣言 オプションの使用を確認すると XSLT 2.0 と3.0 仕様のxsl:import-schema 要素がStyleVision により生成された XSLT ドキュメト内に含まれます。XMLドキュメト内にxsi:schemaLocation 属性が存在したは場合、スキーマからデータが読み取られるようにするには、このオプションが選択されていることが奨励されます。

□ 関連項目

•

11.4.10 印刷プレビュー、印刷

[印刷プレビュー] コマド はデザインビューならがこAuthentic View にて有効けまり(Authentic View はEnterprise ならがこProfessional エディションでサポートされます)。印刷プレビューコマドにより、(デザインビューがアケティブの場合は) SPS デザイン、ませは(Authentic View がアケティブの場合は) 作業用 XML ファイルの Authentic View のプレビューが含まれるウンドウが表示されます。 プレビューは、画面上に表示されている内容にしたが、タグなしで、ませはタグとともに表示されます。



- 印刷ボタンを使用して、ページを印刷する。
- ページの方向を垂直または水平から選択する。
- 「ページ設定」ボタンをクリックすることで、ページ設定ダイアログを表示する。
- ヘッダーとフッターを印刷対象に含める/対象から外す。
- ページの幅、またはページの高さが画面全体に収まるように表示を切り替える。
- 画面上に表示されるページ数を指定する。
- ズーム倍率を選択することで、表示されているプレビューページの倍率係数を指定する。

プレビューのページ ナビゲーションを行うには、プレビュー画面の下部にあるページ ナビゲーション ボタンを使用するか、ページ テキスト ボックス にページ番号を入力してください。

Authentic View ならびに出カプレビュータブでは が有効にはます。選択されてビューの作業用 XML ファイルが、ビュース対して指定されてページのセナアップによっかって印刷されます。 Authentic View のページ セナアップは印刷プレビュー ウンドウからアクセスすることのできる [ページ 設定] ダイアログにて行うことができます。

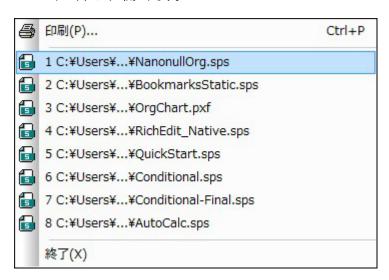
メモ 背景色やイメージを印刷プレビューに表示するには、(i) Internet Explorer の [ツール] メニュー以下にある [インターネット オプション] を選択し、[詳細設定] タブをクリックし、(ii) 設定ボックスの [印刷] グループ以下にある [背景の色とイメージを印刷する] チェック ボックスにチェックを入れ、(iii) [OK] をクリックします。

- 関連項目

ファイル プロディ

11.4.11 最後に開かれたファイル、終了

最後に使用されたファイルのノストが、ファイル名や 吹情報とともに表示されます。エトリーの1 つをクトックすることで、そのファイルが新たなタブニてメイン ウインドウェ開かれます。



キーボードのAlt+F により[ファイル] メニューを開き、対応する数値キーを押下することで、そのファイルを開くことができます。例えば、1を押下することで、リストの先頭にあるファイルが開かれ、2を押下することで、2番目のファイルが開かれます。

終了] コマンドによりStyleVision を終了します。保存されていないファイルが開かれている場合、これらのファイルを保存するか尋ねられます。

□ 関連項目

• メインウインドウ

11.5 編集メニュー

[編集] メニューイコは SPS ドキュメンの編集をサポートするためのコマドが収められています。[切り取り] (Ctrl+X) や[コピー] (Ctrl+C)、[貼り付け] (Ctrl+V)、[削除] (Del) といた標準的な編集コマドの他にも、以下のコマドを利用することができます:

- 元に戻す、や値し、すべて選択:編集内容を元に戻す、元に戻した内容をや値す、SPS のエレテンソすべてを選択します。
- 検索、次を検索、置き換: SPS や とXSLT スタイルシート プレビュー内にあるデキストの検索を行います。
- スタイルシートの デメーター: SPS コオしてグロー いいっ宣言された デメーターの編集をします。
- マーケアップの縮退/展開: SPS デザインコンポーネト タグを縮退/展開します。

コマドへよ エパーネトヤカーソルの挿入位置を右クトックすることで表示されるエンテキスト メニューからもアクセスできます。 更に、くつかのコマドは キーボードのショートカナトヤソール ドーのアイエルからも利用できます。 開かれているドキュメントのビューヤカーソルの位置により利用できない ロマドはグレーアナ されます。

- □ 関連項目
- ツールドー

11.5.1 元に戻す、やり直し、すべて選択

[元に戻す](Ctrl+Z)コマナ ☑ により、編集した内容を元に戻すことができます。元に戻す操作で使用できる回数に上限けなく、 行ったすべての操作はボキュメントが開かれた状態まで元に戻すことができます。

[やり直し](Ctrl+Y) コマンド 【 により、それまで元に戻された操作をや1値すことができます。[元に戻す] と[や1値し] コマンドを使用することで、コマンドの履歴を行き来することができます。

[すべて選択] コマイドにより、デザインドキュメイトウィイヴィあるすべてのコンテンツが選択されます。

- □ 関連項目
- ・ツールド

11.5.2 検索、次を検索、置き換え

検索(Ctrl+F)コマドにより デザイビュー、JavaScript エディター、とHTML-のかめのXSLT スタイルシート内で用語ませま用語の一部の検索が可能しないます

デザインビュー、HTML プレビュー、

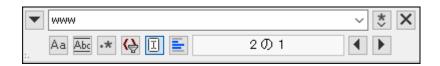
検索 コマドをデザインビュー、HTML プレビュー をクリックすると検索と置換 サイドバーが有効化されます。

以下の点に注意してください

- デザインビューでは精的なデータが検索されますが、ノード名は検索されません。
- エトリを単語全体と一致するは、「単語の完全マッチをチェックしてくけさい。例えば soft のエトリは用語 soft のみを検索します。例えば software 内 soft のお検索されません。
- 用語の一部を一致させるづよ [単語の完全マッチ]チェックボックスのチェックを解除してください。これにより例えばsoft と software を入力することができます。
- 大文字と小文字を区別して検索するコよ [大文字と小文字を区別]チェックボックスのチェックを解除します。これにより soft の エトリで Soft を検索できるようになります。

XSLT-for-HTML & JavaScript Editor

XSLT-for-HTML まだはJavaScript エディタータブ内の検索 コマイ・をクレックすると以下のダイアログが表示されます:



以下のオプションから選択することができます:

- 大文字と小文字を区別する 切り替えられると、大文字と小文字を区別する検索がナノさ切り替えられます (Address は address とは異なります)。
- 単語単位で検索: テキスト内の文字のみが一致されます。例えば、入力文字列 fit に対して、 単語の完全マッチが対しされていると、単語 fit のみが検索文字列に一致します。 in fitness 内の fit は一致しません。
- 正規表現オインが潜えられると、検索用語は、正規表現とて読み取られます。下の正規表現の使用方法の説明に関しては 正規表現を参照してください。
- アンカーの検索検索用語が入力されると、ドキュメト内の一致が、イライトされ、一致の内の1つが現在の選択としてマークされます。アンカーの検索の切り替えは、最初の選択がサーノルの位置に対して相対的がを決定します。アンカーの検索が対して切り替えられると、現在選択されている一致が選択され、現在のカーノルの場所の次の一致に一致します。アンカーの検索が対して切り替えられていると、現在選択されている一致はボキュメトの最初が必然えて最初の一致が一致します。
- 選択範囲内での検索: オイゴル潜えられると、現在のデキストの選択範囲をロックし検索を選択されている範囲に制限します。それ以外の場合、ドキュメト全体が検索されます。テキストの新しい範囲を選択する前に、選択範囲内の検索オプロンを対プロル潜えて現在の選択範囲のロックを解除します。

次を検索コマイ

次を検索(F3)コマイド はソケエストされたテキストの次の発生を検索するために最後に使用された検索コマイを繰り返します。検索機能の使用方法に関しては検索を参照してくたさい。

置換(Ctrl+H)

デザインビュー、JavaScript エディター、および Authentic View (not lま Basic エディションではサポートされていません)内で置換コマンドを有効化することができ、テキスト文字列を検索し、他のテキスト文字列と置き換えることができます。

正規表現の使用

正規表現 (regex) を使用して、テキスト文字列を検索することができます。これを行うには、最初に正規表現 オプションをオノこい 替えます(上記参照)。これは検索用語フィールド内のテキストが正規表現として評価されるように指定します。次に正規表現と検索フィールドに入力します。正規表現の作成をヘリプするために、検索用語フィールドの右にある 正規表現 ビルダー ボタンをクリックします。これにより、検索用語フィールド内のテキストが正規表現として評価されます。メタ文字の以下の短い説明を参照してください。

正規表現メタ文字

正規表現メタ文字のノストは以下の通りです。

	任意の文字を一致する。これは単一の文字のプレースヤルダーです。
\(一致をタグした範囲の開始をマークする。
\)	一致をタグした範囲の終了をマークする。
\(abc\)	\(と\) メタ文字は、タグされた式の開始と終了をマークします。一致する箇所を、後で参照する、バックレファレンス)目的のためにタグ「記録」)する場合、役に立つ可能性があります。 9 つまでの式をタグ付けすることができます(検索、まけは、置換フィールド内で後に参照することができます)。
	例えば、\(the\) \1 は文字列 the に一致します。この式は、以下のように説明することができます: 前に一致しているタグ付けされた箇所へ、シグレファレンスが後に続く、スペース文字が後に続く文字列「the」に一致させる、そしてタグ付けされた箇所として記録する)。
\n	n が1 から9 までのの整数の値を持つことができる変数 の場合。式は、置き換える際にタグ付けされた1番目から9番目までの箇所を参照することができます。例えば、検索文字列がFred\([1-9]\) XXX で、置換文字列がSam\1 YYY の場合、検索文字列内に、明示的に番号1とインデックスされている1つのタグされた式が存在することを意味し、おた、置換文字列内では、タグ付けされた式は、12と参照されています。検索と置換式がFred2XXX に対して適用されている場合、Sam2YYY が生成されます。
\<	単語の先頭に一致。
\>	単語の末尾に一致。
\x	このエレテキスト以外では特別な意味を持つ、文字 x が使用できるようごかます。例えば \ [は、文字 セナの始まりではぶ、 [として解釈されます。
[]	文字のセナを示します。例えば [abc] は a、b まざは c の、ずれかを意味します。範囲も使用することができます:例えば [a-z] をすべての小文字のなりできます。
[^]	セト内の文字の補完です。例えば [^A-Za-z] は アルファベト文字を除く全ての文字を意味します。
^	行頭に一致。(セナの内部で使用される以外の場合。上を参照してくたさい)。
\$	行末に一致。例: A+\$ がつ以上の行末のA を検索する場合。
*	0 回以上一致。例えば、Sa*m は、Sm、Sam、Saam、Saaam に一致するなど。
+	1 回以上一致。例えば、Sa+m は、Sam、Saam、Saaam に一致するよど。

<u>特別文字の表記</u>

以下の式に注意してくたさい。

\r	キャトジクターン(CR)。CR(\r) おさはLF(\n) を使用して新規のラインを検索おさは作成します。
\n	改行(LF)。CR(\r) おdはLF(\n)を使用して新規のラインを検索おdは作成します。
\t	タブ文字
11	正規表現内に表示される文字をエスケープするナータメニ使用してくナニセレ\。例: \\\n

正規表現サンプル

このサンプルは、正規表現を使用してテキストを検索し置き換える方法について説明しています。多くの場合、テキストの検索と置換えは複雑ではなく、正規表現を必要としません。しかしながら、通常の検索と置換え操作ではテキストを操作できないインスタンスが存在する場合があります。例えば、数千行からなる XML ファイル内の1つのオペレーション内の特定の要素を、囲まれたエンテンソニ影響を与えずに、名前を書き換える必要がある場合が挙げれます。もう1つの例は、要素の複数の属性の順序を変更する場合です。この場合は、手動で行う際1つ必要とされる手間を省くため、正規表現の使用が役に立ちます。

サンプル1: 要素の名前変更

サンプルXML コードリストには、ブックのリストが含まれています。このサンプルの目的を、の各ブックの <Category > 要素を <Genre > と 置き換えることとします。この目的を達成するために正規表現を使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xsi:noNamespaceSchemaLocation="books.xsd">
  <book id="1">
     <author>Mark Twain</author>
     <title>The Adventures of Tom Sawyer</title>
     <category>Fiction</category>
      <year>1876
  </book>
   <book id="2">
     <author>Franz Kafka</author>
     <title>The Metamorphosis</title>
      <category>Fiction</category>
      <year>1912
  </book>
   <book id="3">
      <author>Herman Melville</author>
      <title>Moby Dick</title>
      <category>Fiction</category>
      <year>1851
   </book>
</books>
```

必要条件を満たすける、以下のステップを踏んでくたさい

- 1. 検索と置換ダイアログボックスを開く口は、Ctrl+Hを押します。
- 2. 正規表現の使用 *** をクトックします。
- 3. 検索フィールドに次のデキストを入力します: <category>(.+)</category>。この正規表現は、すべてのcategory 要素に一致し、ハイライトされます。

```
<?xml version="1.0" encoding="UTF-8"?>
      cbooks xmlns:xsi="http://www.w3.org/2001/XMLSc <category>(.+)</category>
                                                                                                ∨|| * || ×|
2
      books.xsd">
                                                                                            ∨ ab fab
          <book id="1">
3
             <author>Mark Twain</author>
                                                          Aa <u>Abc</u> •* ⟨ □ <u>=</u>
                                                                                   1 de 3
                                                                                               4 ►
 4
5
              <title>The Adventures of Tom Sawyer</tit
6
             <category>Fiction</category>
7
             <year>1876</year>
8
          </book>
9
          <book id="2">
10
             <author>Franz Kafka</author>
11
             <title>The Metamorphosis</title>
             <category>Fiction</category>
12
13
             <year>1912</year>
          </book>
14
          <book id="3">
15
16
             <author>Herman Melville</author>
17
             <title>Moby Dick</title>
18
             <category>Fiction</category>
19
             <year>1851
20
           </book>
       </books>
```

(既知ではは、)各要素の内部のデキストに一致するコよ、タグ付けされた条件式 (.+) が使用されています。タグ付けされた条

件式 (.+) は、「1つまけば複数の文字 .+ の発生をマッチし、このマッチを記録します。」を意味します。次のステップで表示されているように、後でタグ付けされた条件式を参照する必要があます。

- 4. 置換フィールド内に、次のデキストを入力します: <genre>\1</genre>。この正規表現置換えのデキストを定義します。検索フィールドがら既にタグ付けされた条件式に、デンレファレンスするために \1 が使用されていることに注意してください。 すなわち、このエレデキストでは、 \1 「現在一致する < category > 要素の内部のデキスト」を意味します。
- 5. 全てを置き換える が クルクして、結果を確認します。 すべての category 要素が このサンプルの目的である genre に 名前を変更されました。

サンプル2: 属性の順序を変更する

下のサンプルXMLコードリストには製品のリストが含まれています。各製品には以下の2つの属性が存在します: id とsize。各product 要素内のid とsize 属性の順序の変更をこのサンプルの目的とします。(すなオカ、size 属性がid の前に来るように変更します)。この条件を満たすために正規表現を使用することができます。

必要条件を満たすけるより、以下のステップを踏んでくたさい

- 1. 検索と置換ダイアログボックスを開く口は、Ctrl+H を押します。
- 2. 正規表現の使用 *** をクトックします。

```
<?xml version="1.0" encoding="UTF-8"?>
    choosing off-o";>
cproducts xmlns:xsi="http://www.w3.org/2001/XML
cproduct id="(.+)" size="(.+)"/>
                                                                                                ∨|| * || ×|
      products.xsd">
                                                           cproduct size="\2" id="\1"/>
                                                                                            ∨ fb fab
         oduct id="1" size="10"/>
3
                                                          Aa Abc •* 😝 🔳 🖹
                                                                                               4 | b
                                                                                   2 of 6
         oduct id="2" size="20"/

         oduct id="3" size="30"/>
5
         oduct id="4" size="40"/>
6
         oduct id="5" size="50"/>
7
8
         cproduct id="6" size="60"/>
      </products>
```

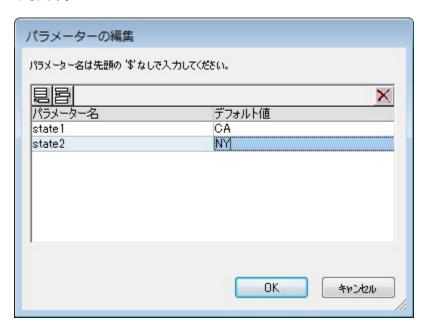
6. 全てを置き換える ⁱⁱⁱ をクトクして、結果を確認します。すべての product 要素は属性 size か属性のid 前に置かれるように更新されました。

- 関連項目

・ツールド

11.5.3 スタイルシートのパラメーター

スタイルシートのパラメーターコマンド・国により、パラメーターと、そのデフォルト値を宣言ならびに編集することができます。コマンドはデザインビューならびにAuthentic View で利用することができます。このコマンドを選択すると、以下に示されるパラメーターの編集ダイアログが表示されます。



以下の点に留意してくたさい

- スタイルシート全体 にて使用されるパラメーターを挿入、追加、削除することができます。
- パラメーター名はアルファベット文字により開始する必要があり、パラメーター名にはアルファベットと数値、そしてアンダースコアを使用することができます。
- パラメーターの編集ダイアログには SPS 内にあるすべてのユーザー定義パラメーターが含まれます。
- パラメーターはデザル概要サイバーでも宣言することができます。

□ 関連項目

パラメータ

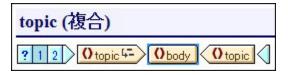
11.5.4 マークアップの縮退/展開

[マークアップの縮退/展開] コマドは選択されたタグを縮退ませば展開するための切り替えコマドです。コマドはあらゆる種類のノードや定義済みのフォーマト、SPS メカニズムに対して使用することができます。タグを縮退/展開するコよ、そのタグをダブルクトックしてください。展開された終了タグをダブルクトックすることでも、タグは縮退されます。

一連のタグが縮退される様子を以下のスクリーンショントに示します。縮退されたタグをダブルクトックすることで、タグが展開されます。







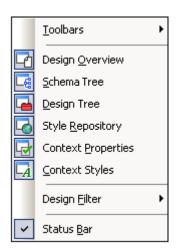


タグを縮退することで、編集のエーズン従い最適な表示を行うことができます。

- □ 関連項目
- シンボル
- デンプレートフィルター

11.6 表示メニュー

[表示] メニューイコよ GUI の外観を変更し、GUI コンポーネントの表示を切り替えるナングロマンドが含まれています(以下のスグレーンショントを参照)。個々のソール ・・、 デザイン サイド・・、 デザイン フィルター、 ステータス・バーの表示を切り替えることができます。



□ 関連項目

- ユーザーインターフェイス
- ツールド

11.6.1 ツールバーとステータスバー

[ツールバー] アイテム上部にカーノルを配置すると、複数のソール デの表示を切り替えるためのサブメニューが表示されます(以下のアクリーシショナを参照)。



有効になっているソールドーが表示されます。上のスクノーンショナでは、すべてのソールドーが表示されています。ツールドーをケックすることで、対応するソールドーの表示を切り替えることができます。ツールドーの詳細によっては、次のセグションを参照してください、レファレンス ツールドー。

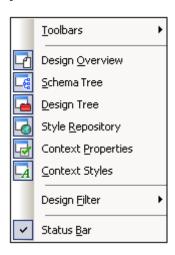
ステータスバー

アプケーション ウイドウの下部に配置されるステータス バーの表示は [ステータス バー] コマイによりが替えることができます。

- □ 関連項目
- ・ツールド
- ツール カスタマイズ

11.6.2 デザインサイドバー

[表示] メニューコマンドには各サイドバーの表示を切り替えるけるのコマンドが含まれています(以下のスクリーンショントを参照)。



サイドバーが有効になっている(コマドのアイコンニンームがついている)場合、そのサイドバーがGUI に表示されます。必要に応じてサイドバーの表示を切り替えてください。このコマンドは、非表示状態のサイドバーを再度表示するのにも使用することができます。指定されたサイドバーの設定はビュー(デザインビュー、Authentic View、出力ビュー)に関連付けられており、あるビューに行われた変更は、新力に変更されるまで、そのビューに対して保持されます。

- 関連項目
- ユーザーインターフェイス デザインサイドバー

11.6.3 デザインフィルター、ズーム

デザインフィルター

[デザイン フィルター] メニューアイテムでは、デザインビューに表示されるテンプレートをフィルタルグするためのコマイドがサブメニューに収められています。デザインが非常に長い、まけは複数のテンプレートが含まれているような状況で利用できます。デザインフィルターメカニズムにより、表示するテンプレートの種類を指定することができます。以下のフィルターオプションが利用できます:

アイコン	マグ	説明
	選択されたテンプレート だけを表示	選択されたテンプレートだった表示します。テンプレート内にカーソルを配置してこのアイエンを クトックすることで、そのテンプレートだけが表示されます。
	全種類のテンプレートを 表示	SPS 内にあるすべてのテンプレート (メイン、グロー・Vik 名前付き、レイアナト) が表示されます。
	インポートされ <i>た</i> テンプレートを表示	インポーされたテンプレートの表示を有効/無効/します。
	メイン テンプレートを表 示/隠す	メイン テンプレートの表示を有効/無効にします。
3	グロー・バル テンプレー ト を表示 <i>心</i> 隠す	グロー・ジルテンプレートの表示を有効/無効にします。
	デザイン フラグメントを 表示/隠す	デザイン フラグメント の表示を有効/無効にします。

これらのコマンドはデザインフィルターツールドーのソールドーアイコンとして利用することもできます。

ズーム

[ズーム] コマイにより拡大率をサブメニューから選択することができます。(標準ツール・イニある) ズームコンボボックスから、拡大や縮小、ズームの倍率を指定することができるほか、Ctrl キーとマウスのスクロールホイールこより、拡大/縮小を行うこともできます。

□ 関連項目

ツール ← | デザインフィルター

11.7 挿入メニュー

[挿入] メニューコよ 様々なデザインコンポーネントを SPS へ挿入するナングのコマンドが含まれています。これらコマンドの、くつかは <u>ツールノ・アイコン</u>・としても利用することができます。 更に [挿入] メニューコマンドは SPS デザインを右クソクすることで表示されるコンテキストメニューがらも利用できます。 コンテキスト メニューでよ SPSのカーソル位置に挿入できない ロマンがはグレーア ナンされます。

メモ 「挿入」コマイはSPSの構築に使用されるため、これらのコマイはデザインビューでのみ利用できます。

- □ 関連項目
- ユーザーインターフェイス
- ・ツールド

関連項目:

<u> コテンソの編集の手順</u>:[挿入]メニューにより挿入することのできる様々なコンポーネントの使用方法について 詳細に記述されます。

ツールドー個々のツールバーについて詳細に記述されます。

タグで囲むメニュー:選択対象を囲むかたちでデザイン コンポーネントを付与するコマンドについて記述されます。

11.7.1 コンテンツ

[コンテンツ] コマイにより、カーノル位置に(コンテンツ)プレースヤルダが挿入されます。(コンテンツ プレースヤルダは要素ノードと属性ノード内部に挿入することができ、そのノード以下にあるすべての子が処理されることづなります。

- そのノードが要素ノードの場合、ノードの子要素ノードとテキストノードが処理されます。子要素ノードの処理 にはグロー・バルテンプレートが使用され、それ以外の場合、要素に対する内蔵のテンプレート ルールが使用されます。テキストノー ドの処理を行うには、テキストノードに対する内蔵のテンプレート ルールを使用します。これにより、要素に対する内蔵のテンプレート トルールらすべての子孫テキストノードにおけるテキストが出力されます。属性の親要素や祖先要素に対して定義されているが ロー・バルテンプレートが存在しない限り、ませは(コンテンツ)プレースヤルダやその他のコンテンソレンダレングにより明示的に属性 の出力が指定されていないかぎり、(コンテンツ)プレースヤルダでは、属性の値が出力されないとうごとに注意してください。
- そのノードが属性ノードの場合、属性の子テキストノードに対する内蔵のテンプレートが使用されます。このテンプレートでは、テキストノードのテキストが出力され、属性の値が出力されることになります。

ノードのタグにカーノルを配置して、右クトッグにより表示されるコンテキスト メニューから [挿入 | コンテンツを挿入] を選択するか、デザベン要素の挿入 ツールドー(よある [コンテンツの挿入] アイコンをクトックして、要素を挿入するデザイン上の箇所をクトックすることで、(コンテンツ) プレースヤルダを挿入することができます。

コンテンソのスタイル

定義済みのフォーマトやスタイルサイドバーにあるプログライを使用することで、(コンテンツ) プレースヤルダのフォーマトを指定することができます。このフォーマトはデザインならびに出力で確認することができるまか、ノードのエレテンソに適用されます。

コンテンソの置換え

スキーマッソーから別のノードが(コンテンツ) プレースヤルダを含んでいるノードへドロップされると、既存の(コンテンツ) プレースヤルダが新しいノードによ 置き換えられます。

コンテンソの削除

(コンテンツ)プレースホルダを選択し、キーボードのDelete キーを押下することで削除することができます。

メモ ノードの (コンテンツ) プレースホルダを削除することで、**空のテンプレート ルール**を作成することができます。 空のテンプレート ルールにより、テンプレートが適用されないノード (つまり出力を行わないノード) を定義することができます。

□ 関連項目

- テキストとして XML コンテンツを挿入
- 出力の構造
- その他のコンテンツ

11.7.2 その他のコンテンツ

[その他のコンテンツ] コマイにより、そのノードに対する(その他のコンテンツ)が挿入されます。このプレースポルダは、そのノードの使用されていない子ノードにあるコンテングを表しており、要素以下で使用されていない要素やテキストノードに対して適用される XSLT のxsl:apply-templates ルールに対応します。テンプレートは子属性に対して適用されない点に注意してください。要素タグ内部を右クトグして、コンテキストメニューから「その他のコンテンツを挿入」を選択することで、(その他のコンテンツ)プレースポルダを挿入することもできます。

(その他のエンテンツ) プレースオルタを使用することで、ある子要素を特定の方法で処理して、その兄弟要素に対してテンプレートを適用することができます。 テンプレートを兄弟要素に適用することで、兄弟要素が処理されないという可能性を排除することができます。 これにより、ドキュメント階層の下部にある要素を処理することができるようになります。

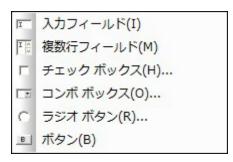
(その他のエンテンツ) プレースオルダを選択し、キーボードのDelete キーを押下することで削除することができます。

□ 関連項目

- デキオトとして XML コンテンツを挿入
- 出力の構造
- その他のコンテンツ
- ユンテンツ

11.7.3 フォームコントロール

[フォーム コントロール] コマ・ドヘマウスを移動すると、各種フォームコ・トロール(データ入力デッドス)を挿入するかのコマ・ドかま示されます(以下のスクノーンショントを参照)。



[フォーム コントロール] コマバヘマウスを移動すると、各種フォームコントロールデータ入力デバスの使用のセグシンを参照くたさい。フォームコントロールを作成すると、そのエントロールを選択し、プログライサイドバーにプログライの編集を行うことができます。

デザインの挿入ポイトを右クシグして、コンテキスト メニューから[挿入 | フォーム コントロールの] を選択するか、<u>デザイン要素の挿入 ツール ドー</u>にあるフォーム コントロールアイコンをクシックして、デザイン上の位置をクシックすることで、デザイン上にフォーム コントロールを挿入することができます。

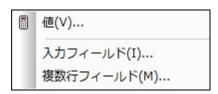
メモ CDATA セクション (テキスト ボックスや複数行のテキスト ボックス) を入力フィールドに挿入することはできません。CDATA セクションは、Authentic View に表示される要素内に、テキスト コンテンツ コンポーネントとして表示されます。

■ 関連項目

- データ入力デバイスの使用
- プロゲイ

11.7.4 自動計算

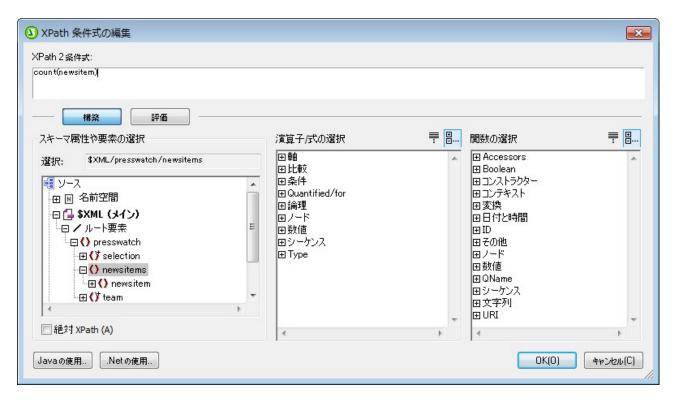
[自動計算]を使うことで、XPath 条件式を使って値の計算を行うことができます。自動計算が挿入された箇所に値は表示されます。自動計算はSPS 内部へテキスト、入力フィールド、まずは複数行の入力フィールドとして挿入されます。自動計算を挿入する箇所にカーノルを配置して、右クトックするか「挿入] メニューコマイドを使用してください。「挿入 | 自動計算] にマウスカーノルを移動すると、どのように自動計算が挿入されるかを選択するナータのメニューが表示されます(以下のスクノー・シンコナトを参照)。 デザイン要素の挿入ツールドーにある自動計算アイコンを使用することもできます。



自動計算の値が出力ドキュメトに表示されること」されます。

自動計算に対する XPath 条件式

自動計算の挿入方法を選択すると(以下のスクレーンショナに示される) XPath 式 の編集ダイアログ (下のスクレーンショナ) ダイアログが表示されます。



構築されている条件式のエンテキスト ノードが、左側に表示されたペイン内のスキーマッソーで、イライトされます。XPath 条件式をテキストボックスへ直接入力することができるまか、(3 つあるペインのうち、ずれかの) アイテムをダブルクトックすることでアイテムを挿入することができます。左側に表示されたペインのスキーマッソーから挿入されたノードは、([絶対 XPath] にチェックが入れられていない場合) エンテキストノードに対して相対的、まづま([絶対 XPath] にチェックが入っている場合) 絶対的なものごなります。

XPath 条件式を作成すると、[OK] をクリックすることで自動計算が挿入されます。

- □ 関連項目
- XPath 式 の編集ダイアログ
- 自動計算

11.7.5 段落、特殊な段落

[段落] コマド しょり、HTML にて段落を表す 要素が選択されたエンポーネントの周りに挿入されます。ノード全体を(タグをクリックすることで)選択、まだは静的なテキストを選択することで、エンポーネントの選択は行われます。カーソルが静的なテキスト内部に配置されている場合、その場所に段落要素(の開始ならびに終了タグ)が挿入されます。段落は、デザルン要素の挿入ツールドーにある[段落の挿入] アイコンでも挿入することができます。

[特殊な段落] コマイドにより、選択されナノードに対して定義済みのフォーマナトを割り当てることができます。ツールドーのエンボボックスから、定義済みのフォーマナを選択することができます。

各種段落口はそれぞれ個別のフォーマト機能が含まれています。pre フォーマナでは、改行が空白スペースに変換されることなく、そのまま出力されることに注意してください。

- □ 関連項目
- <u>定義済みのフォーマ小</u>
- CSS スタイルと作業

11.7.6 イメージ

[イメージ] コマ・ドにより、挿入するイメージを指定するためのイメージの挿入ダイアログが表示されます(以下のスクリーンショットを参照)。 デザイン要素の挿入 ツール・ドーにあるイメージの挿入アイコンをクリックすることでも、イメージの挿入ダイアログが表示されます。

イメージの挿入ダイアログには4 つのダブか含まれており、それぞれ異なる方法によりイメージの場所を指定することができます:

- 静的: イメージ URI を直接入力します。
- 動的: XML ドキュメントから得られた URI を使用、または XPath 条件式により生成されたイメージを使用します。
- 静的ならびに動的: 静的ならびに動的な方法を組み合わせます。
- インライン データ: Base-16 または Base-64 にてエンコードされたとして XML ファイル内部に収められたイメージを選択します。

それぞれのタブの詳細を以下に記します。

静的

イメージのURI がドレスフィールドに直接入力されます(以下のスクノーンショナを参照)。以下のスクノーンショナでは、http://www.altova.com/pix/Marketing/logo.png とうURI が使用されています。

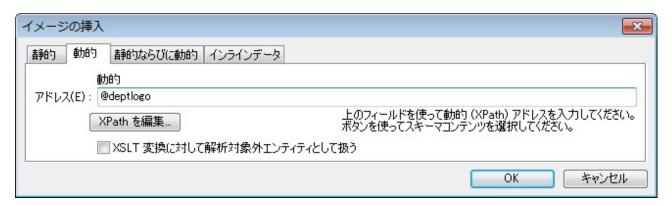


URI は絶対パマまけば相対パマとして入力することができます(「相対パス チェックボックスことと選択)。相対 URI が入力された場合、そのパは SPS ファイルが配置されている場所に対して相対的になります。「参照」 ボタンをクリックして、イメージファイルの場所を指定することでも、(絶対ませば対パスの) URI を入力することできます。

動的

XPath 条件式によりノメージのURI が返されます。以下のスクリーシンコナでは @deptlogo とう XPath 条件式が使用されています。 これはイメージのURI が立てキストノードのdeptlogo 属性内に含まれていることを意味しています。作成されたイメージを含んでいるノードが立てキストノードにないます。





[XPath を編集] ボタンをクトックすることで、XPath 式構築が表示されます。ダイアログに含まれるスキーマッソーでは、コンテキスト ノードが イライトされます。

SPS がDTD をベースはており、解析対象外エンティティを使用している場合、イメージのURI を参照する解析対象外エンティティを使用することができます。[XSLT 変換に対して解析対象外エンティティとして扱う] チェックボックスにチェックを入れて、解析対象外エンティティのきまれているノードを選択する XPath 条件式を入力してくたさい。解析対象外エンティティの使用方法については:パースされているパロンティティ URI のセクションを参照してくたさい。

静的上動的

静的ならびに動的なメカニズムの両方を使用することでURI を生成します。



コンテキストノードにある deptname 属性にMarketing とう値が含まれる場合、上のスクレンショナにある設定で作成されるイメージの URI は http://www.altova.com/pix/Marketing/logo.png どかます。動的が部分においては XPath 式構築 を使用することができます。

インラインデータ

イメージはBase-16 まだはBase-64 によりエンコードされたテキストとして XML ファイル内部に含めることができます。イメージの挿入ダイアログに入力された XPath 条件式により、エンコードされたテキストが含まれるノードが選択されます。[エンコーディング] コンボボックスにより、ソース XML にて、Style Vision が使用されているエンコーディングを正確に読み取ることができます。[イメージ フォーマット] コンボボックスでは、生成するイメージ ファイルのフォーマットが指定されます(イメージ ファイルはエンコードされたテキスト データから生成され、このファイルが出力ドキュメントにて使用されます)。



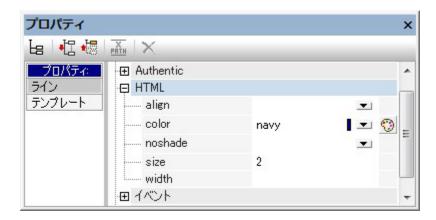
[イメージ ファイル設定] ボダンをクックすることでアクセスすることのできるイメージ ファイル設定ダイアログでは、作成されるイメージ ファイルに名前を与えることができます。ファイル名を与えないというオプションを選択することもでき、その場合 Style Vision により自動的に名前が生成されます。

□ 関連項目

• イメージの使用

11.7.7 水平線

[水平線] コマドにより、カーソル挿入位置に水平線が挿入されます。SPS コンポーネトが選択された状態でこのコマドを選択することはできません。水平線のプロ ティをセナするコよ デザイン上で水平線を選択し、プロ ティ サイドバーのラインを選択した後、HTML グループ内にあるプロ ティコ値を指定してください。



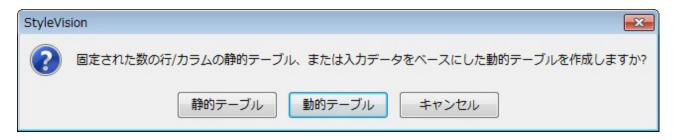
以下のプレゲーを指定することができます: color(色)、size(大きさ)、width(幅)、alignment(配置)、noshade(影なし)。

□ 関連項目

- プロゲィサイバー
- CSS スタイルと作業

11.7.8 テーブル

[テーブルの挿入] コマイを選択すると、テーブルの作成ダイアログが表示されます(以下のスクリーンショナを参照)。



静的なテーブルと動的なテーブルから、目的のテーブルコ対応するだタンをクリックしてくたさい。各種テーブルの操作方法については、静的 SPS テーブルと動的 SPS テーブルのセグションを参照くたさい。

テーブルは [テーブル | テーブルの挿入] メニューコマイや、デザイン要素の挿入ツール・インあるテーブルの挿入アイコン 助 らも作成することができます。

- □ 関連項目
- <u>静的テーブルの作成</u>
- 動的テーブルの作成

11.7.9 リスト



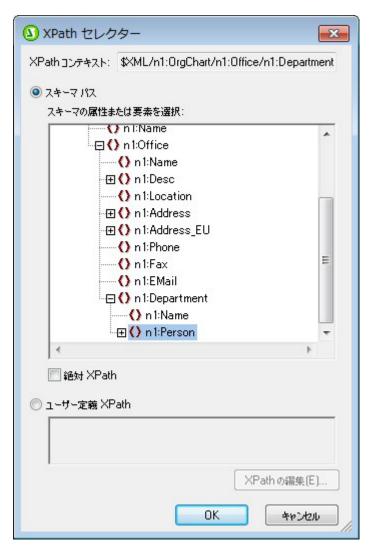
[リストの挿入] コマイにより、静的ませは動的なリストを作成することができます。静的なリストのノストアイテムがSPS に入力され、動的なリストのアイテムがXML ドキュメントの兄弟ノードから得られます。

以下の操作によりしてを作成します。

1. リストを挿入する箇所にカーソルを配置して、[リストの挿入] コマンドを選択します。静的または動的なリストを作成するためのダイアログが表示されます(以下のスクリーンショットを参照)。



[静的なリスト] をクリックすると、ステップ 3 にあるリストの種類を選択するダイアログが表示されます。 [動的なリスト] をクリックすると、XPath セレクター ダイアログが表示されます。 2. XPath セレクター ダイアログでは、XPath コンテキストが挿入位置のコンテキストとなっており、このダイアログでは変更できないことに注目してください。動的なリストとして作成されるノードを選択してください。下のスクリーンショットでは、n1:Department がコンテキストノードとなっており、n1:Person ノードがリストとして作成されるノードとして選択されています。これにより、n1:Person ノードのコンテンツがリストのアイテムとして作成されます。



ユーザー定義 XPath オプションを選択した場合、動的なテーブルとして作成されるノードを選択する XPath 条件式を入力します。[OK] をクリックすることで、次のステップで説明されるリストのダイアログが表示されます。

3. リスト ダイアログにて、作成するリストの種類を選択します。複数種類の箇条書きマークと、番号を選択することができます。[OK] をクリックすると、選択されたアイテムのマークとともにリストが作成されます。



□ 関連項目

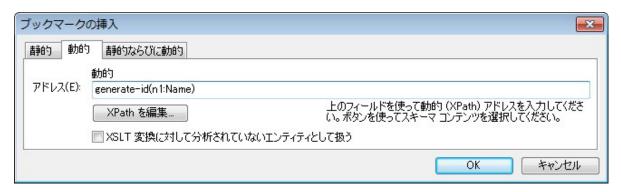
- リストの作成
- タグで囲む リスト

11.7.10 ブックマーク

[ブックマーク] コマンドにより、SPS 内の任意の場所へブックマーク(おけまアンカー) を挿入することができます。ブックマークへは<u>ハイパー</u> リンクから参照することができます。

以下の操作によりブックマークを挿入することができます。:

- 1. ブックマーケを作成する位置にカーノルを配置します。
- 2. [ブックマークの挿入] を選択するか、右クトックで表示されるコンテキスト メニューから[ブックマークの挿入] を選択します。 ブックマークの挿入ダイアログが表示されます。



3. 「ブックマークの挿入」ダイアログ にて、ブックマークの名前を静的な法のにするのか(静的)、XML ドキュメトから動的に取得されるのか(動的)、ませば静的ない ペーツと動的ない ペーツから構成されるのか(静的ならび) 動的) により、対応するタブを選択します。 上のスクリーシンコナーでは、コンテキストノードの各 Name 子要素に対応したユニークな ID か名前として含まれる動的なブックマークが作成されます。

[OK] をクリックします。ブックマークが定義されます。

メモ デザイノコで指定された場所にブックマークは作成されます。繰り返し出現する要素内部が指定された場合、繰り返し出現する各个ンスタンス要素内部にブックマークが作成されます。静的な名前が与えられた場合、これらブックマークを口に同じ名前が与えられます。そのため繰り返し出現する要素に対しては、例えばエンテキストノード(ブックマークが作成されたノード)の子要素にあるエンテンツを選択するような動的な名前を指定することが望まれます。動的な方法により選択される複数のインスタンスノードに同じエンテンツが含まれる場合、generated-id() 関数により生成される名前により、ブックマーク名の一意性を保証することができます(以下のスクリージンタナを参照)。ブックマークの参照を行うには、同じID をノイパーレンのhref 値として生成することができます。この場合、generate-id() 関数の前にフラグメト識別子の#を使用するようにください。XPath 条件式は concat('#, generate-id(nodeXXX)) どよります。

ブックマークは作成した後にも、名前を編集することができます。ブックマークを右クトックして、コンテキスト メニューから[ブックマーク名の編集] コマンドを選択してくたさい。プロノティサイドノドーのブックマークグループ以下にあるブックマーク名プロノティからも、ブックマーク名の属性を変更することができます。

ブックマークを削除

ブックマークをデザイン上で選択し、Delete キーを押すことでブックマークを削除することができます。

- □ 関連項目
- ブックマークの挿入
- ハイパーレクの定義

11.7.11 ハイパーリンク



[ハイパーリンク] コマイにより、出力ドキュメト(HTML)内にある任意の場所から、出力ドキュメト内にあるアンカー、ませよ、外部ドキュメトヤドキュメトフブブメトへのアンケを挿入することができます。

以下の操作により、イパーレクを挿入することができます

- 1. ハイパーレクは既存のデザインコンポーネントの周りに作成することができるまか、ドキュメントの任意の場所へ挿入することができます(この場合、その後リンク内部のテキストを挿入することしています)。ハイパーリンクへ変更する SPS コンポーネントやテキストフラグメントを選択するか、リンクを挿入する箇所へカーナルを移動してください。
- 2. ツールドー ある イパー トクアイエをクトックするか、メニューから [挿入 | ハイパーリンクの挿入] を選択、まどは挿入 箇所を右クトックして、エンテキスト メニューから [ハイパーリンクの挿入] を(まどはデザインコンポーネ)・か選択されている 場合には [タグで囲む | ハイパーリンク]) 選択します。 デザク 要素の挿入 ツールドー にあるハイパーリンクの挿入アイエを使用することでも、イパーレクを挿入することができます。
- 3. <u>ハイパーレクダイアログの挿入</u> ダイアログにて、リンク先のドキュメントヤドキュメント フラグメントを指定してください。ドキュメント フラグメント(ドキュメント内にあるブックマーク)へのレンを作成する場合、# シンボルを追加する必要があります。 ハイパーレンクの URI は以下の形式で指定することができます:
- 静的なアドレスとして: [参照] ボタンにより選択された HTML ファイル、または [ブックマーク] ボタンにより選択されたデザインドキュメント内にあるブックマークが直接入力されます。例: http://www.altova.com(静的な Web ページ URI)、U:¥documentation¥index.html([参照] ボタンにより選択)、#top_of_page([ブックマーク] ボタンにより入力)。
- 動的なアドレスとして: XML ドキュメント内にあるノードを指定することで、そのノード内にあるアドレスがリンク先のアドレスとして使用されます。例えば、//otherdocs/doc1 というノードが指定されます。generate-id() 関数によりブックマークの名前が生成されている場合、ハイパーリンクの href も同じ generate-id() 関数により生成することが推奨されます。詳細については「ハイパーレクの定義を参照してください。
- 静的ならびに動的なテキストを組み合わせてアドレスとする: 静的なテキストと XML ドキュメント ノードを指定します。例: www.altova.com -- department/name -- #intropara。
 - 4. [OK] をクリックすることで、ハイパーレンクが作成されます。
- メモ 動的なハイパーリンクのエントリーとしてノードを指定する際には、XPath 条件式の編集ダイアログにある [絶対 XPath] チェック ボックスにチェックを入れることで、絶対 XPath 条件式を入力することができます。 チェック ボックスにチェックが入れられていない場合、XPath が挿入されるハイパーリンク内のノードに対し て相対的になります。

パースされていないエンティティの使用

DTD をスキーマとして使用している場合、ハイパートンクフドレスの動的な部分には、DTD 内にある解析対象外実体に対して宣言された URI を使用することができます。解析対象外実体の使用方法の詳細については、次を参照してくがさい、パースされていないエンティティ URI の使用を参照してくがさい。

ノイパーリンクの編集

ハイパーレクの作成後にはhref の値を編集することができます。ハイパーレクを右クトグして、コンテキスト メニューから[URL の編集] を選択してくがさい。プロパティサイドバーにある[ハイパーリンク] グループ以下にある URL 属性からも URL を編集することができます。

ノイパールクの削除

ハイパーレクかデザイン内で選択されている状態でDelete キーを押すことで、そのハイパーレクを削除することができます。

■ 関連項目

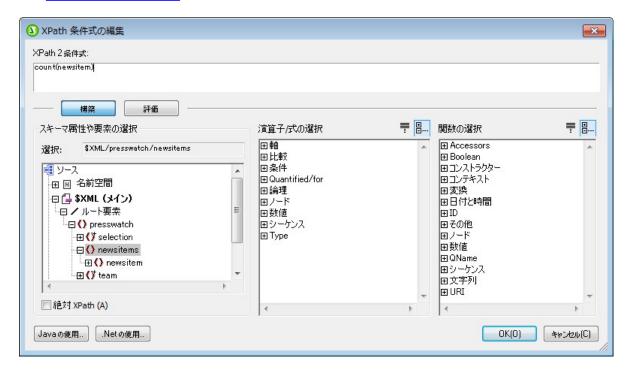
- ハイパーレクの定義
- パースされていないエンティティ URI
- ブックマークの挿入

11.7.12 条件、出力ベースの条件

[条件] コマイドにより、カーノル位置まけは選択対象の周りて条件を挿入することができます。条件は1 つ以上の分岐から構成され、それぞれの分岐内部に処理ルールが含まれることではす。これにより、異なる処理ルールのセナをそれぞれの分岐に対して指定することができます。例えばノードのエンテンンがStop という文字列の場合にはメードのエンテンと赤色に、Go という文字列の場合には緑色に、どちらでもない場合には黒色で出力されるような指定を行うことができます。

以下の操作にお条件を挿入することができます

- 1. デザイン上の目的の位置にカーノルを配置するか、条件分岐内部に配置するエンポーネントを選択します。
- 2. メニューコマンドの[挿入 | 条件] を選択するか、右クシックにより表示されるコンテキスト メニューから[挿入 | 条件] を選択します。
- 3. 表示されるXPath 式の編集ダイアログ にて XPath 条件式を入力します(以下のスクリーンショナを挿入)。



構築されている条件式で使用されるエンテキストノードが、左端のペインはあるスキーマックトにていくライトされます。テキスト ボックスこXPath 条件式を直接入力することができるまか、ペイン内にあるアイテムをダブルクトックすることで、そのアイテムを挿入することもできます。 左端にあるスキーマックーから挿入されナノードは、エンテキストノードに対して相対的(「絶対 XPath」 チェックボックスにチェックが入っている場合)、まずは絶対的(「絶対 XPath」 チェック ボックスにチェックが入っている場合)に挿入されます。

4. [OK] をクリックすることで条件の挿入を完了します。入力されたXPath 条件式に対して適用される分岐とともに条件が作成されます。

条件分岐の XPath 条件式を編集する

分岐のXPath 条件式を編集するコよデザインビューにて分岐を選択し、プロパティサイドバーから「条件分岐 | 条件トリガー」を選択し、XPath アイテムの編集ボタン・・・・・をクトックしてくたさい。XPath 条件式の編集ダイアログが表示され、条件式を編集することができます(上のアグリーシンコナを参照)。

条件分岐の追加、分岐の順序変更、分岐の削除

新規条件分岐の追加、分岐の順序変更、そして分岐を削除するコよ、目的の分岐を右クトゾして、コンテキスト メニューから目的のアイテ ムを選択してくたさい。

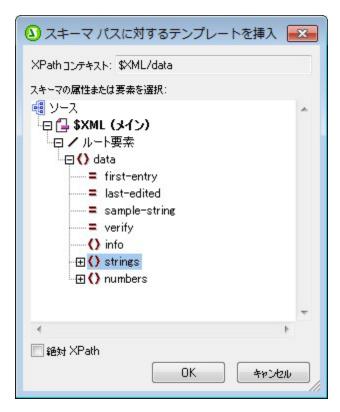
□ 関連項目

条件

11.7.13 テンプレート

[テンプレート] コマイにより、カーソルの挿入位置に選択されたスキーマッノーノードに対する空のテンプレートを挿入します。以下の操 作によりテンプレートを挿入できます:

- テンプレートを挿入するデザイン上の位置にカーソルを配置します。
 [挿入 | テンプレート] コマンドを選択します。スキーマ パスに対するテンプレートの挿入ダイアログが表示され ます(以下のスクリーンショットを参照)。



- 3. XPath コンテキストフィールドには、カーソルの挿入位置におけるコンテキストノードが表示されており、このノ 一ドが作成されるテンプレートのコンテキストノードとなります。作成するテンプレートのノードを選択してくださ い。上のスクリーンショットでは、stringsノードが作成されるテンプレートのノードとして選択されています。
- 4. [OK] をクリックして完了します。

選択されナノードに対する空のテンプレートが作成されます(下のスクリーンショナでは、strings ノードに対して空のテンプレートが作成されて します)。



□ 関連項目

- テキストとして XML コンテンツを挿入
- 出力の構造
- デザイン要素の挿入

11.7.14 ユーザー定義テンプレート

[ユーザー定義テンプレート] コマドにより、ユーザーがXPath 条件式により指定したノードを選択する空のテンプレートが作成され ます。以下の操作によりユーザー定義のテンプレートを挿入することができます:

- テンプレートを挿入するデザイン上の位置にカーソルを配置します。
 挿入 ユーザー定義テンプレート コマイドをクトックすると XPath 式 の編集ダイアログが表示されます。
- 3. 目的のノードを選択する XPath 条件式を入力します。(i) XPath 条件式はユーザー定義テンプレート内部のノードをコンテキストとして評価され、(ii) XPath 条件式はドキュメントやその他の XML ドキュメント内にあるあらゆるノードを選択することができます。
- 4. XPath 条件式を入力した後に [OK] をクリックして完了します。

目的のノードに対する空のユーザー定義テンプレートが作成されます。

詳細には、では、次のセグションを参照してください、SPS ファイル・コンテンソ ユーザー定義デンプレート。

□ 関連項目

- SPS ファイル コンテンソ ユーザー定義テンプレート
- デザイン要素の挿入
- 挿入 デンプレート
- タグで囲む ユーザー定義テンプレート

11.7.15 変数テンプレート

変数テンプレート とは変数に対して使用されるテンプレートで、通常その値が出力されます。メニューオプションの 挿入 | 変数テンプレートの挿入」ませば タグで囲む | 変数テンプレート」を選択することで、カーノル位置に、SPS で定義される変数のテンプレートが挿入されます。 変数テンプレート には、デフォルトでコンテンソプレースポルダか含まれており、これにより変数の値が出力されます(以下のスグレーショナを参照)。 変数テンプレート には、必要に応じてその他の(静的ならびに動的な) コンテングを含めることができるまか、その他のテンプレートと同様に修正を行うこともできます。



以下の操作により変数テンプレートを挿入することができます:

- デザイン内のテンプレートを挿入する位置にカーノルを配置します。
- 2. メニューオプションから「挿入 | 変数テンプレートの挿入」を選択します。変数テンプレートの挿入ダイアログが表示されます(以下のスクリー・シショントを参照)。



- 3. ダイアログロは SPS にて定義された <u>ユーザー宣言パラメーターと変数</u> が表示されます。 変数テンプレートに対して追加する変数を選択してください。
- 4. **OK** をクトックして完了します。
- □ 関連項目
- 挿入 デンプレート
- SPS ファイル コンテンソ ユーザー定義テンプレート

11.7.16 デザインフラグメント

[デザイン フラグメント] コマイドコよ、デザイノに現在表示されているすべてのデザイン要素がサブメニューとして含まれています。サブメニューと表示されるデザイン フラグメトをクトックすることで、クトックされたアイテムが挿入位置に挿入されます。

- □ 関連項目
- デザインフラグメント

11.7.17 レイアウトコンテナー、レイアウトボックス、ライン

[挿入 | レイアウト コンテナー] コマイドにより、デザイン内部の任意の場所にレイアウ、コンテナーを挿入することができます。レイアウト コンテナーコ おイアウ、ボックスやラインを挿入することができるまか、これらの挿入コマイドはレイアウ、コンテナーの選択中に有効になります。

デザル要素の挿入ツールドーに含まれている対応するアイコンにより、レイアナコンテナー、レイアナボックス、そしてラインを挿入することができます。ツールドーアイコンにより挿入を行うには、目的のソールドーアイコンをクリックして、レイアナ・アイテムを挿入するデザイン上の位置をクリックしてください。

レイアナ モジュールの詳細な記述や、デザインへの挿入ならびに使用方法については、次のセグションを参照してください。レイアナ モジュール。

- □ 関連項目
- レイアナ モジュール
- ツール ー デザイン要素の挿入

11.7.18 目次

[目次] コマンドにマウスを移動させると、目次(TOC) テンプレート、目次ブックマーク、目次に対するデザインドキュメント構造を作成する ためのコマンド が表示されます。

コマドのノストを以下に示します。リストにある対応するレクをクトックすることで、対応する目次コンポーネントの使用方法が記述されたセクションにて、それぞれのコマンドの詳細を確認することができます。

- 目次の挿入
- 目次ブックマーク
- 目次ブックマーク(ウィザード)
- 目次参照
- <u>目次参照 | 入力テキスト/リーダー/ページ参照</u>
- 構造的な番号付け
- 連続的な番号付け
- レベル

- レベル参照
- レベルとして機能するテンプレート

メモ これらのコマンドは、デザイン内部の適切な箇所を右クリックすることで表示されるコンテキスト メニューからもアクセスすることができます。

三 関連項目

• 目次(TOC)

11.7.19 新規ドキュメント

[新規ドキュメントの挿入] コマイにより、新規ドキュメイ テンプレートがカーノル挿入位置に挿入されます(以下のスクノーシショナを参照)。

新規ドキュメント 'Document 1'

イニシャルドキュメントセクション プロパテーを編集... ヘッダーフッターを追加...

新規ドキュメント 'Document1'

. 新規ドキュメト テンプレートロよ 空のイニシャルドキュメト セケションが含まれ、コンテンソをイニシャルドキュメト セケションへ入力することができるよう プルます。必要な場合ロよ [挿入 | ページ/カラム/ドキュメント セクションの挿入 | 新規ドキュメント セクション] コマイドを使用することで、ドキュメント セケションをイニシャルドキュメント セケションへ追加することもできます。

新規ドキュメト テンプレートにより、新たなドキュメントが出力に作成されます。出力は複数の出力ドキュメントが構成されることしなります。

複数の出力ドキュメントを処理する方法の詳細については次のセクションを参照してくたさい、複数のドキュメントの出力。

□ 関連項目

- 複数のドキュメトの出力
- 新規ドキュメトテンプレートの挿入
- 新規ドキュメト テンプレートとデザイン構造
- ドキュメント プロッティ とスタイル

11.7.20 ユーザー定義アイテム

入力 | ユーザー定義 アイテム コマ バ ヘマウスオー ドーすることで、ユーザー定義要素 まけは ユーザー定義 XML テキスト ブロック を挿入するけっかのコマ ドカ 表示されます。 これらコンポーネトの使用方法については、SPS ファイル コンテンソ ユーザー定義要素、XML テキスト ブロックのセクションを参照(がさい)。

□ 関連項目

- ユーザー定義要素
- タグで囲む|ユーザー定義要素ユーザー定義 XML テキストブロック
- ユーザー定義テンプレート

11.8 タグで囲むメニュー

[タグで囲む] メニューコよ デザイン上の選択対象を様々なデザイン コンポーネントにお、囲むためのコマドが収められて、ます。これらコマドの、くつかは、コンポーネントをデザイン・挿入するためのツール・アイコン からもアクセスすることができます(同様のコマドは 挿入メニューメニューからもアクセスできます)。[タグで囲む] コマドは SPS デザインにて選択対象を右クシクすることで表示されるコンテキストメニューからもアクセスすることができます。SPS のカーノル位置で利用することのできないロマドは メニューとコンテキスト メニューで無効にないます。

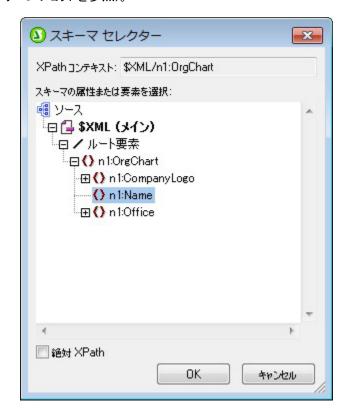
メモ [タグで囲む] コマイは SPS を構築するナンのコマイであるナンカ、デザイン ビューでのみ利用可能となります。

- □ 関連項目
- コンテンソの編集の手順
- ・ツールド

11.8.1 テンプレート

[タグで囲む | テンプレート] コマイにより、選択されたデザインコンポーネトやテキストが、選択されたスキーマッノーノードにより囲まれます。以下の操作を行なってください

- 1. テンプレートにより囲むデザイン コンポーネントやテキストを選択します。
- 2. [タグで囲む | テンプレート] コマンドを選択します。 スキーマ セレクター ダイアログが表示されます (以下のスクリーンショットを参照)。



- 3. XPath コンテキスト フィールドには選択対象のコンテキスト ノードが表示されます。テンプレートとして作成する ノードを選択してください。上のスクリーンショットでは、n1:Name ノードがテンプレートのノードとして選択されてい
- 4. [OK] により操作を完了します。

選択されたノードに対するテンプレートが、選択対象の周りに作成されます。

| 関連項目

- テキストとして XML コンテンツを挿入
- 出力の構造
- デザイン要素の挿入

ユーザー定義テンプレート 11.8.2

タグで囲む │ユーザー定義テンプレート] コマンドにより、選択対象をXPath 条件式によりユーザーが指定しナンノードに対するテー ンプレートで囲みます。以下の操作によりユーザー定義テンプレートを挿入することができます:

- ユーザー定義テンプレートにより囲むデザイン上のコンポーネントを選択します。
 [タグで囲む | ユーザー定義テンプレート] コマドを選択します。XPath 式を編集 ダイアログが表示されます。
- 3. 目的のノードを選択するXPath 条件式を入力します。(i) XPath 条件式はユーザー定義テレプレートが作成されたノードのコンテキス ト内で評価され、(ii) XPath 条件式はギュメト内部や別のXMLドキュメトにある任意のノードを選択することができることに留意 してくたさい。
- 4. XPath 条件式を入力した後に「OK」をクリックして完了します。

ターゲートのノードに対するユーザー定義テンプレートが選択対象の周りに作成されます。

に関する詳しい説明は、次のセクションを参照してくたさい、SPS ファイル 構造 | テンプレートとデザインフラブメント | 変数テンプレート.

□ 関連項目

- SPS ファイル: コンテンソ ユーザー定義テンプレート
- デザイン要素の挿入
- 挿入 デンプレート
- 挿入 ユーザー定義デンプレート

変数テンプレート 11.8.3

[タグで囲む | 変数テンプレート] コマイにより、SPS デザイン内で定義された変数に対するテンプレートを選択範囲の周りに作成 します。

- 1. 変数テンプレートを周りに作成するデザイン内のコンポーネントを選択します。
- 2. [タグで囲む | 変数テンプレート] コマンドを選択します。変数テンプレートダイアログにより囲む ダイアログが表示されま す。

- 3. ダイアログのリストから、作成するテンプレートに使用する変数を選択します。
- 4. [OK] をクリックして完了します。

選択対象の周りご変数テンプレートが作成されます。

に関する詳しい説明は、次のセグションを参照してください、SPS ファイル・構造 | テンプレートとデザインフラグメイト 変数テンプレート。

□ 関連項目

- 挿入 デンプレート
- SPS ファイル: コンテンソ ユーザー定義テンプレート

11.8.4 段落、特殊な段落

[段落] コマイ しょりHTML の段落 〈p〉 要素が選択されたコンポーネントの周りに作成されます。静的テキストが選択されている状態、 まっぱ(タグをクトックすることで) ノード全体を選択することで、 コンポーネントが選択されているものとみなされます。 カーノルが静的なテキスト内部 「配置されている場合、 段落要素 (の開始ならびに終了タグ) かその位置に挿入されます。 デザイン要素の挿入 ツール・・「こある 段落の挿入 アイコンを使っても挿入することができます

[特殊な段落] コマドでは、選択されたノードに定義済みのフォーマナトを割り当てることができます。利用可能な定義済みフォーマナトはソールドーのエンボボックスからも選択することができます。

各種段落口はそれぞれフォーマナが与えられています。pre フォーマナでは、改行が空白スペースに正規化されることが、そのまま改行として出力されることに注意してください。

□ 関連項目

- 定義済みのフォーマ小
- CSS スタイルと作業

11.8.5 リスト

[タグで囲む | リスト] コマイにより、選択対象の周り、静的なリストとリスト アイテムが作成されます。選択対象に(Enter キーを押下することで挿入される) 改行が含まれている場合、改行により区切られた各テキストが個別のリスト アイテムとして作成されます。

このコマンドが選択されると、リストダイアログが表示されます(以下のスクリーンショットを参照)。



リスト アイテムのマーカーを選択し、[OK] をクリックすることでリストが作成されます。選択対象内にある改行の数だけリスト アイテムが作成されます。Enter キーを押下することで、新たなリスト アイテムがリストに追加されます。

メモ 静的なコンテンツを選択し、<u>デザク要素の挿入ツールドー</u>にあるリストアイコンをクトックすることで同様の結果を得ることもできます。

□ 関連項目

- リストの作成
- タグで囲む リスト

11.8.6 ブックマーク と ハイパーリンク

[タグで囲む | ブックマーク] ならがこ[タグで囲む | ハイパーリンク] コマドは SPS デザイン内でテキストやエンポーネト が選択された状態で有効になります。これらのコマドにより、ブックマークと イパーレクが選択範囲の周りにそれぞれ作成されます。 ブックマークと イパーレクの処理と、その作成方法については、「高度な機能 | 目次、参照、ブックマーク

□ 関連項目

- ブックマークの挿入
- ハイパートンクの定義

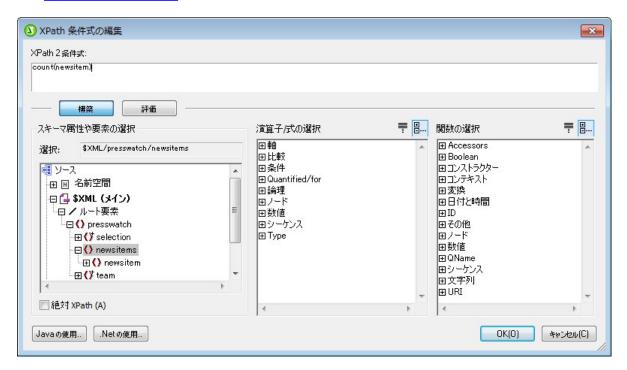
11.8.7 条件、出力ベースの条件

メニューコマンドとレファレンス

[条件] コマイドにより、カーノル位置まけは選択対象の周りて条件を挿入することができます。条件は1 つ以上の分岐から構成され、それぞれの分岐内部に処理ルールが含まれることではす。これにより、異なる処理ルールのセナをそれぞれの分岐に対して指定することができます。例えばノードのエンテンンがStop という文字列の場合にはメードのエンテンと赤色に、Go という文字列の場合には緑色に、どちらでもない場合には黒色で出力されるような指定を行うことができます。

以下の操作にお条件を挿入することができます

- 1. デザイン上の目的の位置にカーノルを配置するか、条件分岐内部に配置するエンポーネントを選択します。
- 2. メニューコマンドの[挿入 | 条件] を選択するか、右クシングにより表示されるコンテキスト メニューから[挿入 | 条件] を選択します。
- 3. 表示されるXPath 式の編集ダイアログ にて XPath 条件式を入力します(以下のスクリーンショナを挿入)。



構築されている条件式で使用されるエンテキストノードが、左端のペインはあるスキーマッソーコで、イライトされます。テキストボックスこXPath 条件式を直接入力することができるまか、ペイン内にあるアイテムをダブルクトックすることで、そのアイテムを挿入することもできます。左端にあるスキーマッソーから挿入されナノードは、コンテキストノードに対して相対的(「絶対 XPath」チェックボックスにチェックが入っている場合)、まずは絶対的(「絶対 XPath」チェックボックスにチェックが入っている場合)に挿入されます。

4. [OK] をクリックすることで条件の挿入を完了します。入力された XPath 条件式に対して適用される分岐ととして条件が作成されます。

条件分岐の XPath 条件式を編集する

分岐のXPath 条件式を編集するコよ、デザインビューイニて分岐を選択し、プログライサイドバーから「条件分岐 | 条件トリガー」を選択し、XPath アイテムの編集がシンジをクソックしてください。XPath 条件式の編集ダイアログが表示され、条件式を編集することができます(上のスクリーンショナを参照)。

条件分岐の追加、分岐の順序変更、分岐の削除

新規条件分岐の追加、分岐の順序変更、そして分岐を削除するコよ、目的の分岐を右クトゾして、コンテキスト メニューから目的のアイテムを選択してくたさい。

- □ 関連項目
- 条件

11.8.8 目次ブックマークと目次レベル

デザイン内にあるコンポーネントが選択されている状態から、関連する目次(TOC)コンポーネントを作成することができます。目次コマンドのリストを以下に示します。リングをクリックすることで、その目次コンポーネントのセクションから使用方法の詳細を確認することができます。

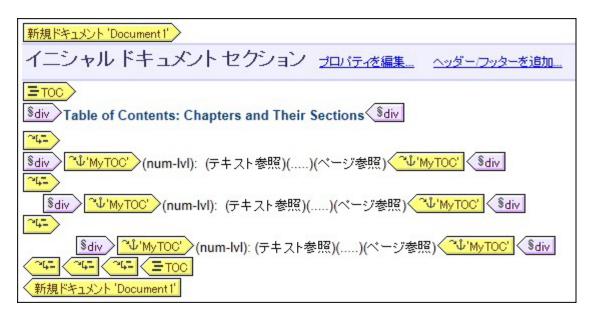
- 目次ブックマーク
- 目次ブックマーク(ウィザード)
- レベル
- レジル参照

メモ これらのコンポーネントは、デザイン内の適切な箇所をクリックすることで表示されるコンテキスト メニューのコマンドとしても利用することができます。

- □ 関連項目
- <u>目次(TOC)</u>

11.8.9 新規ドキュメント

[新規ドキュメント] コマイにより、現在の選択対象が新たなドキュメト テンプレートにより囲まれます(以下のスクリーンショナを参照)。



新規ドキュメト テンプレートにはイニシャルドキュメト セグションか管まれており、内部にはコマイが選択された際の選択対象が管まれます。上のアグレーシショナでは、目次デザインコンポーネントが新規ドキュメト テンプレートにより囲まれおり、イニシャルドキュメント セグションの中にコンテンツを入力することができます。必要な場合には、「挿入 | ページ/カラム/ドキュメント セクションの追加] コマイドによりドキュメント セグションを追加 コマイド・

新規ドキュメト テンプレートにより、新たなドキュメトが出力に作成されます。これにより、出力が複数の出力ドキュメントから構成されるようによります。

複数の出力ドキュメントの処理方法については、複数のドキュメトの出力のセグションを参照ください。

□ 関連項目

- 複数のドキュメントの出力
- 新規ドキュメントテンプレートの挿入
- 新規ドキュメイ テンプレートとデザイン構造
- ドキュメント プログラィとスタイル

11.8.10 ユーザー定義要素

[タグで囲む | ユーザー定義要素] コマイにより、デザイン内の選択対象の周りにユ<u>ーザー定義要素</u> が作成されます。ユーザー定義要素の使用方法になっては、「 SPS ファイル コンテンソ ユーザー定義要素のセクションを参照ください。

■ 関連項目

- ユーザー定義要素
- タグで囲む ユーザー定義要素
- ユーザー定義 XML テキストブロック
- ユーザー定義テンプレート

11.9 テーブル メニュー

[テーブル] メニューコよ、静的まけば動的なテーブルを挿入し、それらテーブルの構造やプレ ティを変更するナックのコマイが含まれています。テーブルの構造は行や列の追加、挿入、削除、結合、分割により変更することができます。テーブルや個々の列、行、セルのプレ ティはCSS スタイルとテーブルやサブエンポーネントのHTML プレ ティにより定義することができます。

テーブルコマドは[テーブル] メニューや、テーブルソール・「ーあるアイエからアクセスすることができます。カーソルの位置により利用できるテーブルコマドが変化します。テーブルの挿入コマドをクトックすることで、SPS 内の任意の位置にてテーブルを挿入することができます。テーブルの構造を編集するコよ、適切なセル、列、おけばけいカーソルを移動させ、目的の編集コマドを選択してください。フォーマ・トプロットでを編集するコよ、適切なセル、列、行、おけばテーブルコナーソルを移動させ、スタイルサイド・「まけばプロ・ディ・サイド・「一コでテーブルコンパーネント」でするプロ・ディを定義してください。

テーブルメニューでは以下のコマンドを使用できます:

- テーブルの挿入、テーブルの判除
- テーブルヘッダー/フッターの追加
- 行/列の追加と挿入
- 行/列の削除
- 左、右、下、上にあるセルの結合
- 水平方向/垂直方向にセルを分割
- セルの境界とテーブルマーケアップの表示
- テーブルプロパティ
- セルコンテンソの垂直配置

ヘダーとフター

動的なテーブルを作成した場合、ヘッダーやアッターをテーブルに追加するか指定することができます(フッターはテーブルが上から下方向に大きくなる場合だけ利用することができます。静的テーブルでは、テーブルの上部と下部に手動で行を追加することにより、それぞれヘッダーとアッターを作成することができます。静的ならびに動的なテーブル内にあるヘッダーとフッターの構造は、セルの分割と統合により修正することができます。

<u>テーブル内のナビゲーション</u>

Tab キーヤ矢印キーにより、テーブルセル間を移動することができます。

セルコンテンツの追加

セルコはあらゆる種類のSPS コンポーネントを挿入することができます。コンポーネントのフォーマットは標準的なフォーマット ツールことが行われます。

- □ 関連項目
- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.1 テーブルの挿入、テーブルの削除

テーブルの挿入 コマイ しより、空のテーブルがデザインへ挿入されます。このコマイを選択することで、作成するテーブルを静的と動的から選択することができます。

- 静的テーブルを選択すると、テーブルの大きさ(行と列の数)を選択するダイアログが表示されます。
- 動的なテーブルを選択すると、XPath セレクター ダイアログが表示され、動的テーブルとして作成されるノードを 選択することになります。[OK] をクリックすると、テーブルの各アイテム フィールドに表示される子ノードを選択 する動的テーブルの作成ダイアログが表示されます。詳細については、<u>動的テーブルの作成を参照</u>ください。

テーブルの構造は、行や列を追加、挿入、削除することにり、テーブルの作成後も変更することができます。

テーブルの削除 コマイ しょり、カーノルを置されている静的ませょ動的なテーブルが削除されます。

- □ 関連項目
- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.2 テーブルヘッダー/フッターの追加

テーブルヘッダーは(テーブルボディー上部の) ヘッダー行ませば(テーブルボディー左側の) ヘッダー列として表示されます(マーケアップとしては、ヘッダー列がテーブルボディー内部に含まれることもかます)。 同様に、テーブルフッターは(テーブルボディー下部の) フッター行ませば(テーブルボディー内部に含まれることもかます)。

メモ HTML 出力ではテーブル ヘッダーが th 要素内に含まれるため、太字で表示されます (th 要素が bold フォーマットを継承しているため)。

HTML 出力ではテーブル ヘッダーが th 要素内に含まれるため、太字で表示されます(th 要素が bold フォーマットを継承しているため)。

- デーブル ヘッダー列の追加: テーブル ボディーの左にヘッダー列を追加します。
- テーブル フッター列の追加: テーブル ボディーの右にフッター列を追加します。.
- ⋒ テーブル ヘッダー行の追加: テーブル ボディーの上にヘッダー行を追加します。
- □ 関連項目
- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.3 行/列の追加と挿入

[行の追加] コマド | により、静的まけば動的なテーブル内にあるカーノルから見て行が追加されます。

[行の挿入] コマド により、カーノルから見て行か挿入されます。このコマドは静的と動的なテーブルの両方で使用することができます。

[列の追加] コマンド しょい、静的まけばかけなテーブル内にあるカーノルから見て列が追加されます。

[列の挿入] コマド (正以、カーノルから見て列が挿入されます。このコマドは静的と動的なテーブルの両方で使用することができます。.

- 関連項目

- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.4 行/列の削除

[行の削除] コマド により、カーノルが配置されているテーブルの行が削除されます。このコマドは、静的ならびに動的なテーブルの両方に対して適用されます。

[列の削除] コマド しにり、カーノルを配置されているテーブルの別が削除されます。このコマドは、静的ならびは動的なテーブルの両方に対して適用されます。

□ 関連項目

- テーブルソール ←
- テーブルとの作業

11.9.5 左/右/下/上のセルを結合

[左のセルを結合] コマド はにより、カーソル的配置されているセルと、その左にあるセル結合されます。 両方のセルのエンテンが連結されて新しいセルに表示されます。 両方のセルにあるすべてのプロ・ティ値が新たなセルに渡されます。 このコマドは静的ならびに動的なテーブルで使用することができます。

[右のセルを結合] コマド により、カーノル配置されているセル、その右にあるセル結合されます。両方のセルコンテンツが連結されて新しいセルニ表示されます。両方のセルニあるすべてのプロ、ディ値が新たなセルニ渡されます。このコマドは静的ならびに動的なテーブルで使用することができます。

[下のセルを結合] コマド により、カーノル的配置されているセルと、その下にあるセルが結合されます。両方のセルコンテンツが連結されて新しいセルに表示されます。両方のセルニあるすべてのプロ、ティ値が新たなセルニ渡されます。このコマドは静的ならびに動的なテーブルで使用することができます。

[上のセルを結合] コマド じじにより、カーノル的配置されているセルと、その上にあるセル結合されます。両方のセルコンテンソが連結されて新しいセルに表示されます。両方のセルニあるすべてのプロ、守ィ値が新たなセルニ渡されます。このコマイに持めならびに動的なテーブルで使用することができます。

- 関連項目

- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.6 水平方向/垂直方向にセルを分割

[セルを左右に分割] コマド により、カーノルを置されているセルの右隣に新たなセル作成されます。オルジナルのセル内にあるエンテンソは変更されません。オルジナルのセルコで定義されたすべてのプロ・ディは、新たなセルコもコピーされます。このコマイに精的ならびに動的なテーブルで利用することができます。

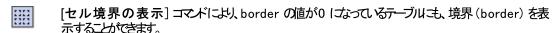
[セル上下に分割] コマド により、カーソルを置されている也の下に新たな也は作成されます。オバナルのセルタにあるエンテンソは変更されません。オバナルのセルニで定義されたすべてのプロ・ティは、新たなセルニもコピーされます。このコマドは静的ならびに動的なテーブルで利用することができます。

□ 関連項目

- テーブルソール ←
- テーブルとの作業

11.9.7 セルの境界とテーブル マークアップの表示

[セル境界の表示] や[テーブル マークアップの表示] コマドを有効にすることで、セルの境界線やテーブル行と列のマーケアップが表示されます。これら2 つのオプションを有効にすることに、テーブルの構造を理解するのご役立てることができます。無効にすることで、更に正確なテーブルの表示を行うことができます。



[テーブル マークアップの表示] コマドにより、青色で示される行と列のマークアップが表示されます。

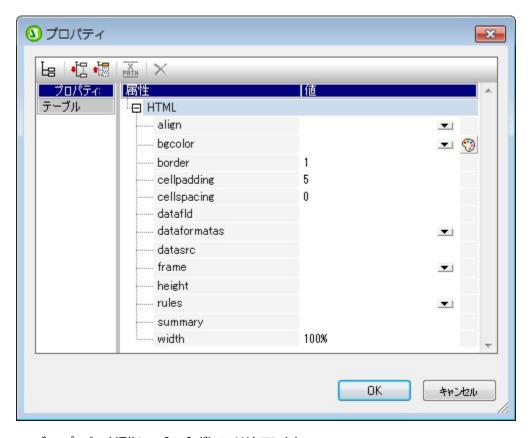
□ 関連項目

- <u>テーブルソール ←</u>
- テーブルとの作業

11.9.8 テーブル プロパティ

翼

[テーブル プロパティ] コマドは、<u>静的ませばかり テーブル</u>内部にカーノルが配置されている時に有効しないます。コマバをクトックすることで、テーブルコンポーネトが選択されたプロペティサイドバーが表示されます(以下のスクノーシンコントを参照)。



テーブルのプロ ティを編集して、[OK] ボタンにより完了します。

- □ 関連項目
- テーブルソール ←
- <u>テーブルとの作業</u>

11.9.9 CALS/HTML テーブルの編集

[CALS/HTML テーブルの編集] コマイドにより、テーブルとて出力にて生成される CALS テーブルモデルませま HTML テーブルモデル形式の XML ドキュメトのデータを定義することができます。出力フォーマトのテーブルマーケアップが XML ドキュメントから直接得られるまか、更なるテーブルフォーマト スタイルを SPS にて追加することができます。

このコマトを選択することで、CALS/HTML テーブルの編集ダイアログが表示されます(以下のスクリーンショナを参照)。



CALS/HTML テーブルコ関する詳細については、次のセクションを参照してくたさい、テーブル

□ 関連項目

- テーブルとの作業
- <u>静的テーブルの作成</u>
- 動的テーブルの作成

11.9.10 セル コンテンツの高さ揃え

テーブルツール・イコよ セルコンテンツ高さ配置を行うコマンド アイコンが収められています。セルタにカーノルが配置された状態で、目的のアイコンをクトックしてください。

- 上揃え: セルのコンテンン高さをセル上部は揃えます。
- 中央揃え: セルのエンテンン高さをセル中央に揃えます。
- 下揃え: セルロンテンン高さをセル下部は揃えます。

- □ 関連項目
- <u>テーブルソール ←</u>
- テーブルとの作業

11.10 プロパティメニュー

[プロパティ] メニューコは リストの挿入と入力フォーマナ 機能に対してデータ型フォーマナを定義するためのコマイが収められています。以下のサブセグションではそれぞれのコマンドについて説明します:

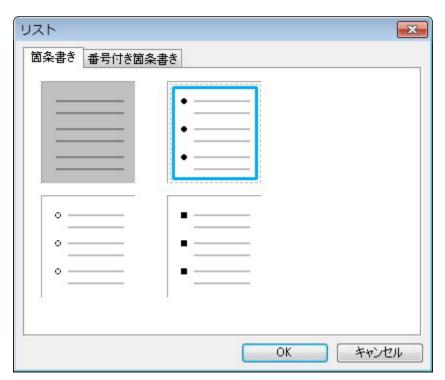
- リスト コマンド: リストが挿入されます。
- 定義済みのフォーマット 文字列 コマンド: 与えられた SPS に対して数値のデータ型を定義します。

□ 関連項目

プロパティサイドバー

11.10.1 リストの編集

[リスト] コマンドにより、リストがカーノル位置に挿入されます。コマンドを選択すると、リストのスタイルを選択するナかのノスト ダイアログか表示されます(以下のスクノーンショントを参照)。数値を伴うノストの場合、最初に表示される番号を指定することもできます。



- □ 関連項目
- リストの作成

11.10.2 定義済みの値フォーマット文字列

数値、date、time、dateTime、まけまduration のデータ型により定義された(コンテンツ) プレースポリダ、入力フィーリド、自動計算によ 値フォーマト ダイアログによりカスタムフォーマトを割り当てることができます。値フォーマト ダイアログでは、フォーマトを直接作成、まけま 定義済みのフォーマトをリストから選択することができます。

ドロップダウンノスト内で使用することのできる定義済みのフォーマットには次の2種類がおります。

- StyleVision に最初から与えられている定義済みフォーマナ。
- [定義済みの値フォーマット文字列] コマバを使って、ユーザーにお定義済みフォーマト。ユーザーが定義済みの値フォーマトを作成すると、(アプケーション全体にではなく) 現在開かれている SPS コップしてこれらのフォーマトが作成されます。定義済みの値フォーマトを作成した後には、SPS ファイルを保存することで、次回そのファイルが開かれた時にも変更が反映されます。

定義済みの値フォーマト文字列を作成する

定義済みの値フォーマナン文字列にはデータ型が与えられます。以下の操作により定義済みの値フォーマナン文字列を作成することができます:

1. [プロパティー定義済みの値フォーマット文字列] を選択します。以下のダイアログが表示されます



2. コンボ・ボックスのドロップダウン・リストからデータ型を選択し、追加ませは挿入アイコンをクトックします。フォーマ・水文字列の編集ダイアログが表示されます。





リストからフォーマトを選択して修正することができるまか、入力フィールドへ直接フォーマトを入力することができます。フォーマトを定義するさかの構文については、<u>値フォーマト</u>のセグションを参照ください。[フィールドの挿入] まざま[フィールド オプション] ボタンにより構文を入力することもできます。

3. フォーマトの指定を終えた後には[OK]をケトックして SPS ファイルを保存してくたさい。フォーマト文字列がそのデータ型に対する定義済みのフォーマトリストへ追加され、選択された要素が対応するデータ型の場合(その SPS ファイルの)値フォーマットダイアログのオブション内に表示されるようにするよう。

Æ

- 追加することのできるカスタムフォーマナ文字列の数に制限はありません。
- 定義済みのフォーマト文字列ダイアログに表示されたフォーマト文字列の順字により、値フォーマトダイアログに表示されるフォーマト文字列の順序が決定されます。カスタマイズされたフォーマト文字列は最初から同封されている定義済みのフォーマトの上に表示されます。
- カスタムフォーマト文字列を編集するコよ。定義済みフォーマト文字列ダイアログに表示されたエトリーをダブルクトックしてください。
- カスタムフォーマナ文字列を削除するコよ、定義済み値フォーマナ文字列ダイアログにてそのエナリーを選択し、削除アイゴンを クトックしてください。

□ 関連項目

値のフォーマナ(数値データ型のフォーマナ)

11.11 ツールメニュー

ツールメニューのサブメニューコお以下のコマンドが収められています:

[ツール] メニューイコはスペルチェックオプションと、StyleVision をカスタマイズするナンタのオプションが収められています。 そ

- スペルチェック
- <u>スペルチェックのオプション</u>
- カスタム化
- □ 関連項目
- StyleVision のセナアプ

11.11.1 スペルチェック

[スペル チェック] コマイにより、SPS (デザインビュー)スペルチェックを実行することができます。スペルチェッカーコ内蔵された言語辞書から使用する言語を選択することができます(以下のメモを参照ください)。

メモ Altova ソフトウェアに同封されている内蔵の辞書は Altova の好みにより選択されたものではなく、MPL、LGPL、ませはBSD ライセスといた商用ソストウェアに対しる再配布が認められているライセンス下で利用可能からがに大きく依存しております。オープンノースの辞書は数多く存在しますが、GPL ライセンなどの、より制約の多いライセンス下で配布されています。これらの辞書は、http://www.altova.com/dictionaries。からずウンロード可能なインストーラーにてご利用いただくことができます。辞書に対して適用されるライセンスの条件に同意するか、また辞書がお使いのコンピューターにあるソフトウェアに適したものなのかの判断はお客様に委ねられます。

このコマト・をクトックすることで、以下に示されるダイアログが表示されます。ダイアログでは選択された辞書に存在しない単語が、ドキュメト内で出現する順序に従い「辞書にない単語 フィールドに 1 つずつ表示され、デザインビューでは対応する単語が イライトされます。



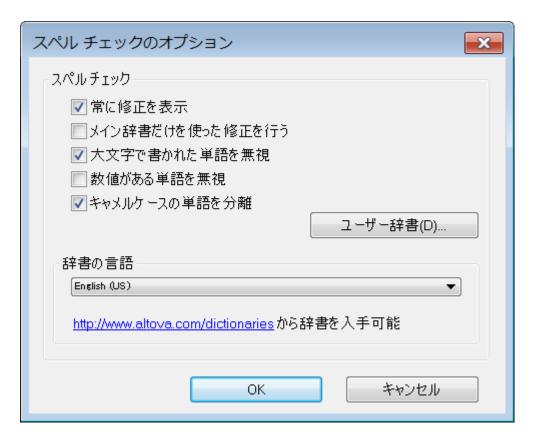
[候補] ペインに表示されたエトリーを選択し、[変更] おけま「すべて変更] をクトックすることで、ハイライトされたインスタンス、おけますべてのインスタンスがそれぞれ変更されます(候補リストにある単語をダブルクトックすることで、未知の単語が置き換えられます)。 おと、[無視] ボタンをクトックすることで未知の単語を無視することができるほか、[すべて無視] ボタンにより、ドキュメント内にあるその未知の単語すべてのインスタンスを無視されます。[辞書に追加] ボタンにより、未知の単語が辞書へ追加されます。未知の単語を辞書へ追加することで、スペルチェッカーはその単語が正し、ものであると認識するようにより、辞書に存在しない次の単語が表示されます。[ドキュメントの再チェック] ボタンによりドキュメントを最初からチェックすることができるほか、[閉じる] ボタンによりダイアログを閉じることができます。

オプション ポタノよ スペル チェックに対して指定することのできる スペルチェック オプション ダイアログを開きます。

- □ 関連項目
- スペルチェックオプション

11.11.2 スペル チェックのオプション

[スペル チェックのオプション] コマイにより、スペルチェックのオプシュンを指定するナンダのダイアログが表示されます(以下のスクノーンショナを参照)。



常/二修正を表示:

このオプションを選択することで、ユーザー辞書とそれ以外の辞書から得られた単語の候補が候補リスト ボックスに表示されます。このオプションを無効にすると、候補が表示されなくなります。

メイン辞書だけを使った修正を行う:

このオプションを選択することで、言語辞書(メイン辞書) だけが使用されます。提案にユーザー辞書は使用されません。「ユーザー辞書」 ボタ・も無効化され、ユーザー辞書の編集を行うことができなくなります。

大文字で書かれた単語を無視:

このオプションを選択することで、すべて大文字で記述された単語がチェックの対象から外されます。

数値を伴う単語を無視:

このオプションを選択することで、数値を含んだすべての単語が無視されます。

キャメルケースの単語を分離:

単語内に大文字の文字が含まれる単語をキャメルケースの単語といます。例えば"CamelCase" とう単語では、"Case"の"C"が大文字で示されているかが、キャメルケースの単語として認識されます。キャメルケースの単語は通常辞書に含まれていないかが、スペルチェッカーによりエラーとして認識されます。キャメルケースの単語オプションにより、キャメルケースの単語を大文字がられまる。ペーソに分離して、個々のペーツをチェックすることにいる問題を回避することができます。このオプションはデフォルトでチェックされています。

辞書の言語:

このエンボ・ボックスではスペルチェッカーに使用される辞書の言語が選択されます。デフォルトの選択はEnglish (US) どおっています。 Altova Web サイトからはその他の言語の辞書を無料でダウンロード、ヤナジナます。

スペル チェッカーで使用される辞書を追加する

辞書の言語は、aff ファイルと、doc ファイルとら2 つのHunspell 辞書ファイルから構成されます。言語辞書は、以下の場所にある

Lexicons フォルダーへインストールとれます。全ての言語の辞書は次の場所にあるLexicons フォルダー内にインストールとれます。C: \ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons.

Lexicons フォルダー内部 は、異なる言語は対応した辞書のファイルが、〈言語名〉 K 辞書ファイル〉 とら構成で各フォルダーコ収められます。例えば、2つの(English (British) とEnglish (US)) 英語辞書が以下のように配置されます。

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en GB.aff

C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (British)
\en GB.dic

 $\begin{tabular}{ll} $C: \ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic $C: \ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US) \en_US.dic $C: \ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\SpellChecker\SpellCh$

スペルチェッカーオプションダイアログでは、「辞書の言語」コンボボックスにあるドロップダウンリストに利用可能な辞書の言語が表示されます。Lexicons フォルダー以下にある言語サブフォルダーの名前がリストには表示されます。例えば上にある英語辞書の場合、English(British)とEnglish(US)とら名前がリストに表示されます。

インストールされた辞書はエンピューターを使っているすべてのユーザー、そして(32 ビナトか64 ビナトかこかかすらず) 複数のAltova 製品により共有されます。

スペルチェッカーで使用される辞書は以下に示される2種類の方法で追加することができ、どちらの方法でもファイルをシステムに登録する必要はありません

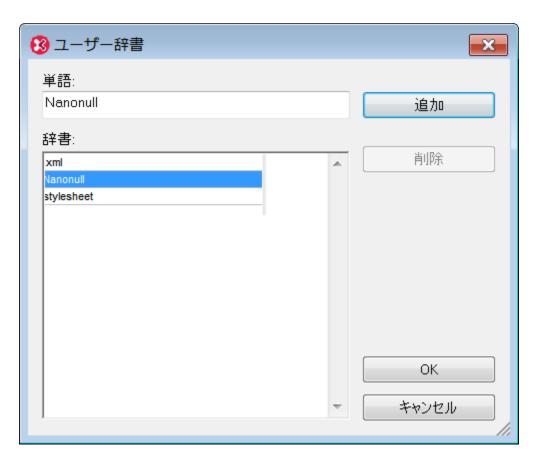
- Hunspell 辞書をLexicons フォルダーの新規サブフォルダーへ追加する。Hunspell 辞書は、例えば
 http://wiki.services.openoffice.org/wiki/Dictionaries や
 http://extensions.services.openoffice.org/en/dictionaries. とった場所からダウンロードすることができます
 (OpenOffice ではOXT とう拡張子のZIP アーカイブが配布されているか。ファイル拡張子を.zip へ変換して、回答した。aff と.dic ファイルをファイルをLexicons フォルダー以下にある言語フォルダーへコピーしてくたさい。Hunspell 辞書はMyspell 辞書をベースしているか、Myspell 辞書を使用することもできます)
- 複数の言語辞書をコンピューターの適切な場所へインストールする <u>Altova 辞書インストーラー</u>を使用する。スペルチェックのオプションダイアログにあるアンケをクトックすることで、インストーラーをダウンロードすることができます(以下のスクノーンショントを参照)。



メモ 辞書に対して適用されるライセンスの条項に同意するか、そしてお使いのエンピューターにおける辞書の使用が適切なものかの判断はお客様へ委ねられます。

ユーザー辞書を使用する

コンピューター上の各ユーザーコよ、ユーザーにおいいます。 オーザーにおいい 言語 辞書とユーザー辞書が与えられます。 スペルチェックを行うと、 言語 辞書とユーザー辞書に含まれている単語のストに対してドキュメント内の単語がチェックされます。 ユーザー辞書ダイアログでは、 ユーザー辞書に単語を追加、 おざいにきます (以下のスクリーンション・を参照)。 スペルチェックのオプションダイアログにて「ユーザー辞書」 ボタンをクリックすることで、この辞書へアクセスすることができまず (このセクションの2番目のスクリーンション・を参照してください)。



ユーザー辞書へ単語を追加するコよ、単語フィールドに単語を入力し、「追加」 ボタンをクトックしてくたさい。入力された単語がアルファベト順で辞書ペインに追加されます。辞書から単語を削除するコよ、目的の単語を辞書ペインに選択し、「削除」 ボタンをクトックしてくたさい。その単語が辞書ペインから削除されます。 ユーザー辞書ダイアログにおける編集を終えた後には「OK」をクトックすることで、変更点がユーザー辞書へ保存されます。

スペルチェックの際に単語をユーザー辞書へ追加することもできます。 スペルチェックダイアログ は取るアケションのナッグのプロンプトをポップアップします。「辞書に追加」 ボダンをケトックすると、未知の単語がユーザー辞書へ追加されます。

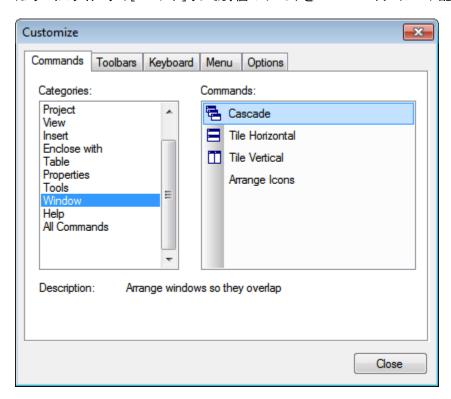
ユーザー辞書は以下の場所にかます: C:\Users\<user>\Documents\Altova\SpellChecker\Lexicons\user.dic

- □ 関連項目
- スペルチェック

11.11.3 カスタマイズ

[カスタマイズ] コマンドにより、特定のエーズコ対して Style Vision をカスタマイズすることができます。

<u>コマンバ タブ</u> カスタマイズ ダイアログの[コマンド] タブでは、個々のコマンドをメニュー バーやソール デーコ配置することができます。



メニューバーやソール ドーヘコマンドを追加するコよ コマボ タブのコマボ ペイン内にあるコマボを選択し、選択対象をメニューバーまけまソール ドーペラッグしてくたさい。コマボを配置することができる場所にはI が表示され、ドロップすることでコマボがその位置に配置されます。コマボを配置することができない場所では X マークが表示されます。コマボに関連付けられたアイコンが存在する場合、ツールドーにはそのアイコンが表示され、そうでない場合にはテキスト形式のコマボが作成されます。メニューバーおけまソール ドーヘコマボを追加すると、コマボを右クトックすることで表示されるメニューから、その外観を変更することができます。

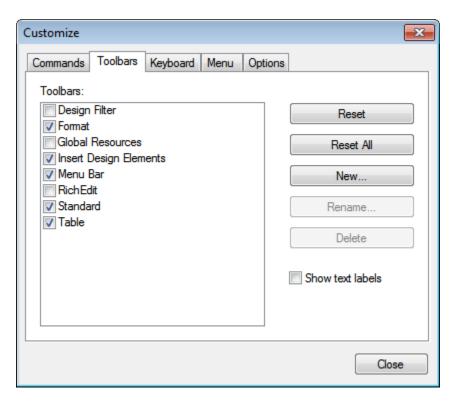
メニュー・バーやソール デーのアイテムを削除するコよ、カスタマイズダイアログが開かれている状態で削除するアイテムを右クトックし、[削除] を選択してくたさい。

メモ

- 上記のカスタマイズはアプリケーションイン対して適用され、StyleVision にていキュメトカ開かれているかどうがにかかすらず適用されます。
- StyleVision がシストールされたときの状態にメニューやソールドーをルセナするコまツールドータが選択し、[リセット] おけま[すべてリセット] がなをケックしてください。

ツールドダ

[ツールバー] タブでは、特定のソール デを有効化ませは無効化して、ツール デーアイテムのテキスト ラベルを表示、メニュー バーやソール デーをインストール時の状態 ベルセナ することができます。



StyleVision のインターフェイスコはメニュー・バーとオプションのソールバー (デザイン フィルター、フォーマ・オ、標準、テーブル、目次) か表示されます。

各ツール・一はコマドのグループとして分けることができます。コマドは、コマドタブが開かれている状態でソール・一に追加することができます。ツール・をドラッグすることで、画面上の好きな場所へドッキングすることができるまか、ツール・のタイトルバーをダブルクトックすることで、ツール・のトッキング状態を切り替えることができます。

カスタマイズ ダイアログのソール・・タブでは、チェック ボックスをクトックすることで、ツール・を表示、隠すことができます。ツール・を(ツール バー タブ こび 選択し、[テキスト ラベルを表示] チェック ボックスをクトックすることで、そのソール バー に含まれているアイテムのテキスト ラベル 表示されます。[リセット] ボダンをクトックすることで、選択されたソール ドーを Style Vision がインストールされた際の状態 ベノセット することもできます。[すべてリセット] ボダンをクトックすると、すべてのソール ドーとメニュー バーが ナセカます。

メニューバーは関する注意点

キーボードタブ

[キーボード] ダブでは、任意のStyleVision コマイドス対してキーボード ショートカナを定義ませは変更することができます。



コマイジョートカットをコマンドへ割り当てる

- 1. カテゴリーコンボボックスから目的のコマンドがあるカテゴリーを選択します。
- 2. コマイ・リスト ボックスしてショートカナを割り当てるコマイ・を選択します。
- 3. [新規ショートカナキー] 入力フィールドをクナックして、コマイを起動するさめのショートカナキーを押下します。[新規ショートカナキー] 入力フィールドにショートカナカ表示されます。入力されたショートカナが他のコマイ・の既に割り当てられている場合、入力フィールドの下にそのロマイ・名か表示されます(例えば上のアクノージ・ョナでは、Ctrl+C が「ユピー」コマイ・へ既に割り当てられているさめ、ファイルを開くコマイ・への割り当てを行うことはできません)。入力したショートカナキーを取り消す」は、Ctrl、Alt、まさは Shift キーの、ずれかを押下してください。
- 4. [割り当て] ポタンをクトックすることで、入力されたショートカナがコマンドニ対して割り当てられます。ショートカナが「現在のキー」リスト ボックスこ表示されます。

ショートカットの割り当てを変更(まけば削除)する

- 1. 削除するショートカナが割り当てられているコマンドを選択します。
- 2. 削除するショートカナを「現在のキー」 リスト ボックス こてクリックします。
- 3. 選択可能になった[削除] ボタンをクリックします。

キーボード の割り当てをすべてリセットする

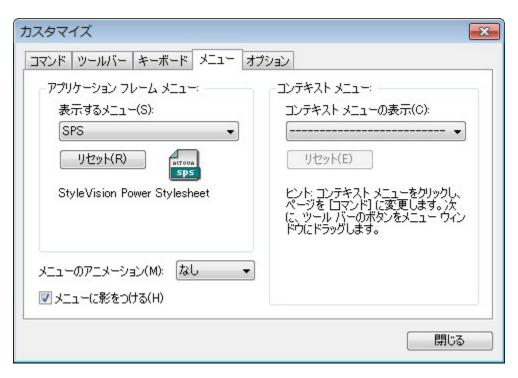
- 1. [すべてリセット] ボタンをケノックすることで、インストール時のオドジナルショートカナへ割り当てが、セナされます。すべてのキーボード割り当てを本当にリセットしても良しかの確認を行うダイアログが表示されます。
- 2. [は、]をクリックすることで、すべてのショートカナ割り当てがリセナされます。

<u>設定するアクセラレーター</u>

現在機能は割り当てられていません。

メニュータブ

[メニュー] タブではメイン メニューや(右クトックにより表示される) コンテキスト メニューをカスタマイズすることができます。 メニュー バーゴ はドキュメト が開かれている時に表示されるデフォルト メニューと SPS ドキュメト が開かれている時に表示される SPS メニューがあります。



メニューをカスタマイズする

- 1. カスタマイズするメニューバーを選択します(スクリーンショナではSPS メニューが選択されています)。
- 2. [コマンド] タブをクトックして、目的のコマンドをメニューバーへドラックしてくたさい。

メニューからコマンドを削除する

- 1. アプケーションフレームメニューペインから、(ドキュメントが開かれていない状態で使用可能なメニューを表示する) デンォルト おこしばドキュメントが開かれている状態で使用可能なメニューを表示する) SPS を選択します。
- 2. カスタマイズダイアログが開かれている状態で(i) アプリケーションのメニューバーから削除するメニュー、ませま(ii) これらのメニューの 一つから削除するコマンドを選択します。
- 3. おけば) メニューからメニュー・バー、おけはメニューコマノドをドラッグ、おけは(ii)メニューを右クトック、おけば 削除」を選択します。

メニューバーをリセッナする

- 1. アプリケーションメニューペインのエンボボックス内でリセットするメニューエトリを選択します。
- メニュー名の下にある[リセット] ボタンをクナックします。メニューバーをノセナしてもより確認するプロンプトカ表示されます。

コンテキストメニューをカスタマイズする(右クリックメニュー)

- 1. コンテキスト メニューのコンボボックスから目的のコンテキスト メニューを選択します。
- 2. 「コマンド」タブをクトックして、表示されたコンテキスト メニューヘコマンドをドラックします。

コンテキストメニューからコマンドを削除する

- 1. コマンドまたはコマンドを表すアイコンを右クリックします。
- 2. ポップアップ メニューから[削除] オプションを選択します。 まけは、コンテキストメニューからコマンドをドラッグして、マウスポインターの下にチェック マークが表示されたらすぐドロップします。

コンテキスト メニューをリセットする

- 1. コンテキスト メニューのコンボボックスから目的のコンテキスト メニューを選択します。
- 2. コンテキスト メニュー名の下にある[リセット] ポタンをクリックします。 リセットしてもより確認するプロンプトカ装示されます。

コンテキスト メニュー ウィンド ウを閉じる

タイトルバーの右側にある閉じるアイコンをクリックするか、

• カスタマイズダイアログボックスの[閉じる] ボタンをクリックしてくたさい。

<u>メニューのアニメーション</u>

メニューのアニメーション オプションにより、メニューがウトックされたときの表示方法を指定することができます。メニューのアニメーションにあるドロップがウン・リストから、目的のオプションを選択してください。

メニューニ影をつける

メニューの周りに影をつける場合にこのオプションを選択してください。すべてのメニューに影か加えられます。

オプションタブ

[オプション] タブでは、ツールドー」あるその他の機能をカスタマイズすることができます。

[ツール・一つ画面のどんを表示] オプシュンを有効にすることで、ツール・ーアイテムのどんが表示されます。画面のどん オプションコよショートカルも画面に表示するかを指定するかめのサブオプションが与えられています。

- 関連項目
- <u>ツールド</u>
- ビュー ツールド
- ユーザーインターフェイス

11.11.4 ツールバーとウィンドウの復元

このコマンドにより、ツール・・、ウインドウ、入力ヘル・・、そのたGUI コンポーネントが、デフォルトの状態に戻されます。変更を反映するには、StyleVision を再起動する必要があります。

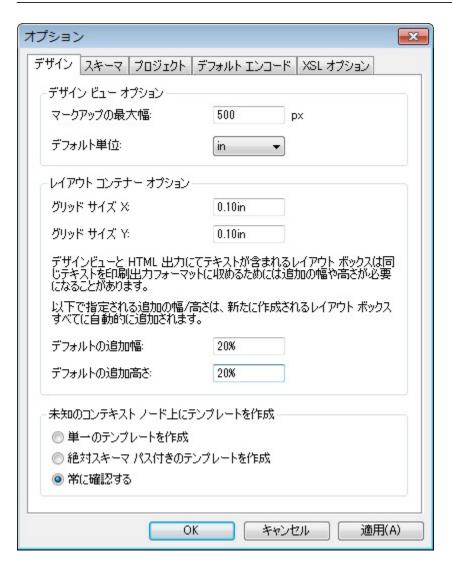
- □ 関連項目
- ・ツールド
- ビュー ツールド
- ユーザーインターフェイス

11.11.5 オプション

[オプション] コマンドにより、HTML 出力 ファイルのエンコードを指定するオンタのダイアログが表示されます。

デザインビューオプション

デザイン タブでは、アプリケーション全体で使用される一般的なデザイン オプションをセナト することができます (以下のスクリーンショナを参照)。



以下のオプションをセナすることができます:

- マーケアップの(ピクセル) 最大幅。ピクセル数として正の整数を入力してくたさい。
- レイアナ コンテナーのグリンド サイズ。指定された長さが、対応するグリンドの軸に表示されるグリンド間の幅となります。
- レイアナ ボックスのデフォルト追加幅と追加高さ。入力された長さが出力フォーマナトにてテキストのレンダルノグを行うために必要な追加 長さ出てレイアナ ボックスに追加されます。これらの値はペーセンテージとして、または絶対長さ単位として入力することができます。
- ユンテキストノードが未知の場所にノード テンプレートが作成された時のデフォルトの振る舞い。このオプションは、デザインのスキーマソース内にあるエンテキスト内部に配置することができないアイテムに対してテンプレートが作成されるようなユーザー定義テンプレートに適用されます。このようなユーザー定義テンプレート内部でノードが作成されると(i) その名前だけで、まけは(ii) スキーマルトからのフル なとともにノードを作成することができます。デフォルトの振る舞いとして、これらオプションンのどれかを選択することができるまか、このような状況でユーザーへの確認を求めるように指定することができます。デフォルトの選択は「常に確認する」です。

スキーマオプション

スキーマッソーでは、要素や属性をアルファベナ順に表示することができます。スキーマタブに対応するチェックボックスにチェックを入れてくたさい(以下のスクノージンコナを参照)。デフォルトでは、属性がアルファベナ順で、要素がスキーマの構造に従うかたちで表示されるようになっています。

デフォルトのエンコード

出力 HTML ファイルのデフォルトのエンコードを設定するコよ、エンコードオプションのリストから必要なオプションを選択し、[OK] をクトックしてください。 以降新規作成されるすべてのHTML 出力のエンコード にてこのダブで設定された設定が使用されるようごなります。

XSLT-for-HTML では、以下の位置に出力エンコーディング情報が登録されます:

• スタイルシートのxsl:output 要素にある encoding 属性:

<xsl:output version="1.0" encoding="UTF-8" indent="no" omit-xml-declaration="no"
media-type="text/html" />

• HTML 〜ッダーのcontent-type meta 要素にあるcharset 属性:

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

メモ これらの設定は新規 SPS にて使用されるデフォルトのエンコーディングで、このダイアログから現在開かれている SPS のエンコーディングを変更することはできません。現在開かれている SPS のエンコーディングを変更するコよファイル プロ ディコマイドを使用してください。

MobileTogether デザイン

SPS ファイルを Mobile Together デザイン(MTD) ファイルに変換するために下にレストされたオプションを使用することができます。(変換は Mobile Together デザインファイルとしてエクスポート するコマンドを使用して行われます)。

- (例えば太字まけは斜線の要素を含む) やグラン要素) インラインフォーマルを含む構文テキストと やグラフノベルのテキストはデフォルトで、テキストがブロックレベルのテキストのフォーマルを持つ箇所で単一のラベルに変換されます。各ラベルが元のテキストフラグメトに対応するフォーマルを持つ箇所で、この種類のテキストを複数のMTD ラベルに分割するオプシュンを設定することができます。
- SPS にブランチを持つ条件が存在する場合、MTD への変換はデフォルトで条件の全てのブランチを変換し、MTD ドキュメント内の他のシーケンスの1つ下に置かれます。条件(と全てのブランチ)の変換を抑制することを任意で選択することができます。

11.12 ウィンドウ メニュー

[ウィンドウ] メニューコは、StyleVision ウイ・ドウがGUI に表示される方法を指定するかのコマイが収められています([重ねて表示]、[上下に並べて表示]、[左右に並べて表示])。ウイドウを最大化するコは、そのウイドウの最大化ポタンをクリックしてください。

[ウィンドウ] メニューコは Style Vision ウイドウがGUI に表示される方法を指定するかのコマドが収められています([重ねて表示]、[上下に並べて表示]、[左右に並べて表示])。 ウイドウを最大化するコよ そのウイドウの最大化ポタンをクトックしてください。

更に、現在開かれているすべてのドキュメント名が表示されます。他のウインドウをアクティブにするコよ、対応するドキュメント名をクトックしてくたさい。

ウルドウダイアログ

ドキュメトのノスト下部コま「ウイドウ」ダイアログヘアクセスするかのコマナが用意されています。このエトリーをクックすることで「ウインドウ」ダイアログが開かれ、現在開かれているウインドウのノストと選択されたウインドウス対して適用することのできるコマナが表示されます(名前をクックすることで、そのウインドウが選択されます)。

警告: [ウィンドウ] ダイアログから抜けるには、[OK] をクリックして、[ウィンドウを閉じる] ボタンをクリックしないようにしてください。[ウィンドウを閉じる] ボタンをクリックすると、ウィンドウ ダイアログにて現在選択されているウィンドウが閉じられます。

- □ 関連項目
- ユーザーインターフェイス

11.13 ヘルプメニュー

StyleVision[ヘルプ] メニューゴまStyleVision のヘルプマニュアルヘアクセスする」がのロマイドやStyleVision に関する情報ヘアクセスする」がのロマイド、そして Altova Web サイト上のサポートページへのレクが収められています。ヘルプメニューゴま登録ダイアログも含まれており、購入した製品のライセンスキーコードを入力することもできます。

「ヘリプ」メニューのコマンドについては、以下のサブセクションで記述されます:

- 目次、インデックス、検索
- ライセス登録、注文フォーム、レジストレーション、最新情報のチェック
- 他のコマド

11.13.1 目次、インデックス、検索

▼ 目次

□ 説明

ヘノレプウンドウの左側のペイノに目次を表示した、StyleVision の画面上のヘノレプマニュアルを開きます。目次はヘノレプドキュメト全体の概要を表示しています。目次のエトリをクレックしてトピックに移動することができます。

▼ インデックス

□ 説明

へいプナイドウの左側のペイノニキーワード インデックスを表示したStyleVision の画面上のヘッレプマニュアルを開きます。目次 はヘルプ・キュメント 全体の概要を表示しています。 インデックスはキーワードをリストし、キーワードをダブルクテックすることでトピックへ移動することができます。 キーワードが つ以上のトピックコンクされている場合は、トピックのリストが表示されます。

▼ 検索

■ 説明

へいプウイ・プロケー側のペイノコ検索ダイアログを表示したStyleVision の画面上のヘソレプマニュアルを開きます。単語を検索するコよ、入力フィールドコ検索対象を入力して、(i)「Return」を押す、まけはii)「トピックのリスト」をクトックします。を押します。ヘルプンステムは、ヘルパキュメント全体で全文検索を行いセナーしたリストを返します。アイテムを表示するかはフィテムをダブルクトックします。

11.13.2 ライセンス登録、注文フォーム、レジストレーション、最新情報のチェック

▼ ソナウェアのライセンス認証

■ 説明

Altova 製品ソストウェアをダウンロードすると、無料評価キーおけよ購入されたライセンスキーを使用して、製品にライセンスを供与、おけよ、ライセンスの認証を行うことができます。

無料 評価ライセンス初めて製品のダウムロードとインストールを行うと、ソフトウェアライセンス認証ダイアログが表示さ

れます。ダイアログでは無料 評価 ライセンスをリクエストすることができます。ユーザーの名前、所属会社名、そして電子メールアドレスを表示されるダイアログに入力し リクエストをクトックします。ライセンスファイルが入力された電子メールアドレスに送信されます。この手順 は数分を要します。ライセンスファイルを適切な場所に保存します。リクエストをクリックすると、リクエストダイアログの下に入力フィールドが表示されます。このフィールドはライセンスファイルの 父を取ります。ライセンスファイルを参照 まけよライセンスファイルへの 父を入力し「OK」をクトックします。(「ソフトウェアのライセンス認証ダイアログ」内で「新規のライセンスをアップロードする」をクトックしてライセンスファイルへの 父を入力するダイアログニアクセスすることができます。)ソフトウェアは30日の間アンロックされます。

- 永続的なライセンスキー:ソフトウェアライセンス認証ダイアログコは永続的なライセンスキーを購入するがかのだりが、含まれて、ます。このがみをクトックすると、製品の永続的なライセンスキーを購入することのできる Altova オンラインショップに移動することができます。受信する電子メールはライセンスデータを含むライセンスファイルの書式で送信されてします。3つの種類の永続的なライセンスが存在します。インストール済み、同時使用ユーザー、名前を持つユーザー・インストール済みのライセンス は単一のエンピューター上のノフトウェアのエックを解除します。インストール済みのライセンス は単一のエンピューター上のノフトウェアのエックを解除します。インストール済みのライセンスを N 台のエンピューターのナッカに購入すると N 台までのエンピューター上でノフトウェアを使用することができます。 同時使用ユーザーライセンス は N 人の同時使用ユーザーに N 人のユーザーに同時にノフトウェアを使用することを許可します。(10N 台のエンピューターにノフトウェアをインストールすることができます)。 名前を持つユーザーライセンスは特定のユーザーが合の異なるエンピューター上でノフトウェアを使用することを許可します。ソフトウェアのライセンスを認証するコは「新規のライセンスをアップロードする」をクトックして表示されるダイアログ内でライセンスファイルを照、まだはライセンスファイルの やを入力し「OK」をクトックします。
- メモ 複数のユーザーライセンスのために各ユーザーは各自の名前を入力するようにプロンプトされます。

<u>ライセンスの電子メールと Altova 製品へのライセンス供与(有効化)の異なる方法</u> Altova から受信するライセンス電子メールコよライセンスファイルが添付ファイルとして含まれています。ライセンスファイルは、altova_licenses ファイル拡張子を有しています。

Altova 製品のライセンスを認証するコよ 以下のうち1つ行ってください

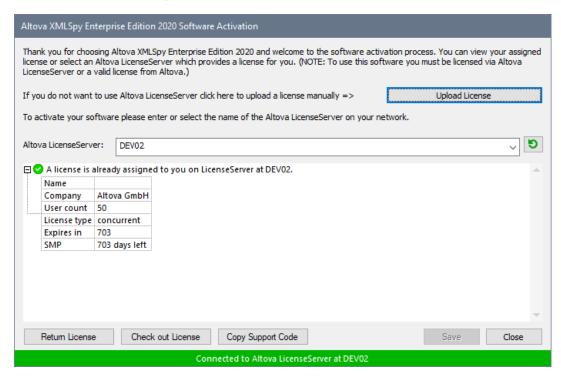
- 適切な場所にライセンスファイル(.altova_licenses)を保存し、ライセンスファイルをダブルクリックし、表示されるダイアログに必要な情報を入力し、「キーの適用」をクリックして完了します。
- ライセンスファイル(.altova_licenses) を適切な場所に保存します。Altova 製品内では、「ヘルプ | ライセンス登録メニューコマドを選択し、新規のライセンスをアップロードします。ライセンスファイルへの、父を入力し「OK」をクトックします。
- 適切な場所にライセスファイル(.altova_licenses)を保存し、Altova LicenseServer のライセスプールにアップロードします。以下を行うことができます:(i) 製品のノフトウェアライセス認証ダイアログから Altova 製品からライセスを取得します。(以下を参照)まけは(ii) Altova LicenseServer から製品へのライセスを割り当てます。LicenseServer の使用の詳細に関しては、下記のトピックを参照してくたさい。

ソストウェアライセンス認証ダイアログ (*下のスクリーンショッ*ナ) は「ヘルプ | ソフトウェア アクティベーション」をクトックすることに
より常にアクセスすることができます。

以下の方法によりノフトウェアをアクティブ化することができます:

- ソフトウェアライセンス認証ダイアログでライセンスを登録する方法。ダイアログ内で、「新規のライセンスをアップロード」をクトックして、ライセンスファイルを参照し選択します。「OK」をクトックしてライセンスファイルへの。父を確認しまず、複数のユーザーライセンスの場合は個人の名前です)。「保存」をクトックして完了します。
- ネットワーク上のAltova LicenseServer を使用してライセンス供与する方法:ネトワーク上のAltova LicenseServer を使用してライセンスを取得するコはノフトウェアのライセンスの認証ダイアログの下にあるAltova LicenseServer を使用するをクリックします。使用するLicenseServer がインストールされているマンンを選択します。License Servers の自動検知はLAN 上で配信が送信されることを意味します。これらの配信がサブネッ

トに制限されているため、License Server は自動検知のためのグライア・イマンンと同じサブネナーに存在する必要があります。自動検知が作動しない場合、サードーの名前を入力します。Altova License Server はライセンスプール内で Altova 製品のためのライセンスを有している必要があります。License Server プール内に存在する場合、ソフトウェアライセンス認証ダイアログロ表示されます(Altova XMLSpy 内のダイアログで表示されている例を参照してくたさい)。「保存」をグリッグしてライセンスを取得します。



マンと固有のライセスがLicenseServer からインストールされると、7日間は、LicenseServer に戻すことができません。7日過ぎると(「ライセンスを戻す」をクトグして)マンンのライセスをLicenseServer に戻すことができ、このライセンスは、他のグライア・トによりLicenseServer から取得することができます。LicenseServer 管理者は、LicenseServer のWeb UIを使用して、取得されたライセンスの割り当てを解除することができます。ライセンスの返却は、マンと固有のライセンスのみに適用され、現在使用中のライセンスコは適用されないことに注意してくたさい。

ライセンスのチェックアウト

ライセスが製品でシン上に保管されるように、ライセスをライセスプールから30 日間チェックアナトすることができます。これにより、オフラインで作業することが可能しています。この機能はどても役に立ちます。Altova License Server にアクセスできない環境(例えば、旅行中にAltova 製品がインストールされたラップトップエンピューターで作業する場合などが挙げられます。ライセンスはチェックアナトされて、ますが、License Server は、ライセンスが使用中と表示し、ライセンスは他のマンンで使用することができません。ライセンスはチェックアナーの期間が終わると自動的にチェックインされた状態を戻します。ませは、チェックアナトされたライセンスはノナ・ウェアのライセンスの認証ダイアログのが安とを使用して「チェックイン」することができます。

ライセンスをチェックアナするコお以下をおごれ、ます: (i) ソストウェアのライセンスの認証ダイアログで「ライセンスのチェックアナ」をクトックします(上のスクノーンショナ参照)。(ii) ライセンスのチェックアナダイアログ内から、チェックアナの期間を選択し、「チェックアナ」をクトックします。ライセンスがチェックアナされます。ライセンスのチェックアナトを2つの状態が発生します: (i) ソストウェアのライセンス認証ダイアログは時刻およびチェックアナの期限を含む・チェックアナトに関する情報を表示します。 (ii) ダイアログ内の「ライセンスのチェックアナト」 ボダンは「チェックイン」 ボダンに変更されます。「チェックイン」 ボダンをクトックして、ライセンスをチェックインすることができます。チェックアナ 期間の期限が切れると、ライセンスは自動的にチェックイン状態に戻されるため、選択したチェックアナの期間がサフラインで作業する期間をカバーするように確認してくたさい。

メモーライセンスのチェックアナを可能にするコは、LicenseServer上でチェックアナ機能が有効化されている必要ががます。チェックアナを試みる際この機能が有効化されていない場合、エラーメッセージが表示されます。この場合、LicenseServer管理者に連絡してください。

サポートコードのコピー

「サポートコードのコピー」をクトックして、ライセンスの詳細をクトップボードにコピーしてください。これは<u>オンラインサポートフォームを</u>使用してサポートをリクエストする際に必要なデータです。

Altova LicenseServer を使用することより、IT 管理者は、リアルタイムでネトワーク上の全てのライセンスの概要、および、クライアトの割り当てと、クライアトのライセンスの使用状況を確認することができます。LicenseServer を使用する利点は、ですから、多数のAltova ライセンスを管理することのできる管理機能です。Altova LicenseServer は、Altova Webサルトで無料で提供されています。Altova LicenseServer およびAltova LicenseServer を使用したライセンスの供与に関する詳細は、Altova LicenseServerドキュメトを参照してください。

▼ 注文フォーム

■ 説明

ソフトウェア製品のライセンス許与バージョンを注文する準備が整っている場合、(前のセクション参照) ソフトウェアライセンス認証ダイアログ内の「永久ライセンスの購入」ボダン、まけは「注文フォーム」 コマイドを使用して Altova オンラインショップに移動して注文することができます。

▼ 登録

■ 説明

Altova 製品登録ページをブラウザーのタブに表示します。Altova ソフトウェアを登録することにより、最新の製品の情報が得られます。

▼ 更新のチェック

■ 説明

Altova サーバーは接続して、お新しいドージョンの製品が利用可能かどうかチェックし、その結果を表示します。

11.13.3 他のコマンド

▼ ポーセター

■ 説明

インターネット上にある Altova サポートセンターへのレクとなっています。サポートセンターコま FAQ やディスカッションフォーラム か含まれており、問題の解決方法を探り、Altova の技術サポートスタッフへアクセスすることができます(現在英語のみの提供となります)。

▼ WEB 上のFAQ

■ 説明

インターネット上にある Altova の FAQ へのレクとなっています。 FAQ データベースは Altova のサポートスタッフにより常時更

新されています。

▼ コポーネトのダウロード

□ 説明

イクターネ外上にある Altova のエンポーネトダウルロードセクターへのレクとなっています。このレク先から様々なエンポーネントソフトウェアをダウルロードして、Altova 製品とされて使用することができます。ソフトウェアエンポーネントは XSLT や XSL FOプロセッサーからアプリケーションサービスプラナフォームまで、幅広く提供されています。 エンポーネントダウルロードセクターにてご利用しているノフトウェアは、通常無料でご利用していれます。

▼ インターネ小上のStyleVision

□ 説明

インターネット上にある<u>Altova Web サイト</u>へのレクとなっています。<u>Altova Web サイト</u>では、StyleVision や関連するテクノロジーについて確認することができます。

▼ StyleVision について

□ 説明

スプラッシュ画面と製品の、デンシン番号が表示されます。StyleVision の64 ビッナ・デションを使用している場合、これはアプリケーション名の後のサフィックス(x64) により示されています。32 ビッナ・デションコ まサフィックス は存在しません。

付録 491

付録 12

この付録には(i) StyleVision にて使用されるXSLT エンジンの情報、(ii) データベース データ型から XML スキーマ データ型への変換 情報、(iii) StyleVision の技術情報、(iv) StyleVision のライセンス情報について記述されます。各情報は以下のサブセケンコンに含め られます:

XSLT エンジンの情報

StyleVision にて出力を生成するために使用される Altova XSLT エンジンの実装に関する情報が記されます。

- Altova XSLT 1.0 エンジン
- Altova XSLT 2.0 エンジン
- Altova XSLT 3.0 エンジン
- XSLTとXPath/XQuery 関数

技術情報

StyleVision 口関する技術的な情報が記述されます。

- OS とメモルの要件
- Altova XMLパーサー
- Altova XSLTとXQuery エンジン
- Unicode のサポート
- インターネトの接続

<u>ライセンス情報</u> StyleVision の配布とライセンスに関する情報が含まれます。

- 電子的なソナウェアの配布
- ライセスメータング
- 著作権
- 使用許諾契約書

12.1 XSLT および XQuery エンジンに関する情報

StyleVision のXSLT およびXQuery エンジンは、W3C 仕様に従っています、ですから、XMLSpy の以前の デジュン内のAltova エンジンよ 上厳密です。この結果、以前のエンジンで無視されていた小さなエラーが、StyleVision によりエラーとして挙げられます。

例えば

- パス演算子の結果がメードと非ーノードを両方含む場合、型エラー(err:XPTY0018)です。
- パス式 E1/E2 内のE1 が一下のシーケンスを評価しない場合、型エラー(err:XPTY0019) です。

この種類のエラーが発生した場合、XSLT/XQuery ドキュメトまけはインスタンスドキュメントを必要に応じて修正してください。

このセクションは、エンジンの実装固有の機能を仕様別に整理して説明します。

- XSLT 1.0
- XSLT 2.0
- XSLT 3.0
- XQuery 1.0
- XQuery 3.1

12.1.1 XSLT 1.0

StyleVision のXSLT 1.0 エンジンは W orld W ide W eb Consortium (ワール・ワイ・ウェブ・コンノーシアム) (W 3C) の1999 年 11 月 16 日版のXSLT 1.0 勧告 および 1999 年 11 月 16 日版のXPath 1.0 勧告 に準拠します。実装に関しての以下の情報に注意してください。

実装についての注意点

xsl:output のmethod 属性がHTMLI設定された場合、または、がHTML 出力 デフォルトで選択されている場合、XML または XSLT ファイル内の特殊文字はHTMLドキュメトにHTML 文字参照として出力内に挿入されます。例えば、文字 U+00A0 (ブレーケ無しのスペースのための16 進数レフィレンス) がHTML コード内に文字の参照(or)、または、エンティティ参照 &mbsp; として挿入されます

12.1.2 XSLT 2.0

<u>このセクション</u>.

- エンジン 適合性
- 下位互換性
- スキーマ認識
- 実装固有の振る舞い

適合性

StyleVision のXSLT 2.0 エンジイよ W orld W ide W eb Consortium (ワール・ワイド・ウェブ・コンノーシアム) (W 3C) の2007年1月23日版のXSLT 2.0 勧告 および2010年12月14日版のXPath 2.0 勧告に準拠します。

下位互換性

XSLT 2.0 エンジン は下位互換性を有します。XSLT 2.0 エンジンの下位互換性が有効によるのは、XSLT 1.0 スタイルシートを処理するが以て SLT 2.0 エンジン が使用される際です。XSLT 1.0 エンジン と下位互換性を持つ XSLT 2.0 エンジンにより作成される出力に違いがあるかもしれないで出さき意してください。

名前空間

XSLT 2.0 スタイルシートは、XSLT 2.0. プレフィックス内で使用することのできる型コレストラクタ おより関数を使用するか、以下の名前空間を宣言する必要があます。下のノストは通常使用されるノストです。希望する場合は、代替プレフィックスを使用することもできます。

名前空間	プレフィックス	名前空間 URI
XML スキーマ型	xs:	http://www.w3.org/2001/XMLSchema
XPath 2.0 関数	fn:	http://www.w3.org/2005/xpath-functions

通常これらの名前空間は、以下のノストで表示されるようにxsl:スタイルシートまけは、xsl:transform 要素で宣言されます:

```
<xsl: スタイルシー version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
```

</xsl: スタイルシート>

次の点に注意してくたさい

- XSLT 2.0 エンシノは (上のテーブルでノストされている) XPath 2.0 および XQuery 1.0 関数 名前空間 をデフォルト の関数名前空間として使用します。 XPath 2.0 および XSLT 2.0 関数をプレフィックス無してスタイルシート内で使用することができます。 XPath 2.0 関数 名前空間 をスタイルシート内でプレフィックスと共に宣言すると、割り当てられた宣言内でプレフィックスを追加して使用することができます。
- XML スキーマ名前空間から型コンストラクタと型を使用する場合、名前空間 宣言内で使用された、プレフィックスを使用して型コンストラクタを呼び出されまればかません(例えば、xs:date)。
- XPath 2.0 関数の一部は XML スキーマデータ型と同じ名前を保有します。例えば、XPath 関数 fn:string および fn:boolean のためづよ 同じロケーション名: xs:string および xs:boolean を持つ XML スキーマデータ型 が存在します。ですから、XPath 式 string('Hello')を使用する場合、式は xs:string('Hello')ではなく fn:string('Hello') として検証します。

スキーマ認識

XSLT 2.0 エンジノは、スキーマを認識します。ですから、ユーザー定義 スキーマ型 および xsl:validate 命令を使用することができます。

実装固有の振る舞い

以下は、XSLT 2.0 エンジンが、特定のXSLT 2.0 関数の振る舞いの実装-特定のアスペケトをどのようは扱うかの説明です。

xsl:result-document

追加してサポーされるエンコードは以下の通りです(Altova-固有): x-base16tobinary およびx-base64tobinary.

function-available

インスコープ関数の使用をテストする関数 (XSLT、XPath、および拡張関数)。

unparsed-text

href 属性は 以下を受け入れます(i) ベース-uri フォルダー内のファイルの相対, ス および(ii) 相対, スを持つまけは持たない file://プロトコール。追加してサポートされる エンコードは以下の通りです(Altova-固有): x-binarytobase16 および x-binarytobase64.

unparsed-text-available

href 属性は 以下を受け入れます(i) ベース-uri フォルダー内のファイルの絶対, ス および(ii) 絶対, なき持つませまけさい file://プロトコール。 追加してサポートされる エンコードは以下の通りです (Altova-固有): x-binarytobase16 および x-binarytobase64.

メモ RaptorXML の先行製品であるAltova XML で実装されていり以下のエンコード値は使用しないでけるい base16tobinary, base64tobinary, binarytobase16 and binarytobase64.

12.1.3 XSLT 3.0

StyleVision のXSLT 3.0 エンジンは W orld W ide W eb Consortium (ワール・ワイ・ウェブ・コンノーシアム) (W 3C) の2017 年 6 月 8 日版 XSLT 3.0 勧告候補 および2017 年 3 月 21 日版 XPath 3.1 勧告候補 に準拠します。

XSLT 3.0 エンジノは、XSLT 2.0 エンジンと同様の実装固有の機能を搭載しています。 更に 以下のXSLT 3.0 機能へのサポートを 含みます。 更に次の一連の新規のXSLT 3.0 機能をサポートします: XPath/XQuery 3.1 関数とオペレーターと XPath 3.1 仕様。

メモ 任意のストリーミングの機能は現在サポートされていません。streamable 属性の値に関係なく、ドキュメート全体がデールコードされ、使用することのできるメモルが十分な場合は処理されます。64-ビットアプリでは、これは問題していません。this メモル問題の場合は、システムに解決策を追加する必要があります。

12.1.4 XQuery 1.0

<u>このセクション</u>.

- エノジン 適合性
- スキーマ認識
- **T**
- 名前空間
- XMI、ノースと検証
- 静的および動的な型のチェック
- ライブラノモジュール
- 外部関数

- 照合順序
- 数値データの精度
- XQuery 命令サポト

適合性

StyleVision のXQuery 1.0 エンジンは W orld W ide W eb Consortium (ワール・ワイド・ウェブ・コンノーシアム) (W 3C) の 2010 年 12 月 14 日版の XQuery 1.0 勧告に準拠します。 XQuery 標準は、多数の機能の実装についての裁量を提供します。 下口は XQuery 1.0 エンジンがどのようについる機能を実装するがについて説明するリストが下に挙げられています。

スキーマ認識

XQuery 1.0 エンジンはスキーマを認識します。

エンコード

UTF-8 およびUTF-16 文字のエンコードは サポートされています。

名前空間

以下の名前空間 URI と関連する インドは定義済みです。

名前空間	プレフィックス	名前空間 URI
XML スキーマ型	xs:	http://www.w3.org/2001/XMLSchema
スキーマインスタンス	xsi:	http://www.w3.org/2001/XMLSchema-instance
内蔵の関数	fn:	http://www.w3.org/2005/xpath-functions
Local 関数	local:	http://www.w3.org/2005/xquery-local-functions

次の点に注意してください

- XQuery 1.0 エンジンは、上にリストされナプレフィックスを名前空間に対応する、サンドとして認識します。
- Since the 上にリストされた内蔵の関数 名前空間は、XQuery内のデフォルの関数です。内蔵の関数 が呼び出される際、名前空間、fn: プレフィックスを使用する必要はありません。(例えば、string("Hello") がfn: string 関数を呼び出す場合。)しかし、プレフィックス fn: は クロリプロローグ内で名前空間を宣言することが内蔵の関数を呼び出す時に使用することができます。(サンプル fn: string("Hello")).
- クエノプロローグ内でdefault function 名前空間 式を宣言することには、デフォルの関数 名前空間をすることに変更することができます。
- XML スキーマ名前空間空の型を使用する場合、プレフャクス×s: は、名前空間を明確に宣言することが、また、これらのプレフャクスをクロノプローグ内で、イイドすることが使用することができます。(サンプル×s:date および×s:year Month Duration。) XML スキーマ名前空間のために、他のプレフャクスを使用する場合は、クロノプローグ内で明確に宣言されている必要があます。(サンプル declare 名前空間 alt =
 - "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04").)
- untypedAtomic, dayTimeDuration, おい yearMonthDuration データ型 が23 January 2007 のCR 共 に、XPath データ型 名前空間からXML スキーマ名前空間へ移動されていることで注意してください。ですから以下といます: xs: year Month Duration。

関数のナダの名前空間、型コンストラクタ、ノードテスト、か間違って割り当てられている場合、エラーが発生します。しかし、一部の関数はスキーマデータ型と同じ名前を持つことご注意してくたさい。例:fn: string および fn: bool ean。(xs: string および xs: boolean (は宣言されています) 名前空間 プレフィックス 関数ませま型コンストラクタが使用されるが決定します。

XML ソースドキュメントと検証

XQuery 1.0 エンジンを使用して、実行されるXQuery ドキュメ・ト内のXMLドキュメ・トは、整形式である必要があります。 しかし、 XML スキーマン従い有効である必要はありません。 XML ファイルが外部スキーマと関連付けられ、また有効な場合、ポストスキーマ検証 情報が XML データのために生成され、クエト検証のために使用されます。

静的および動的な型のチェック

静的分析フェーズは、外部レファレンスの存在(例、モジュールのため)、呼ひ出された関数と変数が定義済みであるか、など構文などのケエ リのアスペクトをチェックします。静的分析フェーズでエラーが検知されると、実行は停止されます。

ケエル実際に作動中にランタイム中に動的な型チェックは実行されます。型がオペレーションの必要条件と整合性を持たない場合、エラーが 報告されます。例えば、式 xs:string("1") + 1 は エラーを返します。型 xs:string のオペライを足し算のオペレーションが実 行できないとかです。

ライブラノモジュール

ライブラノモジュー川よ、再利用のナダ関数と変数を保管します。 XQuery 1.0 エンジノは単一の外部 XQuery ファイルに保管されて いるモジュールをサポートします。このようなモジュールファイルはプロローグにターゲット名前空間に関連するモジュール宣言を含む必要があり ます。 以下はモジュールサンプルです:

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns:webaddress() { "http://www.altova.com" };
```

すべての関数おより変数は、モジュールに関連した名前空間に属するモジュール内で宣言されています。モジュールはクエノプロローグ内の import module ステートメートを使用して X Query ファイルコークパーナー する際に使用されます。 import module ステートメート は、ライブラノモジューハファイル内で直接宣言された、関数と変数のみをインポートします。例:

```
import module namespace modlib = "urn:module-library" at "modulefilename.xq";
i f
          ($modlib:company = "Altova")
then
          modlib:webaddress()
          error("No match found.")
else
```

外部関数

外部関数は、サポーされていません。例えば、external キーワードを仕様する式など。以下参照

declare function hoo(\$param as xs:integer) as xs:string external;

照合順序

デフォルの照合順序は、Unicode コードポイトをベースユナン文字列を比較するUnicode-コードポイト照合順序です。その他にサ ポーされる照合順序は、ICU 照合順序はここにリストされるとおりです。特定の照合順序を使用する場合、サポーされる照合順序の リスト 内に与えられているとおりURI を提供します。fn:max とfn:min 関数を含む文字列の比較は、指定された照合順序により行わ れます。 照合順序オプションが指定されていない場合、デフォルトのUnicode-コードポイト照合順序 が使用されます。

数値データの精度

- xs:integer データ型 は任意の精度があり、表記できる桁数に制限はありません。
- xs:decimal データ型(コン)数点の後に20 桁の制限が決ます。
- xs:float とxs:double データ型には15 桁の精度の制限があます。

XQuery 命令サポート

Pragma 命令は、サポーされていません。発生した場合、無視されフォールジクの式が検証されます。

12.1.5 XQuery 3.1

StyleVision のXQuery 3.1 エンシノは W orld W ide W eb Consortium(ワー) は・ワイド・ウェブ・コンノーシアム) (W 3C) の 2017 年 3 月 21 日版のXQuery 3.1 候補勧告に準拠し、またXPath およびXQuery 関数 3.1.へのサポートを含みます。 XQuery 3.1 仕様は、3.0 仕様のスーペーセナです。XQuery 3.1 エンジノは、ですかXQuery 3.0 機能をサポートします。

実装固有の特性はXQuery 1.0 同様です。

XSLT と XPath/XQuery 関数 12.2

このセクションでは、XPath およびぎは、XQuery 式で使用することのできる、Altova 拡張関数と他の拡張関数をリストします。 Altova 拡張関数はAltova のXSLT およびXQueryエンシンで使用することができ、W 3C 標準で定義された。関数ライブラケで使用することので きる機能に追加して機能を提供します。

一般的な情報

以下の一般的な情報に注意してください

- W 3C 仕様により定義されているコア関数ライブラルの関数は、関数の呼び出しにプレフィックスは必要ありません。これは、XSLT およびXQuery エンジンが、XPath/XQuery 関数仕様で指定されているhttp://www.w3.org/2005/xpathfunctionsプレフィックス無しの関数をデフォルト関数の名前空間に属するものとして読み込むためです。
- 関数において、各アイテムが引数となるようなシーケンスが期待されており、2つ以上のアイテムがシーケンスによし呼び出された場 合、エラーが返されます。
- 全ての比較はUnicode コードポイトコレクションを使用することで行われます。
- QName の結果は [prefix:]localname とう形式でシリアル化されます。

xs:decimal の精度

精度とよ、数値内にある桁数のことで、仕様では少なくとも18桁が求められます。xs:decimal 型に結果が収められる除算の場合、端数 処理を行うことなく精度は小数点以下の19桁になります。

黙示的なタイムゾーン

2つのdate、time、まざはdateTime 値を比較する場合、比較する値のタイムゾーンを明らかにする必要があります。値の中にタイムゾ 一ンが明示的に与えられていない場合、黙示的なタイムゾーンが使用されます。黙示的なタイムゾーンはシステムクロックから取得され implicit-timezone() 関数によりその値をチェックすることができます。

照合順序

デフォルりの照合順序は、Unicode コードポイントをベースに文字列を比較するUnicode コードポイント照合順序です。エンジンは Unicode 照合アルゴリズムを使用しています。他のサポートされる照合順字は下にリストされるICU 照合順序です。 使用するコよ サポ ートされる照合順序のJストのURI を提供してくたさい(アのテーブル)。max とmin 関数を含む、文字列の比較は 指定された照合順 序に沿って行われます。照合順序オプシンが指定されていない場合、デフォルトのUnicode コードポイント照合順序が使用されます。

言語	URI
da: デンマーケ語	da_DK
de : ドイツ語	de_AT, de_BE, de_CH, de_DE, de_LI, de_LU
en : 英語	en_AS, en_AU, en_BB, en_BE, en_BM, en_BW, en_BZ, en_CA, en_GB, en_GU, en_HK, en_IE, en_IN, en_JM, en_MH, en_MP, en_MT, en_MU, en_NA, en_NZ, en_PH, en_PK, en_SG, en_TT, en_UM, en_US, en_VI, en_ZA, en_ZW
es: スペル 語	es_419, es_AR, es_BO, es_CL, es_CO, es_CR, es_DO, es_EC, es_ES, es_GQ, es_GT, es_HN, es_MX, es_NI, es_PA, es_PE, es_PR, es_PY, es_SV, es_US, es_UY, es_VE
fr: 万汉語	fr_BE, fr_BF, fr_BI, fr_BJ, fr_BL, fr_CA, fr_CD, fr_CF, fr_CG, fr_CH, fr_CI, fr_CM, fr_DJ, fr_FR, fr_GA, fr_GN, fr_GP, fr_GQ, fr_KM, fr_LU, fr_MC, fr_MF, fr_MG, fr_ML, fr_MQ, fr_NE, fr_RE, fr_RW, fr_SN, fr_TD, fr_TG

it: /タ /ア語	it_CH, it_IT
ja : 日本語	ja_JP
nb: ノルケー語 (ブーケモール)	nb_NO
nl: オランダ語	nl_AW, nl_BE, nl_NL
nn: ノルウェー語 (ニーノシュク)	nn_NO
pt: ポルガ 語	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: ロシア 語	ru_MD, ru_RU, ru_UA
sv: スケーデン語	sv_FI, sv_SE

<u>名前空間軸</u>

名前空間軸はXPath 2.0 にて廃止されましたが、名前空間軸の使用はサポーされています。XPath 2.0 メカニズムにより名前空間情報へアクセスするづよ、in-scope-prefixes()、namespace-uri()、namespace-uri-for-prefix(関数を使用してくたさい。

12.2.1 Altova 拡張関数

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラケで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいなられており、altova:プレフィンクスが、このセグションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してください。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメントを参照してください。

W 3C のXPath/XQuery 関数仕様で定義された関数は、以下で使用することができます: (i) XSLT 子アンテキスト内のXPath 式と (ii) XQuery 文書内のXQuery 式。このドキュメトでは、前者(XSLT 内のXPath)のエンテキストで使用することのできる関数を、XP シンボルと共に表示し、と称します。後者(XQuery)で使用することのできる関数はのXQ シンボルと共に表示され、XQuery 関数と共に作業することができます。W 3C のXSLT 仕様は XPath/XQuery 関数の仕様ではなく、XSLT 文書内のXPath 式でも使用することのできる関数を定義します。これらの関数は XSLT シンボルと共に表示され、XSLT 関数と称されます。関数を使用することのできるXPath/XQuery およびXSLT の デジョンは、関数の詳細に記載されて、ます(アのシンボルを参照してください)。 XPath/XQuery およびXSLT 関数ライブラからの関数は、プレフィックス無しでノストされています。 Altova 拡張関数などの、他のライブラからの関数はプレフィックスと共にフストされています。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

XSLT 関数

XSLT 関数はXSLT 2.0 のcurrent-group() やkey() 関数と同様に XSLT コンテキストにて使用することができます。(例え ば、XQuery コンテキストなどの)非-XSLT コンテキストでは使用することができません。XBRL に対する XSLT 関数は、XBRL をサポ ートするエディションのAltova 製品でのみ使用することができます。

XPath/XQuery 関数

XPath/XQuery 関数は、XSLT コンテキスト、XQuery 関数のXPath 式で使用することができます:

- 日付/時刻
- 位置情報
- イメージン関連した

- 文字列

12.2.1.1 XSLT 関数

XSLT 拡張関数 はXSLT コンテキスト内のXPath 式にて使用することができます。(例えば、XQuery コンテキストなどの) 非-XSLT コンテキストでは使用することができません。

関数の名前指定と言語の適用性は関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラノで使用可能 な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xsltextensions は収められており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のバージョンが拡張機能への 継続的サポート、まだは個別の関数の振る舞いは変更する可能性があることに注意してください。 Altova 拡張機能へのサポートに関し ては、今後のリノースのドキュメントを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

標準関数

distinct-nodes [altova:]

altova:distinct-nodes(node()*) をnode()* とする XSLT1 XSLT2 XSLT3 入力として1つ以上のノードを必要とし、同じセナから重複した値を持つノードを除しナノードを返します。XPath/XQuery 関数 fn:deep-equal を使用して比較を行うことができます。

- *∃ サンプル*
 - altova:altova:distinct-nodes (country) は重複しが直を持つものを除く、全ての子 country ノード返しま す。
- evaluate [altova:]

altova:evaluate(XPathExpression as xs:string[, ValueOf\$p1, ... ValueOf\$pN]) XSLT1

XSLT2 XSLT3

XPath 式を必要と、必須引数として文字列を いします。評価された式の出力を返します。例えば altova:evaluate('//Name[1]') はドキュメト内の最初のName 要素のエンテングを返します。式 //Name[1] は、一重引用符を使用することには、文字列としていなされます。

altova: evaluate 関数は オプロとして追加の引数を持つことができます。これらの引数は、p1, p2, p3... pN の名前を持つスコープ内の変数の値です。使用に関して以下の点に注意してくたさい(i)変数は、x が整数である箇所のフォームpx の名前と共に定義される必要が対けます。(ii) altova: evaluate 関数の引数は、(*上の署名参照*) 2 番目の引数からは、数値順の変数のシーケンスつ対応した引数のシーケンス変数の値を与えます。p1 to pN: 第 2 引数は変数 p1 の値で、第 3 引数は、変数 p2 の値です。(iii)変数の値は型item*である必要が対けます。

□ サンプル

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
outputs "hi 20 10"
```

上のノストに関して、以下の点に注意してくたさい

- altova: evaluate 式の第2引数は変数 \$p1 に割り当てられた値で、第三の引数は変数 \$p2 に割り当てられた値で、第三の引数は変数 \$p2 に割り当てられた値です。
- 関数の第4番目の引数は、引用符による囲いで表示された文字列の値です。
- xs:variable 要素のselect 属性は XPath 式を提供します。この式はxs:string の型である必要が あり、一重引用符で囲まれています。

□ 変数の使用方法を更に説明するサンプル

- <xs1:variable name="xpath" select="'\$p1'" />
 <xs1:value-of select="altova:evaluate(\$xpath, '//Name[1]')" />
 Outputs "//Name[1]"

altova:evaluate() 拡張関数は、XSLT スタイルシート内のXPath 式が動的に評価される必要のあるシチュエーションで役に立ちます。例えば、ユーザーが並べ替えの必要条件をソウエストする場合、このシチュエーションは属性 UserReq/@sortkey に保管されます。スタイルシートでは、以下の式が使用できます:<xsl:sort

select="altova:evaluate(../UserReq/@sortkey)" or der="ascending"/>。altova:evaluate() 関数は、コンテキストノードの親のUserReq 子要素のsortkey 属性を読み込みます。sortkey 属性の値が Price の場合、Price は、altova:evaluate() 関数により返され、select 属性 <xsl:sort select="Price" or der="ascending"/> の値づいます。この sort 命令が、Order とう要素のコンテキスト内で発生する場合、Order 要素は Price の子の値ご従い並べ替えられます。また、@sortkey の値が、Date の場合、Order 要素は、Date の子の値に従い並べ替えられます。ですから、Order の並べ替えの条件は、ランタイムでの sortkey 属性から選択されます。これは、以下の式などでは達成することはできません: <xsl:sort select="../UserReq/@sortkey" or der="ascending"/>。上の場合、並べ替え条件は sortkey 属性自身であり、Price または、Date (または、現在の sortkey のエンテング)ではありません。

メモ 静的なコンテキストは、変数以外以外で、呼び出し環境の名前空間、型、機能、を含みます。ベースURI とデンカルの名前空間は継承されます。

■ 追加サンプル

静的な変数: <xsl:value-of select="\$i3, \$i2, \$i1" /> 3 つの変数の値を出力します。

動的な変数を持つ動的 XPath 式:

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 30)" />
"30 20 10 "を出力します。
```

動的な変数を持たないXPath 式:

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath)" />
出力エラー $p3 (対して定義されている変数はありません。
```

encode-for-rtf [altova:]

```
altova:encode-for-rtf(input as xs:string, preserveallwhitespace as xs:boolean, preservenewlines as xs:boolean) をxs:string とする XSLT2 XSLT3

RTF のよめのロードに入力文字列を変換します。空白と新しい行は、それぞれの引数にお指定されるboolean の値に基づき保管されます。
```

[トップ]

XBRL 関数

Altova XBRL 関数はXBRL をサポーナするAltova 製品のエディンョンのみで使用することができます。

xbrl-footnotes [altova:]

```
altova:xbrl-footnotes(node()) を node()* とする XSLT2 XSLT3
ノードを入力引数として必要と、入力ノードに参照されるXBRL フォノートノードを返します。
```

xbrl-labels [altova:]

```
altova:xbrl-labels(xs:QName, xs:string) を node()* とする XSLT2 XSLT3 以下の2 つの入力引数を必要とします: ノード名とノードを含むタケノノミファイルロケーション。 関数は、入力ノードと関連した XBRLラベルノードを返します。
```

[<u>トップ</u>]

12.2.1.2 XPath/XQuery 関数: 日付と時刻

Altova の日付/時刻拡張関数はXPath とXQuery 式で使用することができ、XML スキーマの異なる日付おより時刻データ型で保存されているデータを処理するための追加機能を提供します。このセクションの関数は、Altova のXPath 3.0 およびXQuery 3.0 エンジンで使用することができます。これらの関数は、XPath/XQuery コンテキストで使用することができます。

関数の名前指定と言語の適用性に関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラノで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいないされており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してくたさい。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

▼ 機能によりバループ化

- xs:dateTime に期間を追加して、xs:dateTimeを返す
- xs:date に期間を追加して、xs:date を返す
- xs:time に期間を追加して、return xs:time を返す
- フォーマナと期間の取得
- 現在の日付/時刻を生成する関数からタイムゾーンを削除する
- 期間から日付、時刻、分数、および、砂数を返す
- 日付から整数を週の曜日として返す
- 日付から週数を整数として返す
- 各型の構文コンポーネントから日付、時刻、期間の型を構築する
- 文字列入力から日付、日付時刻 または時刻 を構築する
- 年齢は関連した関数

▼ アルファベナ順にグループ化

```
altova:add-days-to-date
altova:add-days-to-dateTime
altova:add-hours-to-dateTime
altova:add-hours-to-time
altova:add-minutes-to-dateTime
altova:add-minutes-to-time
altova:add-months-to-date
altova:add-months-to-dateTime
altova:add-seconds-to-dateTime
altova:add-seconds-to-time
altova:add-years-to-date
altova:add-years-to-dateTime
altova:age
altova:age-details
altova:build-date
altova:build-duration
altova:build-time
altova:current-dateTime-no-TZ
altova:current-date-no-TZ
altova:current-time-no-TZ
altova:date-no-TZ
altova:dateTime-no-TZ
altova:days-in-month
altova:hours-from-dateTimeDuration-accumulated
altova:minutes-from-dateTimeDuration-accumulated
altova:seconds-from-dateTimeDuration-accumulated
altova:format-duration
altova:parse-date
```

altova:parse-dateTime

altova:parse-duration
altova:parse-time
altova:time-no-TZ
altova:weekday-from-date
altova:weekday-from-dateTime
altova:weeknumber-from-date
altova:weeknumber-from-dateTime

[トップ]

xs:dateTime (期間を追加する xp3.1 xQ3.1

これらの関数はxs:dateTime に期間を追加し、xs:dateTime を返します。xs:dateTime 型はCCYY-MM-DDThh:mm:ss.sss のフォーマナです。これはxs:date とxs:time フォーマナの連結で、T により区切られて、ます。タイムゾーンサフィックス+01:00 (for example)は任意です。

add-years-to-dateTime [altova:]

altova:add-years-to-dateTime(DateTime as xs:dateTime, Years を xs:integer) asxs:dateTime とする XP3.1 XO3.1

日付までの期間を年数で表示します。第2の引数は第1の引数として与えられたxs:date に追加される年数です。結果はxs:date 型です。

■ サンプル

- altova:add-years-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), 10) は2024-01-15T14:00:00 を返します。
- altova:add-years-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), -4) は2010-01-15T14:00:00 お返します。
- add-months-to-dateTime [altova:]

altova:add-months-to-dateTime(DateTime as xs:dateTime, Months を xs:integer)
asxs:dateTime とする XP3.1 XQ3.1

xs:dateTime に月数での期間を追加します(下のサンプル参照)。第2 の引数は、第1 の引数として与えられたxs:dateTime に追加される月数です。結果はxs:dateTime 型です。

□ <u>サ</u>ンプル

- altova:add-months-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), 10) 2014-11-15T14:00:00 を返します。
- altova:add-months-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), -2) 2013-11-15T14:00:00 を返します。
- add-days-to-dateTime [altova:]

altova:add-days-to-dateTime (DateTime as xs:dateTime, Days as xs:integer) をxs:dateTime

xs:dateTime に日数での期間を追加します(アのサンプル参照)。第2 の別数は、第1 の別数とて与えられた xs:dateTime に追加される日数です。結果はxs:dateTime 型です。

□ <u>サ</u>ンプル

• altova:add-days-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), 10)は2014-01-25T14:00:00 を返します。

- add-hours-to-dateTime [altova:]

altova:add-hours-to-dateTime (DateTime as xs:dateTime, Hours as xs:integer) をxs:dateTime とする XP3.1 XO3.1

xs:dateTime に時間数での期間を追加します(アのサンプル参照)。第2の引数は第1の引数として与えられたxs:dateTime に追加される時間数です。結果はxs:dateTime 型です。

□ サンプル

- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), 10) は2014-01-15T23:00:00 を返します。
- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), -8) は2014-01-15T05:00:00 を返します。
- add-minutes-to-dateTime [altova:]

altova:add-minutes-to-dateTime (DateTime as xs:dateTime, Minutes as xs:integer) をxs:dateTime とする XP3.1 XQ3.1

xs:dateTime に分数での期間を追加します(下のサンプル参照)。第2 の引数は、第1 の引数として与えられたxs:dateTime に追加される分数です。結果はxs:dateTime 型です。

■ サンプル

- altova:add-minutes-to-dateTime(xs:dateTime("2014-01-15T14:10:00"), 45) 2014-01-15T14:55:00 お风します。
- altova:add-minutes-to-dateTime(xs:dateTime("2014-01-15T14:10:00"), -5) 2014-01-15T14:05:00 を返します。
- add-seconds-to-dateTime [altova:]

altova:add-seconds-to-dateTime(DateTime as xs:dateTime, Seconds を xs:integer)
asxs:dateTime とする XP3.1 XO3.1

xs:dateTime 「沙数での期間を追加します(下のサンプル参照 。第2 の引数は 第1 の引数として与えられた xs:dateTime 「追加される秒数です。結果はxs:dateTime 型です。

■ サンプル

- altova:add-seconds-to-dateTime(xs:dateTime("2014-01-15T14:00:10"), 20) 2014-01-15T14:00:30 を返します。

[トップ]

xs:date に期間を追加する xp3.1 xQ3.1

これらの関数はxs:date に期間を追加し、xs:date を返します。xs:date 型はCCYY-MM-DD フォーマトです。

add-years-to-date [altova:]

altova:add-years-to-date(Date as xs:date, Years as xs:integer) をxs:date とする XP3.1
XQ3.1

日付までの期間を年数で表示します。第2の引数は第1の引数として与えられたxs:date に追加される年数です。結果はxs:date 型です。

■ サンプル

- altova:add-years-to-date(xs:date("2014-01-15"), 10) は2024-01-15 を返します。
- altova:add-years-to-date(xs:date("2014-01-15"), -4) は2010-01-15 を返します。

add-months-to-date [altova:]

altova:add-months-to-date(Date as xs:date, Months as xs:integer) をxs:date とする XP3.1

日付までの期間を月数で表示します。第2の引数は第1の引数として与えられたxs:date に追加される月数です。結果はxs:date 型です。

□ サンプル

- altova:add-months-to-date(xs:date("2014-01-15"), 10) 2014-11-15 を返します。
- altova:add-months-to-date(xs:date("2014-01-15"), -2) 2013-11-15 を返します。

add-days-to-date [altova:]

altova:add-days-to-date (Date as xs:date, Days as xs:integer) をxs:date とする XP3.1 XQ3.1 日付まで期間を日数で表示します。第 2 の引数は第 1 の引数とて与えられたxs:date に追加される日数です。結果はxs:date 型です。

■ サンプル

- altova:add-days-to-date(xs:date("2014-01-15"), 10)は 2014-01-25 を返します。
- altova:add-days-to-date(xs:date("2014-01-15"), -8) は2014-01-07 を返します。

「トップ 1

フォーマトと期間の取得 xp3.1 xo3.1

これらの関数は入力 xs:duration おはxs:string を解析し、それぞれan xs:string おはxs:duration を返します。

format-duration [altova:]

altova:format-duration(Duration as xs:duration, Picture as xs:string) asxs:string

第 1 の引数として提出された期間を、第 2 の引数として提出された文字列によをフォーマナーします。出力は、文字列によりフォーマナされたテキスト文字列です。

∃ サンプル

- altova:format-duration(xs:duration("P2DT2H53M11.7S"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") は"Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7" を返します。
- altova:format-duration(xs:duration("P3M2DT2H53M11.7S"), "Months:[M01] Days: [D01] Hours:[H01] Minutes:[m01]") は"Months:03 Days:02 Hours:02 Minutes:53" を返

します。

parse-duration [altova:]

altova:parse-duration(InputString as xs:string, Picture as xs:string) をxs:duration と なる XP3.1 XQ3.1

/ やーン化された文字列を最初の引数として、文字を第 2 の引数とします。入力文字列は文字をベースで解析され、xs:duration が返されます。

■ サンプル

- altova:parse-duration("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"),
 "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") は
 "P2DT2H53M11.78" を返します。
- altova:parse-duration("Months:03 Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7", "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") は "P3M2DT2H53M" を返します。

[トップ]

xs:time (期間を追加する xp3.1 xq3.1

これらの関数はxs:time に期間を追加し、xs:time を返します。xs:time 型はhh: mm ss. sss 構文フォームです。文字 Z は協定世界時(UTC)を表します。他のタイムゾーンは、UTC との差異を +hh: mm おけま -hh: mm のフォーマナで表示しています。タイムゾーンの値が無い場合は、UTC ではない未知のタイムゾーンとして見なされます。

add-hours-to-time [altova:]

altova:add-hours-to-time(Time as xs:time, Hours as xs:integer) をxs:time とする XP3.1

日付までの期間を時間数で表示します。第2の引数は第1の引数として与えられたxs:time に追加される時間数です。結果はxs:time 型です。

■ サンプル

- altova:add-hours-to-time(xs:time("11:00:00"), 10) は21:00:00 を返します。
- altova:add-hours-to-time(xs:time("11:00:00"), -7) は04:00:00 を返します。
- add-minutes-to-time [altova:]

altova:add-minutes-to-time(Time as xs:time, Minutes as xs:integer) をxs:time とする

日付までの期間を分数で表示します。第2の引数は第1の引数として与えられたxs:date に追加される分数です。結果はxs:date 型です。

■ サンプル

- altova:add-minutes-to-time(xs:time("14:10:00"), 45) 14:55:00 を返します。
- altova:add-minutes-to-time(xs:time("14:10:00"), -5) 14:05:00 を返します。
- add-seconds-to-time [altova:]

altova:add-seconds-to-time (Time as xs:time, Minutes as xs:integer) をxs:time とする
XP3.1 XQ3.1

時間までの期間を秒数で表示します。第2の引数は、第1の引数として与えられたxs:time に追加される秒数です。結果は xs:time 型です。第2のエンポーネトは0から59.999の範囲であることができます。

□ サンプル

- altova:add-seconds-to-time(xs:time("14:00:00"), 20) は14:00:20 を返します。
- altova:add-seconds-to-time(xs:time("14:00:00"), 20.895) は14:00:20.895 を返します。

[トップ]

日付/時刻データ型からタイムゾーンの部分を削除する xp3.1 xo3.1

これらの関数は、現在のxs:dateTime、xs:date、おはxs:time値からそれぞれタイムゾーンを削除します。xs:dateTimeと xs:dateTimeStamp の差異は、後者のタイムゾーンが必要な場合です。(前者の場合は任意です。) xs:dateTimeStamp 値の フォーマトはCCYY-MM-DDThh:mm:ss.sss±hh:mm おはCCYY-MM-DDThh:mm:ss.sssz です。、日付と時刻が xs:dateTimeStamp としてシステムクロソカルら読み込まれる場合、 current-dateTime-no-TZ() 関数がタイムゾーンを削除 するために使用されます。

current-date-no-TZ [altova:]

altova:current-date-no-TZ()をxs:date とする XP3.1 XQ3.1 この関数は5|数を必要としません。current-date()(システムケロックによる現在の時刻)のタイムゾーンの部分を削除し、 xs:date の値を返します。

■ サンプル

現在のdate が2014-01-15+01:00 の場合:

- altova:current-date-no-TZ() は2014-01-15 を返します。
- current-dateTime-no-TZ [altova:]

altova:current-dateTime-no-TZ()をxs:dateTime とする XP3.1 XQ3.1 この関数には一数を必要としません。current-dateTime()(システムケロックによる現在の時刻)のタイムゾーンの部分を削除し、 xs:dateTime の値を返します。

□ サンプル

現在のdateTime が2014-01-15T14:00:00+01:00 の場合:

- altova:current-dateTime-no-TZ() は2014-01-15T14:00:00 を返します。
- current-time-no-TZ [altova:]

altova:current-time-no-TZ()をxs:time とする XP3.1 XQ3.1 この関数は3|数を必要としません。current-time()(システムケロックによる現在の時刻)のタイムゾーンの部分を削除し、 xs:time の値を返します。

■ サンプル

現在のtime が14:00:00+01:00 の場合:

• altova:current-time-no-TZ() は14:00:00 を返します。

date-no-TZ [altova:]

altova:date-no-TZ(InputDate as xs:date) をxs:date とする XP3.1 XQ3.1 この関数はxs:date 引数を必要とし、タイムゾーンの部分を削除し、xs:date の値を返します。日付か変更されていい点に注意してください。

- ∃ サンプル
 - altova:date-no-TZ(xs:date("2014-01-15+01:00")) は2014-01-15 を返します。
- ▼ dateTime-no-TZ [altova:]

altova:dateTime-no-TZ(InputDateTime as xs:dateTime) をxs:dateTime とする XP3.1 XQ3.1 この関数はxs:dateTime 引数を必要とし、タイムゾーンの部分を削除し、xs:dateTime の値を返します。日付か変更されていない点に注意してください。

- *<u>サンプル</u>*
 - altova:dateTime-no-TZ(xs:date("2014-01-15T14:00:00+01:00")) は2014-01-15T14:00:00 を返します。
- ▼ time-no-TZ [altova:]

altova:time-no-TZ(InputTime as xs:time) をxs:time とする XP3.1 XQ3.1 この関数はxs:time 引数を必要とし、タイムゾーンの部分を削除し、xs:time 値を返します。時刻は変更されていたという 意してください。

- *サンプル*
 - altova:time-no-TZ(xs:time("14:00:00+01:00")) は14:00:00 を返します。

[<u>トップ</u>]

期間から日数、時間数、分数、および、秒数を返す xp3.1 xq3.1

これらの関数は、期間から、月内の日数、時間数、分数、秒数をそれぞれ返します。

days-in-month [altova:]

altova:days-in-month(Year as xs:integer, Month as xs:integer) asxs:integer XP3.1 XQ3.1 は指定された月内の日数 を返します。Year とMonth 引数を使用して月を指定することができます。

- ∃ サンプル
 - altova:days-in-month(2018, 10) は31 を返します。
 - altova:days-in-month(2018, 2) は28 を返します。
 - altova:days-in-month(2020, 2) は29 を返します。
- hours-from-dayTimeDuration-accumulated

altova:hours-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) asxs:integer XP3.1 XO3.1

DayAndTime 引数(which is of type xs:duration)にお提供される期間内の時間の総数を返します。Day とTime コンポーネト内の時間は、整数である結果に追加されます。60 分として新規の一時間は考えられます。 わげってがい時間の値で期間の結果をナビゲートすることができます。

■ サンプル

- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5D") は日間の時間の絵 計である120 を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H") は5日間の時間の 総計プラス2時間である122 を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H60M") は5日間の時 間の総計プラス2時間プラス60分である123 を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H119M") は5日間の 時間の総計プラス2時間とプラス119分である123を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H120M") は5日間の 時間の総計プラス2時間プラス120分である124を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("-P5DT2H") は-122 を返 します。
- minutes-from-dayTimeDuration-accumulated

altova:minutes-from-dayTimeDuration-accumulated(DayAndTime as xs:duration)

asxs:integer XP3.1 XQ3.1

DayAndTime 引数(which is of type xs:duration) にい提出される期間内の分数の総数を返します Day ATime コン ポーネントは、追加され、整数である結果に追加されます。ネガティブなマイナスの値で期間の結果をナビゲートします。

🖃 サンプル

- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT60M") は を返します。
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H") は一時間内の分 数である 60 を返します。
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT1H40M") は100 を
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("P1D") は一日の砂数の絵 数である 1440 を返します。
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("-P1DT60M") は を返しま **す。**-1500
- seconds-from-dayTimeDuration-accumulated

altova:seconds-from-dayTimeDuration-accumulated(DayAndTime as xs:duration)

asxs:integer XP3.1 XQ3.1

(xs:duration の型である)DayAndTime 引数 にい提出された期間内の秒数の総数を返します。Day とTime コンポーネ ント内の秒数は、整数である結果に追加されます。オガティブな秒の値で期間をナビゲートします。

🖃 サンプル

- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1M") は一分内の秒数 **である** 60 を返します。
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H") は一時間内の秒 数*の*総数である3600 を返します。
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("PT1H2M") は3720 を返
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("P1D") は一日の秒数の総 数である86400 を返します。
- altova:seconds-from-dayTimeDuration-accumulated(xs:duration("-P1DT1M") は-86460 を返します。

xs:dateTime またはxs:dateReturn から週数を返す xp3.1 xo3.1

これらの関数は、xs:dateTime おはxs:date から曜日を整数としてを返します。曜日は(米国式フォーマルを使用して) 1 から 7 と番号づたれています。 日曜-1 と番号付けられます。欧州のフォーマルでは月曜(-1)として番号付けられます。日曜-1 である米国フォーマルは整数 0 がフォーマルを表示するために使用できる箇所で設定することができます。

weekday-from-dateTime [altova:]

時刻付きの日付を単一の引数として、この日付の曜日を正数として返します。曜日は日曜=1 から開始して番号を付けます。(月曜 =1 とする) ヨーロップのフォーマナが必要な場合、この関数の他の署名を使用します(*下の次の署名を参照*)。

□ <u>サ</u>ンプル

• altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00")) は月曜を表示する2 を返します。

altova:weekday-from-dateTime (DateTime as xs:dateTime, Format as xs:integer)

asxs:integer 25 XP3.1 XQ3.1

時間月の日付を最初の引数として、この日付の曜日を正数として返します。曜日は月曜-1 から開始して番号を付けます。第2 の引数(整数)が0 の場合、日曜-1 から開始して、曜日は1から7 と番号を付けます。第2 の引数が0 以外の整数の場合、月曜-1 で。第2 の引数が0 以外の整数の他の署名を持つと読み込まれます (前の次の署名を参照)。

□ <u>サ</u>ンプル

- altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 1) は月曜を表示する1 を返します。
- altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 4) は月曜を表示する1 を返します。
- altova:weekday-from-dateTime(xs:dateTime("2014-02-03T09:00:00"), 0) は月曜を表示する2 を返します。
- weekday-from-date [altova:]

altova:weekday-from-date (Date as xs:date) を xs:integer とする XP3.1 XQ3.1 日付を単一の別数とて、この日付の曜日を正数として返します。曜日は日曜=1 かり開始して番号を付けます。(月曜=1 とする) ヨーロッ のフォーマトが要な場合、この関数の他の署名を使用します(下の次の署名を参照)。

■ サンプル

• altova:weekday-from-date(xs:date("2014-02-03+01:00")) は月曜を表示する2 を返します。

altova:weekday-from-date(Date as xs:date, Format as xs:integer) を xs:integer とする
XP3.1 XO3.1

日付を最初の引数とし、この日付の曜日を正数として返します。曜日は月曜=1 から開始して番号を付けます。第2(フォーマナ)引数が0,の場合、日曜=1 から開始し、曜日は1 から7で番号付けられます。第2 引数が整数で0以外の場合、月曜=1です。第2 の引数がは場合、関数はこの関数の他の署名を持つと読み込まれます(前の署名を参照)。

□ サンプル

- altova:weekday-from-date(xs:date("2014-02-03"), 1) は月曜を示す1 を返します。
- altova:weekday-from-date(xs:date("2014-02-03"), 4) は月曜を示す1 を返します。
- altova:weekday-from-date(xs:date("2014-02-03"), 0) は月曜を示す2 を返します。

[トップ]

xs:dateTime またはxs:date から週数を返す xp2 xQ1 xp3.1 xQ3.1

これらの関数は週数(整数として)をxs:dateTime からxs:date.から返します。週の番号付けは、米国、欧州、イスラムのカレンダーフォーマトで使用することができます。 週の始まが異なるか、 週数の番号付けは、カレンダーのフォーマトにより異なります。 (米国フォーマトでは、日曜、欧州フォーマトでは月曜、イスラムフォーマトでは土曜が週の開始日です)。

weeknumber-from-date [altova:]

altova:weeknumber-from-date(Date as xs:date, Calendar as xs:integer) をxs:integer とする
XP2 XQ1 XP3.1 XQ3.1

正数として提出されたDate 引数の週数を返します。第2の引数(カレンダー)は続くカレンダーシステムを指定します。 サポーされる カレンダー の値は欠のとおりです:

- 0 = US 米国のカレンダー(週の始まりは日曜日)
- 1 = I SO 標準、欧州のカレンダー(週の始まりは月曜日)
- 2 = イスラムのカレンダー(週の始まりは土曜日)

デフォルトはo です。

□ サンプル

- altova:weeknumber-from-date(xs:date("2014-03-23"), 0) は13 を返します。
- altova:weeknumber-from-date(xs:date("2014-03-23"), 1) は13 を返します。
- altova:weeknumber-from-date(xs:date("2014-03-23"), 2) は13 を返します。
- altova:weeknumber-from-date(xs:date("2014-03-23")) は13 を返します。

上のサンプル(2014-03-23) **の**date **の曜日は日曜日です。米国およびイスラムのカレンダーは欧州カレンダーの工の日付** より、過間先です。

weeknumber-from-dateTime [altova:]

正数として提出されたDateTime 引数の週数を返します。第2の引数(カレンダー)は続くカレンダーシステムを指定します。サポーされるカレンダーの値は欠のとおりです:

- 0 = US 米国のカレンダー(週の始まりは日曜日)
- 1 = I SO 標準、欧州のカレンダー(週の始まりは月曜日)
- 2 = イスラムのカレンダー(週の始まりは土曜日)

Default is o.

■ サンプル

- altova:weeknumber-from-dateTime(xs:dateTime("2014-03-23T00:00:00"), 0) は13 を返します。
- altova:weeknumber-from-dateTime(xs:dateTime("2014-03-23T00:00:00"), 1) は13 を返します。

- altova:weeknumber-from-dateTime(xs:dateTime("2014-03-23T00:00:00"), 2) は13 を返します。
- altova:weeknumber-from-dateTime(xs:dateTime("2014-03-23T00:00:00")) は13 を返します。

上のサンプル(2014-03-23T00:00:00) のdate Time の曜日は日曜日です。米国およびイスラムのカレンダー 却次州カレンダーのこの日付より 上の間先です。

[トップ]

各型の構文コンポーネントから日付、時刻、期間の型を構築する xp3.1 xo3.1

関数は、xs:date, xs:time **お**はxs:duration の構文工パーネトを入力引数と、引数を結合させて対応するデータ型を構築します。

build-date [altova:]

altova:build-date(Year as xs:integer, Month as xs:integer, Date as xs:integer) をxs:date とする XP3.1 XQ3.1

第 1、第 2、第3 の別数は、それぞれ、年、月、日を表します。xs:date 型の値を構築するが対結合されます。整数の値は、特定の日付の一部の適正な範囲内である必要が別ます。例えば第 2 の別数(月の部分)は 12 以上であってはおりません。

- *サンプル*
 - altova:build-date(2014, 2, 03) は2014-02-03 を返します。
- build-time [altova:]

altova:build-time(Hours as xs:integer, Minutes as xs:integer, Seconds as xs:integer)

*\frac{\text{\$\exitt{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\exitt{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\exitt{\$\text{\$\exitt{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\exitt{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\text{\$\e

- *サンプル*
 - altova:build-time(23, 4, 57) は23:04:57 を返します。

altova:build-time(Hours as xs:integer, Minutes as xs:integer, Seconds as xs:integer, TimeZone as xs:string) をxs:time とする XP3.1 XQ3.1

第1、第2、第3引数はそれぞれ時間数(0 から23)、分数(0 から59)、およし利数(0 から59)の値です。第四の引数は値の一部としてタイムゾーンを与えます。4 つの引数が結合され、xs:time 型の値を構築します。例えば、第2の引数(分数)は59以上であってはよりません。

- *サンプル*
 - altova:build-time(23, 4, 57, '+1') は23:04:57+01:00 を返します。
- build-duration [altova:]

altova:build-duration(Years as xs:integer, Months as xs:integer)をxs:yearMonthDuration とする XP3.1 XQ3.1

xs:yearMonthDuration型の値を構築するためコは2つの引数が必要です。最初の引数は期間の年数値の部分を与え、

514

第2の引数は、月数値の部分を与えます。第2の引数(月数)が記しませませい大きくなると、整数は12により分割されます。 商は、年数の部分を表す最初の別数に加算され、(除算の)残りはの部分は月数の部分を表すために使用されます。期間の xs:dayTimeDuration 型を作成するづよ次の署名を参照してくたさい。

🖃 サンプル

- altova:build-duration(2, 10) はP2Y10M を返します。
- altova:build-duration(14, 27) はP16Y3M を返します。
- altova:build-duration(2, 24) はP4Y を返します。

altova:build-duration(Days as xs:integer, Hours as xs:integer, Minutes as xs:integer, Seconds as xs:integer) Exs:dayTimeDuration Left XP3.1 XQ3.1

xs:dayTimeDuration型の値を構築するためコよ4つの引数が必要です。最初の引数は期間の日数値の部分で、第2、 第3、第4引数は、それぞれ期間の時間数、分数、秒数値のです。これらの3つの時間引数は、次の高い単位の値に加算さ れ、結果は期間の全体の値を計算するために使用されます。例えば、72 秒は、1M+12S(1分と12秒)に変換され、この値が 期間全体の値を計算するために使用されます。xs:yearMonthDuration 型の期間を構築するためコよ次の署名を参照し てください。

🖃 サンプル

- altova:build-duration(2, 10, 3, 56) はP2DT10H3M56S を返します。
- altova:build-duration(1, 0, 100, 0) はP1DT1H40M を返します。
- altova:build-duration(1, 0, 0, 3600) はP1DT1H を返します。

「トップ 1

文字列入力から日付、日付時刻 または時刻 を構築する xp2 x01 xp3.1 x03.1

関数は、文字列を引数として、xs:date、xs:dateTime まはxs:time データ型を構築します。文字列は データ型のエノポーネ ントを提出された。サーン引数をベースは、て分析されます。

parse-date [altova:]

altova:parse-date(Date as xs:string, DatePattern as xs:string) をxs:date とする XP2 XQ1 XP3.1 XQ3.1

Date 入力文字列をxs:date の値とて返します。第二の引数であるDatePattern は、入力文字列のペーン(コンポーネ ントのシーケンス)を指定します。DatePattern は、下にしていされるエンポーネント指定子および任意の文字であるエンポーネント セルータと共に説明されています。下のサンプルを参照してくたさい。

- 日付
- М 月
- 年

DatePattern のダーンは Date の ダーンに一致する必要ががます。出力はxs:date 型であるらめ、出力は常に YYYY-MM-DD 構文フォーマナープよります。

□ サンプル

- altova:parse-date(xs:string("06-03-2014"), "[D]-[M]-[Y]") は2014-03-06を返します。
- altova:parse-date(xs:string("06-03-2014"), "[M]-[D]-[Y]") は2014-03-06 を返します。
- altova:parse-date("06/03/2014", "[M]/[D]/[Y]") は2014-03-06 を返します。
- altova:parse-date("06 03 2014", "[M] [D] [Y]") は2014-03-06 を返します。
- altova:parse-date("6 3 2014", "[M] [D] [Y]") は2014-03-06 を返します。

parse-dateTime [altova:]

altova:parse-dateTime(DateTime as xs:string, DateTimePattern as xs:string) をxs:dateTime とする XP2 XQ1 XP3.1 XQ3.1

DateTime 入力文字列をxs:dateTime の値として返します。第二の引数であるDateTimePattern は、入力文字列のパターシ(コンポーネントのシーケンス)を指定します。DateTimePattern は、下にリストされるコンポーネント指定子おより任意の文字であるコンポーネント セッシータ と共に説明されています。 下のサンプルを参照してください。

```
D 日
M 月
Y 年
H 時間
m 分
```

DateTimePattern の &ーンは DateTime の &ーンに一致する必要があます。出力はxs:dateTime 型であるよめ、出力は常にxxxx-MM-DDTHH:mm:ss 構文フォーマケ しないます。

🖃 サンプル

- altova:parse-dateTime(xs:string("06-03-2014 13:56:24"), "[D]-[M]-[Y] [H]:[m]: [s]") は2014-03-06T13:56:24 を返します。
- altova:parse-dateTime("time=13:56:24; date=06-03-2014", "time=[H]:[m]:[s]; date=[D]-[M]-[Y]") は2014-03-06T13:56:24 を返します。

parse-time [altova:]

altova:parse-time(Time as xs:string, TimePattern as xs:string) をxs:time とする XP2 XO1 XP3.1 XO3.1

Time 入力文字列をxs:time の値とて返します。第二の引数である TimePattern は、入力文字列の やーン(コンポーネントのシーケンス) を指定します。 TimePattern は、下にリストされるコンポーネント 指定子おど、狂意の文字であるコンポーネント セノレータ と共に説明されています。 下のサンプルを参照してください。

```
H 時間
m 分
s 秒
```

TimePattern の &ーンは Time の &ーンに一致する必要がかます。出力はxs:time 型であるが、出力は常に YYYY-HH:mm:ss 構文フォーマケーないます。

□ サンプル

- altova:parse-time(xs:string("13:56:24"), "[H]:[m]:[s]") は13:56:24 を返します。
- altova:parse-time("13-56-24", "[H]-[m]") は13:56:00 を返します。
- altova:parse-time("time=13h56m24s", "time=[H]h[m]m[s]s") は13:56:24 を返します。
- altova:parse-time("time=24s56m13h", "time=[s]s[m]m[H]h") は13:56:24 を返します。

[トップ]

年齢(関連した関数 xp3.1 xq3.1

関数は計算された年齢()入力引数の日付と現在の日付の期間(ii)2 つの入力引数の日付の期間を返します。altova:age 関数は、年齢を年数で返します、altova:age-details は年齢を年数、月数、日数からなるを3つの整数のシーケンスで返します。

age [altova:]

altova:age (StartDate as xs:date) <mark>をxs:integer とする XP3.1 XQ3.1</mark> 引数として提出された開始日からシステムクロックから取得された)現在の日付 までの日数を数えて、あるオブジェクトの年齢の年数である正数を返します。入力引数が1 年より大きい場合、または一年の場合、返される値は負の数です。

∃ サンプル

If the current date is 2014-01-15:

- altova:age(xs:date("2013-01-15")) は1 を返します。
- altova:age(xs:date("2013-01-16")) は0 を返します。
- altova:age(xs:date("2015-01-15")) は-1 を返します。
- altova:age(xs:date("2015-01-14")) は0 を返します。

altova:age (StartDate as xs:date, EndDate as xs:date) をxs:integer とする xP3.1 xQ3.1 最初の別数とて提出された開始日から第2の別数の終了日までの日数を数えて、あるオブジェケトの年齢の年数である正数を返します。最初の別数が1年または第2の別数と比後の日付の場合、返される値は負の数です。

∃ サンプル

If the current date is 2014-01-15:

- altova:age(xs:date("2000-01-15"), xs:date("2010-01-15")) は10 を返します。
- altova:age(xs:date("2000-01-15"), current-date()) は 現在の日付が2014-01-15 の場合 14 を返します。
- altova:age(xs:date("2014-01-15"), xs:date("2010-01-15")) は-4 を返します。

age-details [altova:]

altova:age-details (InputDate as xs:date) を(xs:integer) * とする XP3.1 XQ3.1 S|数とシステムクロックから取得された)現在の日付として提出された日付の間の年数、月数、日数の3 つの整数を返します。返された years+months+days の合計は2 つの日付(入力の日付と現在の日付) の時間差です。入力の日付は現在の日付より、中にませる遅い値をもつことができますが、入力の日付が早いが遅いがは返される値で示されていません。戻される値は常に正の数です。

∃ サンプル

現在の日付が2014-01-15 の場合:

- altova:age-details(xs:date("2014-01-16")) は(0 0 1)を返します。
- altova:age-details(xs:date("2014-01-14")) は(0 0 1) を返します。
- altova:age-details(xs:date("2013-01-16")) は(1 0 1) を返します。
- altova:age-details(current-date()) は(0 0 0)を返します。

altova:age-details(Date-1 as xs:date, Date-2 as xs:date) <mark>を(xs:integer)* とする XP3.1</mark> XQ3.1

二つの引数間の年数、月数、日数の3つの整数を返します。 返された years+months+days の合計は、2つの入力の日付の時間差です。最初の引数として提出される二つの日付はとちらか速くても、まち遅くてもかましません。 返される値は入力の日付が現在の日付より早しかまち遅いを示しません。 戻される値は常に正の数です。

■ サンプル

- altova:age-details(xs:date("2014-01-16"), xs:date("2014-01-15")) は(0 0 1) を返します。
- altova:age-details(xs:date("2014-01-15"), xs:date("2014-01-16")) は(0 0 1)を返しま

す。

[トップ]

12.2.1.3 XPath/XQuery 関数: 位置情報

以下の位置情報 XPath/XQuery 拡張関数は、StyleVision の現在の、一ジョンによりサポートされています。また、次で使用することができます: (i) XSLT コンテキスト内のXPath 式、まどは(ii) XQuery ドキュメト内のXQuery 式。

関数の名前指定と言語の適用性は関するとも

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラナで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions に以められており、altova:プレフィケフスが、このセグションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してください。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してください。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

format-geolocation [altova:]

altova: format-geolocation (Latitude as xs:decimal, Longitude as xs:decimal, GeolocationOutputStringFormat as xs:integer) をxs:string とする XP3.1 XQ3.1 緯度と経度を最初の2つの別数と、位置情報を文字列とて出力します。第3の別数 GeolocationOutputStringFormat は、位置情報出力文字列のフォーマトです。出力文字列のフォーマトを識別するため、正数の値1から4を使用します。(下の「位置情報出力文字列フォーマト」を参照してくたさい)。緯度の値の範囲は +90から-90(北から南)です。経度の値の範囲は+180から-180(東から西)です。

メモ <u>image-exif-data</u> 関数とExif メタデータの <u>@Geolocation</u> 属性を位置情報入力文字列を提供する際に使用することができます。

■ サンプル

- altova:format-geolocation(33.33, -22.22, 4) はxs:string "33.33 -22.22" を返します。
- altova:format-geolocation(33.33, -22.22, 2) はxs:string "33.33N 22.22W" を返します。
- altova:format-geolocation(-33.33, 22.22, 2) はxs:string "33.33s 22.22E" を返します。
- altova:format-geolocation(33.33, -22.22, 1) はxs:string "33°19'48.00"S 22° 13'12.00"E" お返します。

□ 位置情報出力文字列フォーマナ:

与えられた緯度と経度は下にリストされる出力フォーマトによりフォーマトされます。希望するフォーマトは整数 ID(1 から

4) により臓別されます。 緯度の値の範囲は +90 から -90 (北から南)です。 経度の値の範囲は +180 から -180 (東か ら西)です。

1

度、分、10 進の秒、方角のサフィックス付き (N/S, E/W) D°M'S.SS"N/S D°M'S.SS"E/W

サンプル: 33°55'11.11"N 22°44'66.66"W

10 進の度、方角のサフィックス付き(N/S, E/W)

D.DDN/S D.DDE/W

サンプル: 33.33N 22.22W

度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。

 $+/-D^{\circ}M'S.SS'' +/-D^{\circ}M'S.SS''$

サンプル: 33°55'11.11" -22°44'66.66"

10 進の度、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。

+/-D.DD +/-D.DD

サンプル: 33.33 -22.22

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンはカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は 4 つの Exif タグの連結です: 単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

parse-geolocation [altova:]

altova:parse-geolocation(GeolocationInputString as xs:string) をxs:decimal+ とする XP3.1

GeolocationInputString 引数を解析して、位置情報の緯度と経度(この通りの順番)を2 つのxs:decimal アイテム のシーケンスとして返します。位置情報入力文字列が提供されることのできるフォーマナは以下のノストの通りです。

メモ image-exif-data 関数とExif メタデータの@Geolocation 属性を位置情報入力文字列を提供する際に使用するこ とができます。

■ サンプル

- altova:parse-geolocation("33.33 -22.22") は2つ0xs:decimals (33.33, 22.22) のシー ケンスを返します。
- altova:parse-geolocation("48°51'29.6""N 24°17'40.2""") は2つのxs:decimals (48.8582222222222, 24.2945) のシーケンスを返します。
- altova:parse-geolocation('48°51''29.6"N 24°17''40.2"') は2つのxs:decimals (48.8582222222222, 24.2945) のシーケンスを返します。
- altova:parse-geolocation(image-exifdata (//MyImages/Image20141130.01) / @Geolocation) は2つの xs: decimals のシーケンスを返 します。

□ 位置情報入力文字列フォーマット:

位置情報入力文字列は空白で区別された緯度と経度(この通りの順番)を含む必要がみます。緯度と経度は以下のフォー マルをとることができます。組み合わせることも可能です。緯度が1つのフォーマルで、経度が他のフォーマルをとることができま す。緯度の値の範囲は+90 から-90 (N からS)。経度の値の範囲は+180 から-180 (E からW)。

メモ 単一おむダブル・川用符が入力文字列引数を区切るために使用されていると、使用されている単一おむダブル引用が、 それぞれ、分の値と秒の値、不一致をもたらします。この様な場合、分の値と秒の値を表すための使用されている引用符は、ダ ブルコ、てエスケープされる必要がかます。このセケションのサンプルでは、入力文字列を区別するために使用されている引用符 は黄色い(") で イライトされておく、エスケープした単位インジケーターは青い("") で イライトされています。

- 度、分、10 進の秒、方角のサフィックス付き(N/S, E/W) D°M'S.SS"N/S D°M'S.SS"W/E サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M'S.SS" +/-D°M'S.SS" サンプル: 33°55'11.11" -22°44'55.25"
- 度、分、10 進の分、方角のサフィックス付き (N/S, E/W)D°M.MM'N/S D°M.MM'W/E *サンプル*: 33°55.55'N 22°44.44'W
- 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のオータのプラスサインは任意です。+/-D°M.MM' +/-D°M.MM' サンプル: +33°55.55' -22°44.44'
- 10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E *サンプル*: 33.33N 22.22W
- 10 進の度、プレフィックスサイン付き(+/-): (N/S E/W)) のなっかのプラスサインは任意です。+/-D.DD +/-D.DD *サンプル*: 33.33 -22.22

フォーマナの組み合わせのサンプル 33.33N -22°44'55.25" 33.33 22°44'55.25"W 33.33 22.45

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンオカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。

Geolocation は4つのExif タグの連結です:単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

geolocation-distance-km [altova:]

altova:geolocation-distance-km(GeolocationInputString-1 as xs:string, GeolocationInputString-2 as xs:string) をxs:decimal とする XP3.1 XQ3.1 2 つの位置情報の間の距離をキロメートルで計算します。位置情報入力文字列を提供することのできるフォーマナは下にリストされて します。緯度の値の範囲は+90 から-90(北 から南)です。経度の値の範囲は+180 から-180(東 から西)です。

メモ image-exif-data 関数とExif メタデータの@Geolocation 属性を位置情報入力文字列を提供する際に使用するこ とができます。

■ サンプル

• altova:geolocation-distance-km("33.33 -22.22", "48°51'29.6""N 24°17'40.2""") はxs:decimal 4183.08132372392 を返します。

□ 位置情報入力文字列フォーマル:

位置情報入力文字列は空白で区別された緯度と経度(この通りの順番)を含む必要があります。緯度と経度は以下のフォー マトをとることができます。組み合わせることも可能です。緯度が1つのフォーマナで、経度が他のフォーマナをとることができま す。緯度の値の範囲は+90 から-90 (N からs)。経度の値の範囲は+180 から-180 (E からw)。

メモ 単一おむがブル引用符が入力文字列引数を区切るためご使用されていると、使用されている単一おむがブル引用が、 それぞれ、分の値と秒の値、不一致をもたらします。この様な場合、分の値と秒の値を表すための使用されている引用符は、ダ ブルユ、てエスケープされる必要があります。このセクションのサンプルでは、入力文字列を区別するために使用されている引用符 は黄色い(") でいったされており、エスケープした単位インジケーターは青い("") でいくライトされています。

- 度、分、10 進の秒、方角のサフィックス付き(N/S, E/W) D°M'S.SS"N/S D°M'S.SS"W/E サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M'S.SS" +/-D°M'S.SS" サンプル: 33°55'11.11" -22°44'55.25"
- 度、分、10 進の分、方角のサフィックス付き (N/S, E/W) D°M.MM'N/S D°M.MM'W/E サンプル: 33°55.55'N 22°44.44'W
- 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M.MM' +/-D°M.MM' サンプル: +33°55.55' -22°44.44'

- 10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E *サンプル*: 33.33N 22.22W
- 10 進の度、プレフィックスサイン付き(+/-); (N/S E/W)) のためのプラスサインは任意です。+/-D.DD +/-D.DD *サンプル*: 33.33 -22.22

フォーマトの組み合わせのサンプル 33.33N -22°44'55.25' 33.33 22°44'55.25"W 33.33 22.45

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジノはカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は 4 つの Exif タグの連結です: 単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

geolocation-distance-mi [altova:]

altova:geolocation-distance-mi(GeolocationInputString-1 as xs:string, GeolocationInputString-2 as xs:string) をxs:decimal とする XP3.1 XQ3.1 2 つの位置情報の間の距離をマイルで計算します。位置情報入力文字列を提供することのできるフォーマナは下にリストされていま す。緯度の値の範囲は+90 から-90(北から南)です。経度の値の範囲は+180 から-180(東から西)です。

メモ image-exif-data 関数とExif メタデータの@Geolocation 属性を位置情報入力文字列を提供する際に使用するこ とができます。

□ サンプル

• altova:qeolocation-distance-mi("33.33 -22.22", "48°51'29.6""N 24°17'40.2""") はxs:decimal 2599.40652340653 を返します。

□ 位置情報入力文字列フォーマット:

位置情報入力文字列は空白で区別された緯度と経度(この通りの順番)を含む必要があります。緯度と経度は以下のフォー マトをとることができます。組み合わせることも可能です。緯度が1つのフォーマナで、経度が他のフォーマナをとることができま す。緯度の値の範囲は+90 から-90 (N からs)。経度の値の範囲は+180 から-180 (E から W)。

メモ 単一おむダブル・川用符が入力文字列引数を区切るために使用されていると、使用されている単一およびダブル・川用が、 それぞれ、分の値と秒の値、不一致をもたらします。この様な場合、分の値と秒の値を表すための使用されている引用符は、ダ ブルコ、てエスケープされる必要があります。このセクションのサンプルでは、入力文字列を区別するために使用されている引用符 は黄色い(") で イライトされており、エスケープした単位インジケーターは青い("") で イライトされています。

度、分、10 進の秒、方角のサフィックス付き(N/S, E/W)

D°M'S.SS"N/S D°M'S.SS"W/E サンプル: 33°55'11.11"N 22°44'55.25"W

● 度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M'S.SS" +/-D°M'S.SS" サンプル: 33°55'11.11" -22°44'55.25"

度、分、10 進の分、方角のサフィックス付き (N/S, E/W)D°M.MM'N/S D°M.MM'W/E *サンプル:* 33°55.55'N 22°44.44'W

 ・ 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のオーダのプラスサインは任意です。 +/-D°M.MM! +/-D°M.MM' *サンプル*: +33°55.55' -22°44.44'

10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E *サンプル*: 33.33N 22.22W

10 進の度、プレフィックスサイン付き(+/-); (N/S E/W)) のためのプラスサインは任意です。+/-D.DD +/-D.DD *サンプル:* 33.33 -22.22

<u>フォーマナの組み合わせのサンプル</u>

33.33N -22°44'55.25" 33.33 22°44'55.25"W 33.33 22.45

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンノコカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は4つのExif タグの連結です: 単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

geolocations-bounding-rectangle [altova:]

altova: geolocations-bounding-rectangle(Geolocations を xs:sequence として GeolocationOutputStringFormat を xs:integer として)をxs:string として XP3.1 XQ3.1 最初の引数として文字列のシーケンスを取ります。シーケンスの各文字列が位置情報です。 関数はそれぞれが、最初の引数に送信さ れたすべての位置情報を含むために最適にサイズ調整された長方形の左上と右下の位置情報の座標である2つの文字列のシーケン スを返します。位置情報入力文字列が表示される書式は以下にリストされています「位置情報入力文字列書式」を参照してく たさい。緯度の値は+90 から-90 への(N からs への) の範囲です。経度の値は+180 から-180 への(E からW への) の範 囲です。

関数の2番目の引数は出力シーケンス内の2つの位置情報文字列の書式を指定します。引数はそれぞれの値が異なる位置情報文 字列書式を識別する1から4の整数の値を取ります(下記の「位置情報出力文字列書式」を参照してください)。

メモ image-exif-data 関数とExif メタデータの属性は入力文字列を提供するために使用することができます。

■ サンプル

- altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 -0.11832"), 1) ばーケンス ("51°30'33.804"N 0°7'5.952"W", "48°12'51.67116"N 16° 22 '14.61576"E") **を返します。**
- altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 -0.11832", "42.5584577 -70.8893334"), 4) ばんかく ("51.50939 -70.8893334", "42.5584577 16.3707266") **お返します**

□ 位置情報入力文字列フォーマナ:

位置情報入力文字列は空白で区別された緯度と経度(この通りの順番)を含む必要があります。緯度と経度は以下のフォー マルをとることができます。組み合わせることも可能です。緯度が1つのフォーマナで、経度が他のフォーマナをとることができま す。緯度の値の範囲は+90 から-90 (N からs)。経度の値の範囲は+180 から-180 (E からw)。

メモ 単一およびダブル引用符が入力文字列引数を区切るために使用されていると、使用されている単一およびダブル引用が、 それぞれ、分の値と秒の値、不一致をもたらします。この様な場合、分の値と秒の値を表すための使用されている引用符は、ダ ブルコ、てエスケープされる必要があります。このセクションのサンプルでは、入力文字列を区別するために使用されている引用符 は黄色い(") で イライトされておし、エスケープした単位インジケーターは青い("") で イライトされています。

- 度、分、10 進の秒、方角のサフィックス付き(N/S, E/W)D°M'S.SS"N/S D°M'S.SS"W/E サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィックスサイン付き(+/-): (N/E) のためのプラスサインは任意です。+/-D°M'S.SS" +/-D°M'S.SS" サンプル: 33°55'11.11" -22°44'55.25"
- 度、分、10 進の分、方角のサフィックス付き (N/S, E/W) D°M.MM'N/S D°M.MM'W/E サンプル: 33°55.55'N 22°44.44'W
- 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のたまりのプラスサインは任意です。+/-D°M.MM' +/-D°M.MM' サンプル: +33°55.55' -22°44.44'
- 10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E *サンプル:* 33.33N 22.22W
- 10 進の度、プレフィックスサイン付き(+/-); (N/S E/W)) のためのプラスサインは任意です。+/-D.DD +/-D.DD サンプル: 33.33 -22.22

フォーマナの組み合わせのサンプル 33.33N -22°44'55.25" 33.33 22°44'55.25"W 33.33 22.45

□ 位置情報出力文字列フォーマット:

与えられた緯度と経度は下にリストされる出力フォーマトによりフォーマトされます。希望するフォーマトは整数 ID(1 から 4) により臓別されます。 緯度の値の範囲は +90 から -90 (北 から南) です。 経度の値の範囲は +180 から -180 (東 か ら西)です。

1

度、分、10 進の秒、方角のサフィックス付き (N/S, E/W)

D°M'S.SS"N/S D°M'S.SS"E/W

サンプル: 33°55'11.11"N 22°44'66.66"W

10 進の度、方角のサフィックス付き(N/S, E/W)

D.DDN/S D.DDE/W

サンプル: 33.33N 22.22W

度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。

 $+/-D^{\circ}M'S.SS'' +/-D^{\circ}M'S.SS''$

サンプル: 33°55'11.11" -22°44'66.66"

10 進の度、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。

+/-D.DD +/-D.DD

サンプル: 33.33 -22.22

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンイよカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は4つのExif タグの連結です:単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	Ø	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

geolocation-within-polygon [altova:]

altova:geolocation-within-polygon(Geolocation as xs:string, ((PolygonPoint as xs:string)+)) をxs:boolean とする XP3.1 XQ3.1

PolygonPoint 引数により説明されているGeolocation (最初の引数) が多角形のエリア内に存在するかを決定します。も し、PolygonPoint 引数が最初と最後のポイトが同じ場合に作成される閉じられたフィギュアを作成してい場合、フィギュアを 閉じるから、最初のポイトが明示的に最後のポイトとして追加されます。全ての引数Geolocation および PolygonPoint+) は (下バストされるフォーマナの 位置情報入力文字列に以提供されます。Geolocation 引数が 多角形エア内にある場合、関数はtrue(); を返します。その他の場合はfalse()を返します。緯度の値の範囲は+90 から - 90 (北から南) です。経度の値の範囲は+180から-180 (東から西)です。

メモ image-exif-data 関数とExif メタデータの@Geolocation 属性を位置情報入力文字列を提供する際に使用するこ とができます。

サンプル

- altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24", "58 -32")**)** はtrue() **を返します。**
- altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48 24")) は true() を返します。
- altova:geolocation-within-polygon("33 -22", ("58 -32", "-78 -55", "48° 51'29.6""N 24°17'40.2""")) はtrue() を返します。

□ 位置情報入力文字列フォーマル:

位置情報入力文字列は空白で区別された緯度と経度(この涌りの順番)を含む必要が砂ます。緯度と経度は以下のフォー マトをとることができます。組み合わせることも可能です。緯度が1つのフォーマナで、経度が他のフォーマナをとることができま す。緯度の値の範囲は+90 から-90 (N からs)。経度の値の範囲は+180 から-180 (E からW)。

メモ 単一おむダブパ 用符が入力文字列引数を区切るかかで用されていると、使用されている単一おむダブパ 用が、 それぞれ、分の値と秒の値、不一致をもたらします。この様な場合、分の値と秒の値を表すための使用されている引用符は、ダ ブルコ、てエスケープされる必要があります。このセクションのサンプルでは、入力文字列を区別するために使用されている引用符 は黄色い(") でソラかされており、エスケープした単位インジケーターは青い("") でソフテいされています。

- 度、分、10 進の秒、方角のサフィックス付き(N/S, E/W) D°M'S.SS"N/S D°M'S.SS"W/E サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフャクスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M'S.SS" +/-D°M'S.SS" サンプル: 33°55'11.11" -22°44'55.25"
- 度、分、10 進の分、方角のサフィックス付き (N/S, E/W)D°M.MM'N/S D°M.MM'W/E サンプル: 33°55.55'N 22°44.44'W
- 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。+/-D°M.MM' +/-D°M.MM' *サンプル*: +33°55.55' -22°44.44'
- 10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E サンプル: 33.33N 22.22W
- 10 進の度、プレフィックスサイン付き(+/-); (N/S E/W)) のたまかのプラスサインは任意です。+/-D.DD +/-D.DD *サンプル*: 33.33 -22.22

フォーマナの組み合わせのサンプル 33.33N -22°44'55.25" 33.33 22°44'55.25"W 33.33 22.45

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジノはカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は、4 つの Exif タグの連結です: 単位の追加された(下のテーブル参照) GPSLatitude、GPSLatitude Ref、GPSLongitude、GPSLongitude Ref。

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

geolocation-within-rectangle [altova:]

altova:geolocation-within-rectangle (Geolocation as xs:string, RectCorner-1 as xs:string, RectCorner-2 as xs:string) をxs:boolean とする XP3.1 XQ3.1 長方形の対格の角を指定する第 2 及び第 3 引数 RectCorner-1、RectCorner-2)により説明されている Geolocation (最初の別数) が長方形のエア内に存在するかを決定します。全ての別数 Geolocation、RectCorner-1 および RectCorner-2)は(アバンストされるフォーマトの 位置情報入力文字列には提供されます。Geolocation 引数が長方形エア内にある場合、関数は true (); を返します。その他の場合は false () を返します。緯度の値の範囲は +90 から -90 (北から南)です。経度の値の範囲は +180 から -180 (東から西)です。

メモ <u>image-exif-data</u> 関数とExif メタデータの<u>@Geolocation</u> 属性を位置情報入力文字列を提供する際に使用することができます。

□ サンプル

- altova:geolocation-within-rectangle("33 -22", "58 -32", "-48 24") はtrue() を返します。
- altova:geolocation-within-rectangle("33 -22", "58 -32", "48 24") はfalse() を返します。
- altova:geolocation-within-rectangle("33 -22", "58 -32", "48°51'29.6""S 24° 17'40.2""") はtrue() を返します。

□ 位置情報入力文字列フォーマット:

位置情報入力文字列は空白で区別された緯度と経度(この通りの順番)を含む必要があります。緯度と経度は以下のフォーマトをとることができます。組み合わせることも可能です。緯度が1つのフォーマトで、経度が他のフォーマトをとることができます。緯度の値の範囲は+90 から-90 (N からs)。経度の値の範囲は+180 から-180 (E からw)。

メモ 単一およびダブル引用符が入力文字列引数を区切るっぱつ使用されていると、使用されている単一およびダブル引用が、それぞれ、分の値と砂の値、不一致をもたらします。この様な場合、分の値と砂の値を表すための使用されている引用符は、ダブルコ、てエスケープされる必要があります。このセグションのサンプルでは、入力文字列を区別するかっぱまれている引用符け黄色い(") で イライトされており、エスケープした単位インジケーターは青い("") で イライトされています。

- <mark>度、分、10 進の秒、方角のサンベクス付き(N/S</mark>, E/W) D°M'S.SS"N/S D°M'S.SS"W/E <u>サンプル</u> 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のかかのプラスサインは任意です。+/-D°M'S.SS"
 +/-D°M'S.SS"

サンプル: 33°55'11.11" -22°44'55.25"

- 度、分、10 進の分、方角のサフィックス付き (N/S, E/W) D°M.MM'N/S D°M.MM'W/E *サンプル:* 33°55.55'N 22°44.44'W
- 度、分、10 進の分、プレフィックスサイン付き(+/-); (N/E) のためのプラスサインは任意です。 +/-D°M.MM' +/-D°M.MM' *サンプル*: +33°55.55' -22°44.44'
- 10 進の度、方角のサフィックス付き(N/S, E/W) D.DDN/S D.DDW/E *サンプル*: 33.33N 22.22W
- 10 進の度、プレフィックスサイン付き(+/-); (N/S E/W)) のためのプラスサインは任意です。+/-D.DD +/-D.DD *サンプル:* 33.33 -22.22

<u>フォーマナの組み合わせのサンプル</u> 33.33N -22°44'55.25" 33.33 22°44'55.25"W

33.33 22.45

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンはカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は 4 つの Exif タグの連結です: 単位の追加された(下のテーブル参照) GPSLatitude、 GPSLatitudeRef, GPSLongitude, GPSLongitudeRef.

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

[トップ]

12.2.1.4 XPath/XQuery 関数: イメージに関連

以下のイメージは関連したXPath/XQuery 拡張関数は、StyleVision の現在のドージョンによりサポートされています。また、次で使用 することができます: (i) XSLT コンテキスト内のXPath 式、まけは(ii) XQuery ドキュメント内のXQuery 式。

関数の名前指定と言語の適用性は関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラノで使用可能 な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xsltextensions に収められており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のバージョンが拡張機能への 継続的サポート、まだは個別の関数の振る舞いは変更する可能性があることに注意してください。 Altova 拡張機能へのサポートに関し ては、今後のリノースのドキュメントを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(Base64String as string) をstring? とする XP3.1

イメージファイルのBase64 エノコードを引数として、イメージのファイル拡張子を、イメージのBase64 エノコード内の記録として返します。返された値は、エノコード内で使用することのできるイメージ型情報を基づしたといってす。この情報が使用できない場合、空の文字列が返されます。この関数は、Base64 イメージをファイルとして保存し、適切なファイル拡張子を動的取得する際に役に立ちます。

□ <u>サ</u>ンプル

- altova:suggested-image-file-extension(/MyImages/MobilePhone/Image20141130.01) は 'ipg' お返します。
- altova:suggested-image-file-extension(\$XML1/Staff/Person/@photo) は''を返します。

上のサンプルでは、関数の引数として与えられたノードはBase64 エンコードイメージを含むど仮定します。最初のサンプルは jpg をファイルの型および拡張子として取得します。二番目のサンプルでは、与えられたBase64 エンコードは使用できる拡張子の情報を提供しません。

mt-transform-image [altova:]

altova:mt-transform-image (Base64Image as Base64BinaryString, Size as item()+,
Rotation as xs:integer, Quality as xs:integer) をBase64BinaryString とする XP3.1 XQ3.1
Base64-エンコード イメージを最初の引数として、変換されたBase64-エンコード イメージを返します。第 2、第 3、第 4 引数は変換された以下のイメージ・ヴェーターです: サイズ、回転、およびカオルティ。

サイズ引数には3 つのサイズ変更のオプションがあります。

(X,Y)	絶対ピクセルの値。アスペケト率は保持されません。高さと幅は自動的にイメージの長いおより短い辺に逢わされるため、高さと幅の指示は関係ありません。2 つの整数アイテムのシーケンスとして値が入力されます。かってが必要です。
X	x をピクセルの新しい長い辺として、イメージの縦横比のサイズ変更が行われます。アスペケト率は保持されます。値は整数で引用符なしで入力されます。
'X%'	元のディメンションの与えられたパーセンテージにイメージのサイズを変更します。値は文字列として引用符を付けて入力される必要があります。

- 回転は以下の値を持つことができます: 90、180、270、-90、-180、-270。これらの値は回転の度数です。正の値はイメージを時計回りに回転させます。負の値はイメージを反時計回りに回転させます。 Altova Exif 属性 OrientationDegree を使用して、イメージの現在の回転の度数(0, 90, 180, 270) をイメージのExif Orientation タグから取得することができます。ですが、OrientationDegree 属性はデータのOrientation タグから取得されるため、Exif データ内にOrientation タグか存在する場合のみ使用することができます。(下のOrientationDegree 説明を参照してください)。
- クオーティは、0 から 100 の値で、JPEG 圧縮の IJG クオーティスケールの値を参照していますが、クオーティのペーセンテージのインジケーターではおりません。サイズとクオーティのどちらかを優先すると、もう一方の優先度が下がます。フルカラーソースのイメージでは、75 が通常最高値と見なされています。もし、75 が満足の、く結果をもたらざい、場合は、値を上げてく

だい。

メモ Exif データが元のイメージに存在する場合、変換時に削除され、変換されたイメージにはExif データは存在しません。

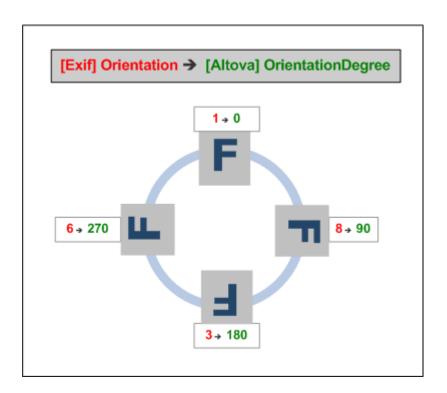
■ サンプル

- mt-transform-image (Images/Image [@id='43'], '50%', 90, 75)
 関数は、43 の@id 値を持つイメージ子孫ノード内でBase64-エンコード 文字列として保管されているイメージを入力とします。関数は変換されたイメージを返します。変換されたイメージは50%まで、サイズ変更され、時計回りに90 度回転され、75 のカオ ティンバを与えられます。
- mt-transform-image (Images/Image [@id='43'], 400, 90, 75)
 関数は前のサンプルと同じ結果を出しますが、長い辺は400 ピクセルの特定の値に設定されています。元のイメージのアスペケト率は保持されます。
- mt-transform-image (Images/Image [@id='43'], (400, 280), image-exif-data (\$XML1/\$XML1/Images/ReferenceImage) /@OrientationDegree, 75)
 このサンプルよ 前のサンプルと同じイメージを選択し、同じクオリティ値 (75) を設定します。イメージのサイズは400×280 ピクセルに設定されて、ます。回転値は、ノード内のBase64-エンコード イメージの @OrientationDegree 属性から得られます。

■ Altova Exif 属性: OrientationDegree

Altova XPath/XQuery エンジンはカスタム属性 OrientationDegree を Exif メタデータタグ Orientation から生成します。

OrientationDegree は標準 Exif タグ Orientation を正数の値 (1、8、3 ま)は 6) からそれぞれ対応する度数の値 (0,90,180,270) へ下の図に示されているように変換します。 2、4、5、7 の Orientation 値の変換 は ないと 注意してください。 (これらの向きはイメージ 1 を垂直方向中央軸で反転して、2 の値を持つイメージを取得します。そして、このイメージを 90-度ごとは計画 リンジャンプきせそれぞれ 7、4、および 5 の値を取得します。)。



三 標準 Exif メタタク

- ImageWidth
- ImageLength
- BitsPerSample
- Compression
- PhotometricInterpretation
- Orientation
- SamplesPerPixel
- PlanarConfiguration
- YCbCrSubSampling
- YCbCrPositioning
- XResolution
- YResolution
- ResolutionUnit
- StripOffsets
- RowsPerStrip
- ullet StripByteCounts
- JPEGInterchangeFormat
- JPEGInterchangeFormatLength
- TransferFunction
- WhitePoint
- PrimaryChromaticities
- YCbCrCoefficients
- ReferenceBlackWhite
- DateTime
- ImageDescription

- Make
- Model
- Software
- Artist
- Copyright

- ExifVersion
- FlashpixVersion
- ColorSpace
- ComponentsConfiguration
- CompressedBitsPerPixel
- PixelXDimension
- PixelYDimension
- MakerNote
- UserComment
- RelatedSoundFile
- DateTimeOriginal
- DateTimeDigitized
- SubSecTime
- SubSecTimeOriginal
- SubSecTimeDigitized
- ExposureTime
- FNumber
- ExposureProgram
- SpectralSensitivity
- ISOSpeedRatings
- OECF
- ShutterSpeedValue
- ApertureValue
- BrightnessValue
- ExposureBiasValue
- MaxApertureValue
- SubjectDistance
- MeteringMode
- LightSource
- Flash
- FocalLength
- SubjectArea
- FlashEnergy
- SpatialFrequencyResponse
- FocalPlaneXResolution
- FocalPlaneYResolution
- FocalPlaneResolutionUnit
- SubjectLocation
- ExposureIndex
- SensingMethod
- FileSource
- SceneType
- CFAPattern
- CustomRendered
- ExposureMode
- WhiteBalance
- DigitalZoomRatio

- FocalLengthIn35mmFilm
- SceneCaptureType
- GainControl
- Contrast
- Saturation
- Sharpness
- DeviceSettingDescription
- SubjectDistanceRange
- ImageUniqueID

- GPSVersionID
- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- GPSAltitudeRef
- GPSAltitude
- GPSTimeStamp
- GPSSatellites
- GPSStatus
- GPSMeasureMode
- GPSDOP
- GPSSpeedRef
- GPSSpeed
- GPSTrackRef
- GPSTrack
- GPSImgDirectionRef
- GPSImgDirection
- GPSMapDatum
- GPSDestLatitudeRef
- GPSDestLatitude
- GPSDestLongitudeRef
- GPSDestLongitude
- GPSDestBearingRef
- GPSDestBearing
- GPSDestDistanceRef
- GPSDestDistance
- GPSProcessingMethod
- GPSAreaInformation
- GPSDateStamp
- GPSDifferential

▼ image-exif-data [altova:]

altova:image-exif-data (Base64BinaryString as string) をelement? とする XP3.1 XQ3.1 Base64 エンコード JPEG イメージを引数として、イメージのExif メタデータを含むExif という名の要素を返します。 Exif メタデータはExif 要素の属性の値ペアとして作成されます。 属性名は、Base64エンコード内で検出された Exif データタグです。 Exif 仕様タグのフストは以下の通りです。 ベンダー特有のタグが Exif データ内に存在する場合、 タグとその値も属性値のペアとして返されます。 標準 Exif メタデータタグに追加して (下のフスト参照) Altova 特有の属性値のペアもまと生成されます。 これらの Altova Exif 属性は以下のとおりです。

■ サンプル

 属性こアクセスするコよ以下の関数を使用します: image-exif-data(//MyImages/Image20141130.01)/@GPSLatitude image-exif-data(//MyImages/Image20141130.01)/@Geolocation

- 全ての属性「アクセスする」は、以下の関数を使用します: image-exif-data(//MyImages/Image20141130.01)/@*
- 全ての属性の名前にアクセスするゴよ 以下の式を使用します:
 for \$i in image-exif-data(//MyImages/Image20141130.01)/@* return name(\$i)
 関数により返される属性の名前を検出するために役し立ちます。

■ Altova Exif 属性: 位置情報

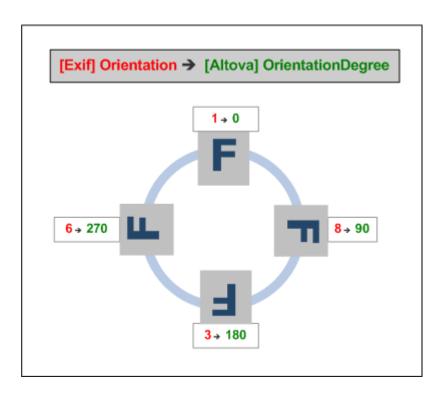
Altova XPath/XQuery エンシンオカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は、4 つの Exif タグの連結です:単位の追加された(下のテーブル参照) GPSLatitude、GPSLatitudeRef、GPSLongitude、GPSLongitudeRef。

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73		33°51'21.91"S 151° 13'11.73"E

■ Altova Exif 属性: OrientationDegree

Altova XPath/XQuery エンジノはカスタム属性 OrientationDegree を Exif メタデータタグ Orientation から生成します。

OrientationDegree は標準 Exif タグ Orientation を正数の値(1、8、3 まけは6)からそれぞれ対応する度数の値(0、90、180、270)へ下の図に示されているように変換します。2、4、5、7 の Orientation 値の変換けばいて出こ注意してください。(これらの向きはイメージ 1 を垂直方向中央軸で反転して、2 の値を持つイメージを取得します。そして、このイメージを 90-度ごとは計回りにジャンプさせそれぞれ 7、4、および 5 の値を取得します。)。



■ 標準 Exif メタデータ タグのリスト

- ImageWidth
- ImageLength
- BitsPerSample
- Compression
- PhotometricInterpretation
- Orientation
- SamplesPerPixel
- PlanarConfiguration
- YCbCrSubSampling
- YCbCrPositioning
- XResolution
- YResolution
- ResolutionUnit
- StripOffsets
- RowsPerStrip
- StripByteCounts
- JPEGInterchangeFormat
- JPEGInterchangeFormatLength
- TransferFunction
- WhitePoint
- PrimaryChromaticities
- YCbCrCoefficients
- ReferenceBlackWhite
- DateTime
- ImageDescription

- Make
- Model
- Software
- Artist
- Copyright

- ExifVersion
- FlashpixVersion
- ColorSpace
- ComponentsConfiguration
- CompressedBitsPerPixel
- PixelXDimension
- PixelYDimension
- MakerNote
- UserComment
- RelatedSoundFile
- DateTimeOriginal
- DateTimeDigitized
- SubSecTime
- SubSecTimeOriginal
- SubSecTimeDigitized
- ExposureTime
- FNumber
- ExposureProgram
- SpectralSensitivity
- ISOSpeedRatings
- OECF
- ShutterSpeedValue
- ApertureValue
- BrightnessValue
- ExposureBiasValue
- MaxApertureValue
- SubjectDistance
- MeteringMode
- LightSource
- Flash
- FocalLength
- SubjectArea
- FlashEnergy
- SpatialFrequencyResponse
- FocalPlaneXResolution
- FocalPlaneYResolution
- FocalPlaneResolutionUnit
- SubjectLocation
- ExposureIndex
- SensingMethod
- FileSource
- SceneType
- CFAPattern
- CustomRendered
- ExposureMode
- WhiteBalance
- DigitalZoomRatio

- FocalLengthIn35mmFilm
- SceneCaptureType
- GainControl
- Contrast
- Saturation
- Sharpness
- DeviceSettingDescription
- SubjectDistanceRange
- ImageUniqueID

- GPSVersionID
- GPSLatitudeRef
- GPSLatitude
- GPSLongitudeRef
- GPSLongitude
- GPSAltitudeRef
- GPSAltitude
- GPSTimeStamp
- GPSSatellites
- GPSStatus
- GPSMeasureMode
- GPSDOP
- GPSSpeedRef
- GPSSpeed
- GPSTrackRef
- GPSTrack
- GPSImgDirectionRef
- GPSImgDirection
- GPSMapDatum
- GPSDestLatitudeRef
- GPSDestLatitude
- GPSDestLongitudeRef
- GPSDestLongitude
- GPSDestBearingRef
- GPSDestBearing
- GPSDestDistanceRef
- GPSDestDistance
- GPSProcessingMethod
- GPSAreaInformation
- GPSDateStamp
- GPSDifferential

[トップ]

12.2.1.5 XPath/XQuery 関数: 数值

Altova の数値拡張関数はXPath とXQuery 内で使用することができ、データを更に処理するための追加機能を提供します。このセクションの関数はAltova のXPath 3.0 とXQuery 3.0 エンジンと使用することができます。XPath/XQuery コンテキストで使用することができます。

関数の名前指定と言語の適用性は関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライプラナで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいないされており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のドージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してくたさい。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

自動付番関数

generate-auto-number [altova:]

altova:generate-auto-number(ID as xs:string, StartsWith as xs:double, Increment as xs:double, ResetOnChange as xs:string) を xs:integer とする XP1 XP2 XQ1 XP3.1 XQ3.1 関数が呼び出される度に番号を生成します。関数が初めて呼び出される際に生成される最初の番号はStartsWith 引数により指定されます。その後の関数の呼び出しは新し、番号を生成します。この番号は前に生成された番号を、Increment 引数で指定された値ごとインクリメトします。実質的は、altova:generate-auto-number 関数は、ID 引数により作定されたカウターを作成し、このカウターが関数が呼び出されるごとにつりメトされます。ResetOnChange 引数の値が、前の関数呼び出しいる変更されると、生成される番号の値がStartsWith 値にしたよれます。自動付番は altova:reset-auto-number 関数を使用してルナトすることができます。

∃ サンプル

- altova:generate-auto-number ("ChapterNumber", 1, 1, "SomeString") は 関数が呼び出される度に番号を1つ返します。1 から始まり、関数が呼び出される度に1 ずつイングリメートします。その後の呼び出しの4番目の'SomeString"引数がである限り、イングリメートは継続されます。4番目の引数の値が変更されると、(ChapterNumber と呼ばれる) カンターは1 コルナされます。ChapterNumber の値もaltova:reset-auto-number 関数の呼び出してよりのaltova:reset-auto-number ("ChapterNumber") ようコルナされます。
- reset-auto-number [altova:]

altova:reset-auto-number(ID as xs:string) XP1 XP2 XQ1 XP3.1 XQ3.1 この関数は、ID 引数内で名づられた自動付番カウターの番号をレセトします。 引数内のカウター名を作成した関数の引数により作文を対します。 altova:generate-auto-number

- *サンプル*
 - altova:reset-auto-number ("ChapterNumber") は <u>altova:generate-auto-number</u> 関数により 作成されてhapterNumber とう名の自動付番カンターをリセントします。
 - ChapterNumber を作成したaltova:generate-auto-number 関数のStartsWith 引数の値に数値をリセナします。

[[トップ]]

数值関数

hex-string-to-integer [altova:]

altova:hex-string-to-integer(HexString as xs:string) をxs:integer とする XP3.1 XQ3.1 10 進のシステム(10進)内の正数の16 進の同値の文字列引数を必要とし、10 進の整数を返します。

■ サンプル

- altova: hex-string-to-integer('1') は1 を返します。
- altova:hex-string-to-integer('9') は9 を返します。
- altova:hex-string-to-integer('A') は10 を返します。
- altova:hex-string-to-integer('B') は11 を返します。
- altova: hex-string-to-integer('F') は15 を返します。
- altova:hex-string-to-integer('G') はつを返します。
- altova: hex-string-to-integer ('10') は16 を返します。
- altova:hex-string-to-integer('01') は1 を返します。
- altova:hex-string-to-integer('20') は32 を返します。
- altova:hex-string-to-integer('21') は33 を返します。
- altova:hex-string-to-integer('5A') は90 を返します。
- altova:hex-string-to-integer('USA') はつを返します。

integer-to-hex-string [altova:]

altova:integer-to-hex-string(Integer as xs:integer) をxs:string とする XP3.1 XQ3.1 正数を引数とて必要と、文字列とて自身のベース16 の同値を返します。

■ サンプル

- altova:integer-to-hex-string(1) は'1' を返します。
- altova:integer-to-hex-string(9) は '9' を返します。
- altova:integer-to-hex-string(10) は'A' を返します。
- altova:integer-to-hex-string(11) は'B' を返します。
- altova:integer-to-hex-string(15) は'F' を返します。
- altova:integer-to-hex-string(16) は'10' を返します。
- altova:integer-to-hex-string(32) は'20' を返します。
 altova:integer-to-hex-string(33) は'21' を返します。
- altova:integer-to-hex-string(90) は'5A' を返します。

[[トップ]]

数値フォーマト関数

[[トップ]]

12.2.1.6 XPath/XQuery 関数: スキーマ

下にリストされる Altova 拡張関数はスキーマの情報を返します。以下は関数の詳細(i) サンプル(ii) スキーマエンポーネントのリストと トスペクティブのプロノ ティです。 Altova の XPath 3.0 と XQuery 3.0 エンジンと共に使用することができ XPath/XQuery コンテキスト内で見つけることができます。

スキーマドキュメントからのスキーマ情報

関数 altova: schema には以下の2つの引数が存在します。1つのゼロ引数と他の2つの引数。ゼロ引数関数はスキーマ全体を返します。この開始点からスキーマ内をナビゲートし必要とするスキーマンポーネトをロケートします。2-引数関数は自身のQName にお、臓別される特定のコンポーネトの型を返します。 両方の場合、戻り値は関数です。 返されたコンポーネト内をナビゲートするコよ、特定のコンポーネトのプロ、ティを選択する必要があります。 プロ、ティか 非動的アイテム(すなオカコンポーネトの場合)このコンポーネトのプロ、ティをさらに選択してナビゲートすることが可能です。 選択されたプロ、ティか動的なアイテムの場合、アイテムの値が返され、ナビゲートを更に行うことはできません。

メモ Xquery 式ではスキーマは明示的に个パートされる必要があます。XPath 式では、スキーマは処理環境にインポートされる必要があます。 例えば XSLT にはxslt:import 命令を使用してインポートします。

XML ノードカらのスキーマ情報

関数 altova: type はXML ドキュメトのノードを送信しノードの型情報をPSVI から返します。

関数の名前指定と言語の適用性に関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラノで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいないされており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してください。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してください。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

▼ Schema (引数無し)

altova:schema() as (function(xs:string) as item()*)? XP3.1 XQ3.1 schema コポーネト全体を返します。schema コポーネトのプロ ティの一つを選択して更にナビゲートすることができます。

- このプロ ティがロボーネトの場合、このロンボーネトのプロ ティのこつを選択して更に深く他のステップをナビゲートすることができます。このステップは更にスキーマをナビゲートするナムが、繰り返すことができます。
- エンポーネトカ動的な値の場合、動的な値が返され更にナビゲートすることはできません。

schema コンポーネントのプロ 守ィ

- "type definitions"
- "attribute declarations"
- "element declarations"
- "attribute group definitions"
- "model group definitions"
- "notation declarations"
- "identity-constraint definitions"

(schema 以外の)他のすべてのエンポーネント型のプロッティが下にリストされています。

メモ Xquery 式ではスキーマオリテトのにインポートされる必要が決ます。XPath 式では、スキーマオ処理環境にインポートされる必要が決ます。例えば XSLT にはxslt:import 命令を使用してインポートします。

■ サンプル

- import schema "" at "C:\Test\ExpReport.xsd"; for \$typedef in altova:schema() ("type definitions") return \$typedef ("name") はスキーマ内のすべての単純型まけは複合型の名前を返します
- import schema "" at "C:\Test\ExpReport.xsd"; altova: schema() ("type definitions")[1]("name") はスキーマ内の単純型ませば複合型の最初の名 前を返します

コンポーネントとそのプロバティ

■ Assertion

プロパティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Assertion"
test	XPath プロゲルコード	

■ Attribute Declaration

プロパティ名	プログライ型	プログライ値
kind	文字列	"Attribute Declaration"
name	文字列	属性のローカル名
target namespace	文字列	属性の名前空間 URI
type definition	Simple Type おせまComplex Type	
scope	プロ 守iを持つ関数 ("class": "Scope", "variety": "global" or "local", "parent": the containing Complex Type or Attribute Group)	
value constraint	存在する場合、プレゲンを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこは"default", "value": atomic value, "lexical form": string。"value" プレゲイは namespace-sensitive 型のために使用することはできないとと注意してなさい。	
inheritable	ブール値	

■ Attribute Group Declaration

プロパティ名	プロ√~▽▽型	プロパティ値
kind	文字列	"Attribute Group Definition"
name	文字列	属性グループのローカル名
target namespace	文字列	属性グループの名前空間 URI
attribute uses	(Attribute Use) <i>の</i> シーケンス	
attribute wildcard	任意の属性のフイルドカード	

■ Attribute Use

プロパティ名	プロ/ 守ィ型	プロ/ 守ィ値
kind	文字列	"Attribute Use"
required	ブール値	属性が必要な場合はtrue、任意の場合はfalse
value constraint	Attribute Declaration を参照	
inheritable	ブール値	

■ Attribute Wildcard

プログライ名	プロ/ 守ィ型	プロパティ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 ("strict" "lax" "skip")	_

■ Complex Type

プロパティ名	プロ/ 守ィ型	プロ/ 守/値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
base type definition	Complex Type Definition	
final	文字列のシーケンス ("restriction" "extension")	
context	空のシーケンス(not implemented)	
derivation method	文字列 ("restriction" "extension")	
abstract	ブール値	
attribute uses	Attribute Use のシーケンス	
attribute wildcard	任意の属性のフイルドカード	
content type	プログラを持つ関数: ("class": "Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited	文字列のシーケンス	

substitutions	("restriction" "extension")	
assertions	アサーションのシーケンス	

■ Element Declaration

プロノティ名	プロパティ型	プロノティ値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
type definition	Simple Type おはComplex Type	
type table	プロ・ティを持つ関数 ("class": "Type Table", "alternatives": sequence of Type Alternative, "default type definition": Simple Type or Complex Type)	
scope	プロ 守心特つ関数 ("class": "Scope", "variety": ("global" "local"), "parent": optional Complex Type)	
value constraint	Attribute Declaration を参照	
nillable	ブール値	
identity-constraint definitions	Identity Constraint のシーケンス	
substitution group affiliations	Element Declaration のシーケンス	
substitution group exclusions	文字列のシーケンス ("restriction" "extension")	
disallowed substitutions	文字列のシーケンス ("restriction" "extension" "substitution")	
abstract	ブール値	

■ Element Wildcard

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 (″strict´ ″lax´' ″skip´´)	

■ Facet

プロパティ名	プロパティ型	プロパティ値
J H V 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ノロバリイ王	ノロ/ リ1 0

kind	文字列	ファセナの名前、例えば "minLength" おは"enumeration"
value	ファセナによる	ファセナの値
fixed	ブール値	
typed-value	ファセナの列挙のみ、 Array(xs:anyAtomicType	列挙値を含む配列。それぞれがトミック値のシーケンスである場合があります。(メモ列挙ファセナー関しては、実際の型にかかわらず "value" 値プロ 守石は文字列のシーケンスです)

■ Identity Constraint

プロパティ名	プロ/ 守ィ型	プロ/ 〒/値
kind	文字列	"Identity-Constraint Definition"
name	文字列	制約のローカル名
target namespace	文字列	制約の名前空間 URI
identity-constraint category	文字列 ("key" "unique" "keyRef")	
selector	XPath プロゲルコード	
fields	XPath プロ・ティレコード のシーケンス	
referenced key	(keyRef のためのみ): Identity Constraint	対応するキー制約

■ Model Group

プロノティ名	プロ/ 守ィ型	プロ/ ~ティ値
kind	文字列	"Model Group"
compositor	文字列 ("sequence" "choice" "all")	
particles	Particle のシーケンス	

■ Model Group Definition

プロノティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Model Group Definition"
name	文字列	モデルグループのローカル名
target namespace	文字列	モデルグループの名前空間 URI
model group	Model Group	

■ Notation

プログライ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Notation Declaration"
name	文字列	表記のローカル名

target namespace	文字列	表記の名前空間 URI
system identifier	anyURI	
public identifier	文字列	

■ Particle

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数まづは文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おはModelGroup	

■ Simple Type

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Simple Type Definition"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
final	文字列のシーケンス ("restriction" "extension" "list" "union")	
context	含まれるコンポーネント	
base type definition	Simple Type	
facets	Sequence of Facet	
fundamental facets	(実装されていない)空のシーケンス	
variety	文字列("atomic"["list"] "union")	
primitive type definition	Simple Type	
item type definition	(リスト型のみのための) Simple Type	
member type definitions	(ユニオン型のナダの) Simple Type のシーケンス	

■ Type Alternative

プロパティ名	プロ/ 守ィ型	プロ/ 守ィ値
kind	文字列	"Type Alternative"
test	XPath プレゲルコード	
type definition	Simple Type ませるComplex Type	

■ XPath Property Record

プロパティ名	プロ/ 守/型	プロパティ値
J 1/1/11	プロ/ リイ王	

namespace bindings	プロ 守rを持つ関数のシーケンス("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の新的べースURI
expression	文字列	文字列とてのXPath 式

▼ Schema (2つの引数)

altova:schema(ComponentKind as xs:string, Name as xs:QName) as (function(xs:string) as item()*)? XP3.1 XQ3.1

2番目の引数内で与えられている名前と同じ名前を持つ最初の引数内で指定されているコンポーネントの型を返します。コンポーネント のプロノティの一つを選択して更にナビゲートすることができます。

- このプロ、ティがエンポーネントの場合、このエンポーネントのプロ、ティの「つを選択して更に深く他のステップをナビゲートするこ とかできます。このステップは更にスキーマをナビゲートするために繰り返すことができます。
- コンポーネントが動的な値の場合、動的な値が返され更にナビゲートすることはできません。

メモ Xquery 式ではスキーマは明示的に个パトされる必要があます。XPath 式では スキーマは処理環境に个パトされる 必要があます。例えば XSLT はxslt:import 命令を使用してんポートします。

∃ サンプル

• import schema "" at "C:\Test\ExpReport.xsd"; altova:schema ("element declaration", xs:QName("OrgChart"))("type definition") ("content type") ("particles") [3]!.("term") ("kind") は番目のparticles ユポーネトの kind プロ テクを返します。 この particles ユポーネトは OrgChart の Qname を持つ要素宣言の子孫です。

```
• import schema "" at "C:\Test\ExpReport.xsd";
 let $typedef := altova:schema("type definition", xs:QName("emailType"))
 for $facet in $typedef ("facets")
 return [$facet ("kind"), $facet("value")]
 は各 emailType ユポーネトの各 facet にそのファセトの型と値を含む列挙を返します。
```

コンポーネントとそのプロノ ディ

Assertion

プログライ名	プロ/ 守ィ型	プロ/ 守ィ値
kind	文字列	"Assertion"
test	XPath プレゲルコード	

Attribute Declaration

プロノティ名	プログライ型	プロノティ値
kind	文字列	"Attribute Declaration"
name	文字列	属性のローカル名

target namespace	文字列	属性の名前空間 URI
type definition	Simple Type おはComplex Type	
scope	プロ 守存特つ関数 ("class": "Scope", "variety": "global" or "local", "parent": the containing Complex Type or Attribute Group)	
value constraint	存在する場合、プレゲンを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこは"default", "value": atomic value, "lexical form": string。"value" プレゲンは namespace-sensitive 型のためで用することはできないという注意してなさい。	
inheritable	ブール値	

■ Attribute Group Declaration

プロパティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Attribute Group Definition"
name	文字列	属性グループのローカル名
target namespace	文字列	属性グループの名前空間 URI
attribute uses	(Attribute Use) <i>の</i> シーケンス	
attribute wildcard	任意の属性のフイルドカード	

■ Attribute Use

プロノティ名	プロ/ 守ィ型	プロ・守ィ値
kind	文字列	"Attribute Use"
required	ブール値	属性が必要な場合はtrue、任意の場合はfalse
value constraint	Attribute Declaration を参照	
inheritable	ブール値	

■ Attribute Wildcard

プロパティ名	プロ/ 守ィ型	プロ/ ~~値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 (″strict′ ́ ″lax″ ″skip″)	

■ Complex Type

プロノティ名	プロ/ 守/型	プロ/ 守/値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
base type definition	Complex Type Definition	
final	文字列のシーケンス ("restriction" "extension")	
context	空のシーケンス(not implemented)	
derivation method	文字列 ("restriction" "extension")	
abstract	ブール値	
attribute uses	Attribute Use のシーケンス	
attribute wildcard	任意の属性のフイルドカード	
content type	プロ 安存特型数: ("class": "Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	文字列のシーケンス ("restriction" "extension")	
assertions	アサーションのシーケンス	

■ Element Declaration

プロパティ名	プロ/ 守ィ型	プロ/ ~~~値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
type definition	Simple Type おはComplex Type	
type table	プロ 守冷持つ関数 ("class": "Type Table", "alternatives": sequence of Type Alternative, "default type definition": Simple Type or Complex Type)	
scope	プロパライを持つ関数 ("class": "Scope", "variety": ("global" "local"), "parent": optional Complex Type)	
value constraint	Attribute Declaration を参照	
nillable	ブール値	
identity-constraint	Identity Constraint のシーケンス	

definitions		
substitution group affiliations	Element Declaration のシーケンス	
substitution group exclusions	文字列のシーケンス ("restriction" "extension")	
disallowed substitutions	文字列のシーケンス ("restriction" "extension" "substitution")	
abstract	ブール値	

■ Element Wildcard

プロパティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 (″strict″ ″lax″ ″skip″)	

■ Facet

プロパティ名	プロ/ 守ィ型	プロパティ値
kind	文字列	ファセナの名前、例えば "minLength" おは"enumeration"
value	ファセナによる	ファセナの値
fixed	ブール値	
typed-value	ファセナの列挙のみ Array(xs:anyAtomicType	列挙値を含む配列。それぞれがトミック値のシーケンスである場合があります。(メモ列挙ファセナに関しては、実際の型にかかわらず "value" 値プロ 守ィは文字列のシーケンスです)

■ Identity Constraint

プロパティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Identity-Constraint Definition"
name	文字列	制約のローカル名
target namespace	文字列	制約の名前空間 URI
identity-constraint category	文字列("key" "unique" "keyRef")	
selector	XPath プロゲルコード	
fields	XPath プロ テルコード のシーケンス	

referenced key	(keyRef のためのみ): Identity Constraint	対応するキー制約
----------------	-------------------------------------	----------

■ Model Group

プロノティ名	プログライ型	プロ/ ~
kind	文字列	"Model Group"
compositor	文字列 ("sequence" "choice" "all")	
particles	Particle のシーケンス	

■ Model Group Definition

プロノティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Model Group Definition"
name	文字列	モデルグループのローカル名
target namespace	文字列	モデルグループの名前空間 URI
model group	Model Group	

■ Notation

プロパティ名	プロ/ 守/型	プロノティ値
kind	文字列	"Notation Declaration"
name	文字列	表記のローカル名
target namespace	文字列	表記の名前空間 URI
system identifier	anyURI	
public identifier	文字列	

■ Particle

プログライ名	プログライ型	プログライ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数まりは文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おせよModelGroup	

■ Simple Type

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Simple Type Definition"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
final	文字列のシーケンス	

	("restriction" "extension" "list" "union")	
context	含まれるユンポーネント	
base type definition	Simple Type	
facets	Sequence of Facet	
fundamental facets	(実装されていない)空のシーケンス	
variety	文字列("atomic" "list" "union")	
primitive type definition	Simple Type	
item type definition	(リスト型 <i>のみのための</i>) Simple Type	
member type definitions	(ユニオン型のナダの) Simple Type のシーケンス	

■ Type Alternative

プロパティ名	プロ/ 守ィ型	プロ/ 守ィ値
kind	文字列	"Type Alternative"
test	XPath プロゲルコード	
type definition	Simple Type ませよComplex Type	

XPath Property Record

プロパティ名	プロ/ 守ィ型	プロ/ ~ティ値
namespace bindings	プロ 守存持つ関数のシーケンス("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の静的べースURI
expression	文字列	文字列としてのXPath 式

▼ 型

altova:type(Node as item?) as (function(xs:string) as item()*)? XP3.1 XQ3.1 関数 altova: type はXMLドキュメントの要素ませは属性ノードを送信しノードの型情報をPSVI から返します。

メモ スキーマか参照可能」なるかかにXMLドキュメトはスキーマ宣言を持つ必要がかます。

■ サンプル

```
• for $element in //Email
 let $type := altova:type($element)
 return $type
  はノードの型情報を含む関数を返します。
• for $element in //Email
 let $type := altova:type($element)
 return $type ("kind")
```

はノードの型のコンポーネト(単純型まけは複合型)を取りコンポーネトのkind property の値を返します。

<u>コンポーネントとそのプロバティ</u>

Assertion

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Assertion"
test	XPath プロゲルコード	

■ Attribute Declaration

プロパティ名	プロ/ 守ィ型	プログライ値
kind	文字列	"Attribute Declaration"
name	文字列	属性のローカル名
target namespace	文字列	属性の名前空間 URI
type definition	Simple Type おはComplex Type	
scope	プロ 守存特つ関数 ("class": "Scope", "variety": "global" or "local", "parent": the containing Complex Type or Attribute Group)	
value constraint	存在する場合、プレゲンを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこは"default", "value": atomic value, "lexical form": string。"value" プレゲイは namespace-sensitive 型のために使用することはできないととう注意してくたさい。	
inheritable	ブール値	

■ Attribute Group Declaration

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Attribute Group Definition"
name	文字列	属性グループのローカル名
target namespace	文字列	属性グループの名前空間 URI
attribute uses	(Attribute Use) <i>の</i> ンーケンス	
attribute wildcard	任意の属性のフイルドカード	

■ Attribute Use

プロパティ名	プロ/ 守/型	プロノティ値
kind	文字列	"Attribute Use"
required		属性が必要な場合はtrue、任意の場合はfalse

value constraint	Attribute Declaration を参照	
inheritable	ブール値	

■ Attribute Wildcard

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 (″strict″]″lax″]″skip″)	

■ Complex Type

プロパティ名	プロパティ型	プロノティ値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
base type definition	Complex Type Definition	
final	文字列のシーケンス ("restriction" "extension")	
context	空のシーケンス(not implemented)	
derivation method	文字列 ("restriction" "extension")	
abstract	ブール値	
attribute uses	Attribute Use のシーケンス	
attribute wildcard	任意の属性のフイルドカード	
content type	プロ 守冷特つ関数: ("class": "Content Type", "variety":string ("element- only" "empty" "mixed" "simple"), particle: optional Particle, "open content": function with properties ("class":"Open Content", "mode": string ("interleave" "suffix"), "wildcard": Wildcard), "simple type definition": Simple Type)	
prohibited substitutions	文字列のシーケンス ("restriction" "extension")	
assertions	アサーションのシーケンス	

■ Element Declaration

プロパティ名	プロパティ型	プロ/ 守/値
kind	文字列	"Complex Type"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
type definition	Simple Type おはComplex Type	
type table	プロ 守冷持つ関数 ("class": "Type Table", "alternatives": sequence of Type Alternative, "default type definition": Simple Type or Complex Type)	
scope	プロ 守心特つ関数 ("class":"Scope", "variety": ("global" "local"), "parent": optional Complex Type)	
value constraint	Attribute Declaration を参照	
nillable	ブール値	
identity-constraint definitions	Identity Constraint のシーケンス	
substitution group affiliations	Element Declaration のシーケンス	
substitution group exclusions	文字列のシーケンス ("restriction" "extension")	
disallowed substitutions	文字列のシーケンス ("restriction" "extension" "substitution")	
abstract	ブール値	

■ Element Wildcard

プロパティ名	プログライ型	プロ/ 守/値
kind	文字列	"Wildcard"
namespace constraint	プロ 守た持つ関数 ("class": "Namespace Constraint", "variety": "any" "enumeration" "not", "namespaces": sequence of xs:anyURI, "disallowed names": list containing QNames and/or the strings "defined" and "definedSiblings"	
process contents	文字列 ("strict" "lax" "skip")	

■ Facet

プロパティ名	プログライ型	プロケティ値
kind	文字列	ファセナの名前、例えば "minLength" おは"enumeration"
value	ファセナによる	ファセナの値
fixed	ブール値	

	Array(xs:anyAtomicType	列挙値を含む配列。それぞれがトミック値 のシーケンスである場合があります。(メモ 列挙ファセットに関しては、実際の型にかか わらず "value" 値プロ ティは文字列のシ ーケンスです)
П		///(9/

■ Identity Constraint

プロパティ名	プロ/ 守ィ型	プロパティ値
kind	文字列	"Identity-Constraint Definition"
name	文字列	制約のローカル名
target namespace	文字列	制約の名前空間 URI
identity-constraint category	文字列("key" "unique" "keyRef")	
selector	XPath プロゲイレコード	
fields	XPath プロ・ティレコード のシーケンス	
referenced key	(keyRef のためのみ): Identity Constraint	対応するキー制約

■ Model Group

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Model Group"
compositor	文字列 ("sequence" "choice" "all")	
particles	Particle のシーケンス	

■ Model Group Definition

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Model Group Definition"
name	文字列	モデルグループのローカル名
target namespace	文字列	モデルグループの名前空間 URI
model group	Model Group	

■ Notation

プロパティ名	プロケティ型	プロノティ値
kind	文字列	"Notation Declaration"
name	文字列	表記のローカル名
target namespace	文字列	表記の名前空間 URI
system identifier	anyURI	
public identifier	文字列	

■ Particle

プロパティ名	プロ/ 守ィ型	プロノティ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数または文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おせよModelGroup	

■ Simple Type

プロパティ名	プロ/ 守ィ型	プロ/ 守/値
kind	文字列	"Simple Type Definition"
name	文字列	(匿名の場合は空)型のローカル名
target namespace	文字列	(匿名の場合は空)型の名前空間 URI
final	文字列のシーケンス ("restriction" "extension" "list" "union")	
context	含まれるユンポーネント	
base type definition	Simple Type	
facets	Sequence of Facet	
fundamental facets	(実装されていない)空のシーケンス	
variety	文字列("atomic" "list" "union")	
primitive type definition	Simple Type	
item type definition	(リスト型 <i>のみのための</i>) Simple Type	
member type definitions	(ユニオン型のナダの) Simple Type のシーケンス	

■ Type Alternative

プロパティ名	プロ/ 守ィ型	プロ/ 守ィ値
kind	文字列	"Type Alternative"
test	XPath プロゲルコード	
type definition	Simple Type おせなComplex Type	

■ XPath Property Record

プロパティ名	プロ/ 守ィ型	プロ/ ~~値
namespace bindings	プロ 守存特つ関数のシーケンス("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の静的べースURI
expression	文字列	文字列としてのXPath 式

12.2.1.7 XPath/XQuery 関数: シーケンス

Altova のシーケンス拡張関数はXPath とXQuery 式で使用することができ、XML スキーマの異なる日付および時刻データ型で保存されているデータを処理することができます。このセクションの関数は、Altova のXPath 3.0 およびXQuery 3.0 エンジンで使用することができます。これらの関数は、XPath/XQuery コンテキストで使用することができます。

関数の名前指定と言語の適用性は関するとも

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライプラナで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいながったがり、altova:プレフィックスが、このセクションでは使用されます。製品の今後のドージョンが拡張機能への継続的サポート、ませは固別の関数の振る舞いは変更する可能性があることに注意してくたさい。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

attributes [altova:]

• altova:attributes("MyAttribute") はMyAttribute()* を返します。

altova:attributes(AttributeName as xs:string, SearchOptions as xs:string)
asattribute()* XP3.1 XQ3.1

入力引数 AttributeName 内で与えられた名前と同じローカル名を持つすべての属性を返します。検索は大文字と小文字を区別し、attribute:: 軸づして行われます。コンテキストノードが親要素ノードである必要が別ます。第2引数はオプションのフラッグを含みます。使用することのできるフラッグは以下の通りです:

- ▼ = 正規表現検索を切り替えます; AttributeName は正規表現検索文字列である必要があります;
- **f** = このオプランが指定されている場合、AttributeName は完全一致を提供します。それ以外の場合は、AttributeName は属性名に部分的に一致するとその属性を返します。例えば **f** が指定されている 場合、MyAtt は MyAttribute を返します。
- i = 大文字と小文字を一致させる検索に切り替えます。
- **p = 検索に名前空間プレフィックスを含みます。**AttributeName は 名前空間プレフィックスを含みます。例えば altova:MyAttribute。

フラッグは順序に関わりなく書き込むことができます。無効なフラッグはエラーを生成します。1つませば複数のフラッグを省略することができます。空の文字列は許可されていますが、引数を1つしか特かは、関数と同じ効果を持ちまず、前の署名)、ですが、空のシーケンスは第2の引数としては許可されていません。

- *サンプル*
 - altova:attributes("MyAttribute", "rfip") はMyAttribute() * を返します。

- altova:attributes("MyAttribute", "pri") はMyAttribute() * を返します。
- altova:attributes("MyAtt", "rip") はMyAttribute()* を返します。
- altova:attributes("MyAttributes", "rfip") は不一致を返します。
- altova:attributes("MyAttribute", "") はMyAttribute() * を返します。
- altova:attributes("MyAttribute", "Rip") 認識されないフラッグエラーを返します。
- altova:attributes ("MyAttribute",) 見つからは第2引数を返します。

elements [altova:]

altova:elements(ElementName as xs:string) <mark>をelement()* とする xP3.1 xQ3.1</mark> 入力引数 ElementName で与えられた名前と同じローカル名を持つすべての要素を返します。検索は大文字と小文字を区別して child: 軸 対して実行されます。コンテキストノードは、検索される要素の親ノードである必要があります。

□ サンプル

• altova:elements("MyElement") はMyElement()* を返します。

altova:elements(ElementName as xs:string, SearchOptions as xs:string) aselement()*
XP3.1 XQ3.1

入力引数 Element Name 内で与えられた名前と同じローカル名を持つすべての属性を返します。検索は大文字と小文字を区別し、child:: 軸つ対して行われます。コンテキストノードが親要素ノードである必要があります。第2引数はオプションのプラングを含みます。使用することのできるフラックは以下の通りです:

▼ = 正規表現検索を切り替えます: ElementName は正規表現検索文字列である必要があります:

f = このオプションが指定されている場合、ElementName は完全一致を提供します。それ以外の場合は、ElementName は属性名に部分的に一致するとその属性を返します。例えば **f** が指定されていい場合、MyElem はMyElement を返します。 **i = 大文字と小文字を一致させる検索に切り替えます。**

p = 検索に名前空間プレフィックスを含みます。ElementName は、名前空間プレフィックスを含みます。例えば altova:MyElement。

プラッグは順序に関わりなく書き込むことができます。無効なプラッグはエラーを生成します。1つまけば複数のプラッグを省略することができます。空の文字列は許可されていますが、引数を1つしか特かな、関数と同じ効果を持ちまず、前の署名)、ですが、空のシーケンスは第2の引数としては許可されていません。

■ サンプル

- altova:elements("MyElement", "rip") はMyElement()* を返します。
- altova:elements("MyElement", "pri") はMyElement()* を返します。
- altova:elements("MyElement", "") はMyElement()* を返します。
- altova:attributes("MyElem", "rip") はMyElement()* を返します。
- altova:attributes("MyElements", "rfip") 不一致を課wします。
- altova:elements ("MyElement", "Rip") 認識されないフラックエラーを返します。
- altova:elements("MyElement",) 見つかけれ第2引数を返します。

▼ find-first [altova:]

altova:find-first((Sequence as item()*), (Condition(Sequence-Item as xs:boolean)) をitem()? とする XP3.1 XQ3.1

この関数は2 つの別数を必要とます。最初の別数は1 つ、ませは1 つ以上のデータ型のアイテムのシーケンスです。第2 の別数 Condition は(1 のアリティを持つ)1 つの別数 を必要と、boolean を返す XPath 関数に対する参照です。
Condition で参照された関数の代わりに、Sequence の各アイテムが提出されます。(注意: この関数は1 つの別数のみを必要とします。) Condition 内の関数にtrue()と評価させる最初の Sequence アイテムは altova:find-first、反復の

∃ サンプル

終了の結果として返されます。

- altova:find-first(5 to 10, function(\$a) {\$a mod 2 = 0}) はxs:integer 6 を返します。 Condition 引数は、\$a とり名のインライン関数を宣言し、定義します。 XPath 3.0 インライン関数 function()を 参照します。altova: Sequence 引数内の各アイテムでは find-first がりなれ、代わりに \$a を入力値としま す。入力値は関数定義(şa mod 2 = 0)内の条件に対してテストされます。この条件を満たす最初の入力値が altova:find-first(この場合は6)の結果として返されます。
- altova:find-first((1 to 10), (function(\$a) {\$a+3=7})) はxs:integer 4 を返します。

更なるサンプル

ファイルC:\Temp\Customers.xml が存在する場合:

• altova:find-first(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) はxs:string C: \Temp\Customers.xml を返します。

ファイルC:\Temp\Customers.xml が存在せず、http://www.altova.com/index.html が存在する場合:

• altova:find-first(("C:\Temp\Customers.xml", http://www.altova.com/index.html を返します。

ファイルC:\Temp\Customers.xml が存在せず、http://www.altova.com/index.html も存在しむ 場合

• altova:find-first(("C:\Temp\Customers.xml", "http://www.altova.com/index.html"), (doc-available#1)) 結果無しを返します。

上のサンプルこついての注意点

- XPath 3.0 関数 doc-available はURI として使用され、ドキュメトノードが提出されたURI で検出される場合 true を返する単一の引数を必要とします。(ですから、提出されたURI でのドキュメトはXMLドキュメトである必 要がります。)
- doc-available 関数は altova:find-first の第2引数である、Conditionで使用するとができます。 これは、1 つの引数(アリティ=1)のみを必要とするからであり、item()を入力(URIとして使用される文字列)とし て、boolean の値を返すからです。
- doc-available 関数は参照されているだけで、呼び出されていない点に注意してください。アタッチされている#1 サ フィックスは関数が1つのア・ティであることを表示するためです。doc-available#1の意味は以下のとおいです。アリ ティー を持つ doc-availabe() 関数を使用し、最初のシーケンスの各アイテムの代わりに単一引数としていてします。 この結果、2つの文字列の各自つよ文字列をURIとして使用し、URIIさキュメイノードが存在するかテストする doc-available()に vされます。1 つか答当する場合、doc-available() 関数はtrue() を評価し、シー ケンス内のその文字列のインデックス ポジションは、altova:find-first 関数の結果として返されます。docavailable() 関数に関しての注意点:相対パスよ デフォルトで関数がロードされるXML ドキュメトの現在のベース URI(対して相対的に解決されます。
- find-first-combination [altova:]

```
altova:find-first-combination((Seq-01 as item()*), (Seq-02 as item()*),
(Condition (Seg-01-Item, Seg-02-Item as xs:boolean)) をitem()* とする XP3.1 XO3.1
この関数は3つの引数を必要とします:
```

- 最初の2 つの引数 seq-01 とseq-02, は、1 つまけは つ以上のデータ型のアイテムです。
- 第3の別数 Condition は(2のアケを持つ) 2つの別数を必要と、boolean を返す XPath 関数に対しての参照です。

Seq-01 とSeq-02 のアイテム
対指定された組み合わせ、各シーケンスからの1つずつにアイテムで構成されるペア)で、Condition 内の関数の引数として、いされました。組み合わせは以下のようは指定されて、ます。

```
If Seq-01 = X1, X2, X3 ... Xn And Seq-02 = Y1, Y2, Y3 ... Yn Then (X1 \ Y1), (X1 \ Y2), (X1 \ Y3) ... (X1 \ Yn), (X2 \ Y1), (X2 \ Y2) ... (Xn \ Yn)
```

Condition 関数にtrue() と評価するように指示する最初のペアはaltova:find-first-combination の結果として 返されます。以下の点に注意してください(i) Condition 関数が提出された引数ペア内で繰り返され、true()を評価しない場合、altova:find-first-combination は結果を返しません(ii) altova:find-first-combination の結果 が常にデータ型のアイテムのペアである場合ませゴアイテムでは場合。

■ サンプル

- **altova:find-first-combination**(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) はxs:integers (11, 21) のシーケンスを返します。
- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) はxs:integers (11, 22) の一かを返します。
- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 34}) はxs:integers (11, 23) の一ケンを返します。

find-first-pair [altova:]

altova:find-first-pair((Seq-01 as item()*), (Seq-02 as item()*), (Condition(Seq-01-Item, Seq-02-Item as xs:boolean)) をitem()* とする xP3.1 xQ3.1 この関数に3つの別数を必要とます:

- 最初の2 つの引数 Seq-01 とSeq-02, は、1 つまけは つ以上のデータ型のアイテムです。
- 第3の別数 Condition は(2のアケを持つ) 2つの別数を必要と、boolean を返す XPath 関数に対しての参照です。

Seq-01 とSeq-02 のアイテム お指定された組み合わせで、Condition 内の関数の引数として、欠されました。組み合わせお以下のよう:指定されています。

```
If Seq-01 = X1, X2, X3 ... Xn And Seq-02 = Y1, Y2, Y3 ... Yn Then (X1 \ Y1), (X2 \ Y2), (X3 \ Y3) ... (Xn \ Yn)
```

Condition 関数にtrue() と評価するようは示する最初のペアはaltova:find-first-pair の結果として返されます。以下の点に注意してください(i) Condition 関数が提出された引数ペア内で繰り返され、true() を評価しない場合、altova:find-first-pair は 結果を返しません(ii) altova:find-first-combination の結果が常に(データ型の) アイテムのペアである場合ませるアイテムでは場合。

□ サンプル

- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) は xs:integers (11, 21) のシーケンスを返します。
- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) は結果 無しを返します。

上の2 つのサンプルに表示される通り、ペアの順序は以下の通りです: (11, 21) (12, 22) (13, 23) ...(20,

30)。(33 を返す指示されたペアがは ヤーめ) この理由で第2 のサンプルは 結果無しを返します。

find-first-pair-pos [altova:]

altova:find-first-pair-pos((Seq-01 as item()*), (Seq-02 as item()*), (Condition(Seq-01-Item, Seq-02-Item as xs:boolean)) をxs:integer とする XP3.1 XQ3.1 この関数は3つの別数を必要とます:

- 最初の2つの引数 Seq-01 とSeq-02, は1つまけはつ以上のデータ型のアイテムです。
- 第3の別数 Condition は(2のアケを持つ) 2つの別数を必要とし、boolean を返す XPath 関数に対しての参照です。

Seq-01 とSeq-02 のアイテム 指定された組み合わせで、Condition 内の関数の引数としていてされました。組み合わせは以下のように指定されています。

```
If Seq-01 = X1, X2, X3 ... Xn And Seq-02 = Y1, Y2, Y3 ... Yn Then (X1 \ Y1), (X2 \ Y2), (X3 \ Y3) ... (Xn \ Yn)
```

Condition 関数にtrue()を評価させる。最初に指示されたペアのインデックスポジョンは、はaltova:find-first-pair-pos の結果として返されます。関数が提出された引数ペア内で繰り返され、true()を一度も評価しない場合、altova:find-first-pair-pos 結果無しが返します。

∃ サンプル

- altova:find-first-pair-pos(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) は 1 を返します
- altova:find-first-pair-pos(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) は結果無しを返します。

上の2 つのサンプルに表示される通り、ペアの順序は以下の通りです: (11, 21) (12, 22) (13, 23)...(20, 30)。最初のサンプルでは、最初のペアはCondition 関数にtrue() を評価させ、シーケンス内のインデックス ポジンコンには1 が返されます。第 2 のサンプルでは、33 を返すペアがは ナーめ、結果無しが返します。

find-first-pos [altova:]

altova:find-first-pos((Sequence as item()*), (Condition(Sequence-Item as xs:boolean))をxs:integer とする XP3.1 XQ3.1

この関数は2 つの別数を必要とします。最初の別数は1 つ、ませは1 つ以上のデータ型のアイテムのシーケンスです。第2 の別数 Condition は(1 のアリティを持つ)1 つの別数 を必要とし、boolean を返す XPath 関数に対する参照です。
Condition で参照された関数の代わりに Sequence の各アイテムが提出されます。(注意:この関数は1 つの別数のみを必要とします。) Condition 内の関数にtrue() と評価させる最初の Sequence アイテムは altova:find-first-posの 結果として返された Sequence 内インデックスポジションを持ちます。

■ サンプル

• altova:find-first-pos(5 to 10, function(\$a) {\$a mod 2 = 0}) はxs:integer 2 を返します。

Condition 引数は、\$a とう名のインライン関数を宣言し、定義します。XPath 3.0 インライン関数 function()を参照します。Sequence 引数内の各アイテムでは、find-first-pos がいされ、代わりに、\$a を入力値とします。入力値は関数定義(\$a mod 2 = 0)内の条件に対してテストされます。この条件を満たす最初の入力値がaltova:find-first-pos((シーケンス内で)条件を満たす最初の値である6 がシーケンスのインでクス位置 2 にあるためこの場合は2、)の結果として返されます。

• altova:find-first-pos((2 to 10), (function(\$a) {\$a+3=7})) はxs:integer 3 を返します。

更なるサンプル

ファイルC:\Temp\Customers.xml が存在する場合:

• altova:find-first-pos(("C:\Temp\Customers.xml",
 "http://www.altova.com/index.html"), (doc-available#1)) は1 を返します。

ファイルC:\Temp\Customers.xml が存在せず、http://www.altova.com/index.html が存在する場合:

• altova:find-first-pos(("C:\Temp\Customers.xml",
 "http://www.altova.com/index.html"), (doc-available#1)) は2 を返します。

ファイルC:\Temp\Customers.xml が存在せず、http://www.altova.com/index.html も存在しむ 場合

• altova:find-first-pos(("C:\Temp\Customers.xml",
 "http://www.altova.com/index.html"), (doc-available#1)) 結果無しを返します。

上のサンプルこついての注意点

- XPath 3.0 関数 doc-available はURI とて使用され、ドキュメトノードが提出されたURI で検出される場合 true を返する単一の別数を必要とします。(ですから、提出されたURI でのドキュメトはXMLドキュメトである必要がいます。)
- doc-available 関数は altova:find-first-pos の第2引数である Condition で使用することができます。これは、1つの引数(アケティ)のみを必要とするからであり、item()を入力(URI として使用される文字列)として、boolean の値を返すからです。
- doc-available 関数は参照されている対すで、呼び出されていたい点に注意してください。アタッチされている#1 サフィックスは関数が1 つのアライであることを表示するすめです。doc-available#1 の意味は以下のとおりです: アリティニ1 を持つ doc-availabe() 関数を使用し、最初のシーケンスの各アイテムの代わりに、単一引数としていなします。この結果、2 つの文字列の各自つは、文字列をURIとして使用し、URIはギュメントノードが存在するかテストするdoc-available() にいてきれます。1 つが各当する場合、doc-available() 関数はまtrue() を評価し、シーケンス内のその文字列のインデックス ボジュンは、altova:find-first-pos 関数の結果として返されます。doc-available() 関数に関しての注意点: 相対パスは、デフォルトで関数がロードされる XML ドキュメントの現在のベース URI に対して相対的に解決されます。
- for-each-attribute-pair [altova:]

altova:for-each-attribute-pair(Seq1 as element()?, Seq2 as element()?, Function as
function()) asitem()* XP3.1 XQ3.1

ペアの一つの属性が最初の要素を取得し、もう一つの属性が、番目の要素が、取得された箇所で、最初の2つの月数は、2つの要素、および、属性ペアを作成するために使用された属性を識別します。要素ペアは同じ名前を持つことをベースに選択され、ペアは名前別に、アルファベト順にセットに並べ替えられます。属性のために他の要素上に対応する属性が存在しな、場合、ペアのジョイトは解除されます。これは、一つのメンバーには飲み構成されることを意味します。関数アイテム(3番目の月数 Function)はアイテムのシーケンスである出力であるペアのシーケンス内のペア(ジョイント、ジョイントの解除)に固別に適用されます。

□ サンプル

• altova:for-each-attribute-pair(/Example/Test-A, /Example/Test-B, function(\$a, \$b) {\$a+b}) は以下を返します。...

```
(2, 4, 6) if

<Test-A att1="1" att2="2" att3="3" />

<Test-B att1="1" att2="2" att3="3" />

(2, 4, 6) if

<Test-A att2="2" att1="1" att3="3" />

<Test-B att3="3" att2="2" att1="1" />

(2, 6) if

<Test-A att4="4" att1="1" att3="3" />

<Test-B att3="3" att2="2" att1="1" />
```

<u>メモ</u> 結果 (2, 6) は次のアクションにより取得されます: (1+1, ()+2, 3+3, 4+()) 。演算子の一つが空のシーケンスの場合、アイテム 2 と 4 の場合同様、結果は空のシーケンス」なります。

• altova:for-each-attribute-pair(/Example/Test-A, /Example/Test-B, concat#2) は以下を返します。

```
(11, 22, 33) if

<Test-A att1="1" att2="2" att3="3" />

<Test-B att1="1" att2="2" att3="3" />

(11, 2, 33, 4) if

<Test-A att4="4" att1="1" att3="3" />

<Test-B att3="3" att2="2" att1="1" />
```

for-each-combination [altova:]

- altova:for-each-combination(('a', 'b', 'c'), ('1', '2', '3'), function(\$i, \$j) {\$i || \$j}) は('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3') を返します。
- for-each-combination [altova:]

altova:for-each-combination(FirstSequence as item()*, SecondSequence as item()*, Function(\$i,\$j){\$i || \$j}) asitem()* XP3.1 XQ3.1 最初の2つの別数内の2つのシーケンスのアイテムは最初のシーケンスの各アイテムが番目のシーケンスはるように結合されます。3番目の別数として与えられる関数は結果のシーケンス内の組み合わせに適用され、アイテムのシーケンスである出力内で出力されます。サンプル参照。

□ <u>サンプル</u>

```
• altova:for-each-combination(('a', 'b', 'c'), ('1', '2', '3'), function($i, $j) ($i || $j} ) は('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3') を返します。
```

substitute-empty [altova:]

altova: substitute-empty(FirstSequence as item()*, SecondSequence as item()) をitem()*
とする XP3.1 XQ3.1

FirstSequence が空の場合、SecondSequence を返します。FirstSequence が空では場合、FirstSequence を返します。

■ サンプル

- altova: substitute-empty((1,2,3), (4,5,6))は(1,2,3)を返します。
- altova: substitute-empty((), (4,5,6)) は(4,5,6) を返します。

12.2.1.8 XPath/XQuery 関数: 文字列

Altova の文字列拡張関数はXPath とXQuery 式で使用することができ、XML スキーマの異なる日付およい時刻データ型で保存されているデータを処理するさかの追加機能を提供します。このセクションの関数は、Altova のXPath 3.0 およびXQuery 3.0 エンジンで使用することができます。これらの関数は、XPath/XQuery コンテキストで使用することができます。

関数の名前指定と言語の適用性は関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラナで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions にいないされており、altova:プレフィックスが、このセクションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、まけは固別の関数の振る舞いは変更する可能性があることに注意してくたさい。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

camel-case [altova:]

altova:camel-case(InputString as xs:string) をxs:string とする xP3.1 xQ3.1 入力文字列をInputString をキャメルケースで返します。文字列は空白スペースのショートカルである) 正規表現 '\s' を使用して分析されます。空白文字おけば連続する空白文字のシーケンスの後の最初の非空白スペース文字は大文字です。出力文字列の最初の文字は大文字です。

■ サンプル

- altova:camel-case("max") Max を返します。
- altova:camel-case("max max") Max Max を返します。
- altova:camel-case("file01.xml") File01.xml を返します。
- altova:camel-case("file01.xml file02.xml") File01.xml File02.xml を返します。
- altova:camel-case("file01.xml file02.xml") File01.xml File02.xml を返します。
- altova:camel-case("file01.xml -file02.xml") File01.xml -file02.xml を返します。

altova:camel-case(InputString as xs:string, SplitChars as xs:string, IsRegex as xs:boolean) をxs:string とする XP3.1 XQ3.1

SplitChars を使用して、次の大文字をトリガーする文字を決定し、入力文字列を InputString キャメルケースに変換しま

す。SplitChars は、IsRegex = true()の場合、おは、IsRegex = false()の場合、ルーン文字は正規表現と して使用されます。出力文字列の最初の文字は大文字です。

🖃 サンプル

- altova:camel-case("setname getname", "set|get", true()) setName getName を返しま
- altova:camel-case("altova\documents\testcases", "\", false()) Altova\Documents\Testcases を返します。

char [altova:]

altova:char(Position as xs:integer) をxs:string とする XP3.1 XQ3.1 xs:string.Position 「対してのコンテキストアイテムの値を変換することに以得られた文字列内のPosition 引数に以指 定されたポジションにある文字を含む文字列を返します。引数により提出されたインデックスに文字が存在してむ場合、結果文字列は 空です。

□ サンプル

コンテキスト アイテムが1234ABCD の場合:

- altova:char(2) は2 を返します。
- altova:char(5) はA を返します。
- altova: char (9) は空の文字列を返します。
- altova:char(-2) は空の文字列を返します。

altova:char(InputString as xs:string, Position as xs:integer) & xs:string & XP3.1

引数とて提出された文字列内のPosition 引数にお指定されたポジョンでの文字を含む文字列を返します。Position 引数により提出されたインデックスに文字が存在しない場合、結果文字列は空です。

🖃 サンプル

- altova:char("2014-01-15", 5) は- を返します。
- altova:char("USA", 1) はu を返します。
- altova: char ("USA", 10) は空の文字列を返します。
- **altova:char**("USA", -2) は空の文字列を返します。

create-hash-from-string[altova:]

altova:create-hash-from-string(InputString as xs:string) asxs:string XP2 XQ1 XP3.1

altova:create-hash-from-string(InputString as xs:string, HashAlgo as xs:string)

asxs:string XP2 XQ1 XP3.1 XQ3.1

HashAlgo 引数により指定されている ッシュアルゴリズムを使用して InputString から ッシュ文字列を生成します。次の ッ シュアルゴリズムは(大文字、まけよ 小文字で)指定されている可能性があります: MD5, SHA-224, SHA-256, SHA-384, SHA-512。(最初の署名を参照してください) 2番目の別数が指定されていない場合、SHA-256 ハッシュアルゴリズムが使用され ます。

□ サンプル

- altova:create-hash-from-string('abc') はSHA-256 ハッシュアルゴリズ上を使用して生成されたハッシュ 文字列 を返します。
- altova:create-hash-from-string('abc', 'md5') はMD5 / ツシュアルゴリズムを使用して生成された/ ツ シュ文字列を返します。
- altova:create-hash-from-string('abc', 'MD5') は MD5 / ハジュアルゴバムを使用して生成された

ハッシュ文字列を返します。

▼ first-chars [altova:]

altova:first-chars (X-Number as xs:integer) をxs:string とする XP3.1 XQ3.1 xs:string に対するコンテキストアイテムの値を変換することにお得られた最初のX-Number 文字を含む文字列を返します。 サンプル

コンテキストアイテムが1234ABCD の場合:

- altova:first-chars(2) は12 を返します。
- altova:first-chars(5) は1234A を返します。
- altova:first-chars(9) は1234ABCD を返します。

altova:first-chars(InputString as xs:string, X-Number as xs:integer) をxs:string とする
XP3.1 XO3.1

InputString 引数とて提出された文字列の最初の文字を含む文字列を返します。

∃ サンプル

- altova:first-chars("2014-01-15", 5) は2014- を返します。
- altova:first-chars("USA", 1) はU を返します。

format-string [altova:]

altova:format-string(InputString as xs:string, FormatSequence as item()*) asxs:string

入力文字列(最初の別数) には、配置/ ラメーター(%1, %2, etc) か含まれています。各/ ラメーターは(2番目の別数として提出されている)フォーマ・トシーケンス内の対応するポジションでロケートされる文字列アイテムと置き換えられます。フォーマ・トシーケンス内の最初のアイテムは、配置/ ラメーター%1 を置換、二番目のアイテムは%2 を置き換えます。関数は、書式設定された代替を持つ文字列を返します。配置/ ラメーターに文字列が存在しない場合は、配置/ ラメーターが返されます。これは、配置/ ラメーターのインデッケスが書式シーケンス内のアイテムの数と見大きい場合発生します。

■ サンプル

- altova:format-string('Hello %1, %2, %3', ('Jane', 'John', 'Joe')) は "Hello Jane, John, Joe" を返します。
- altova:format-string('Hello %1, %2, %3', ('Jane', 'John', 'Joe', 'Tom')) は"Hello Jane, John, Joe" を反します。
- altova:format-string('Hello %1, %2, %4', ('Jane', 'John', 'Joe', 'Tom')) は "Hello Jane, John, Tom" を返します。
- altova:format-string('Hello %1, %2, %4', ('Jane', 'John', 'Joe')) は "Hello Jane, John, %4" を返します。

last-chars [altova:]

altova:last-chars(X-Number as xs:integer) <mark>をxs:string とする XP3.1 XQ3.1</mark> xs:string (対してのコンテキストアイテムの値の変換より取得された文字列の最後のX-Number 文字を含んでい文字列を返します。

<u> サンプル</u>

ゴンテキストアイテムが1234ABCD の場合:

- altova:last-chars(2) はCD を返します。
- altova:last-chars(5) は4ABCD を返します。
- altova:last-chars(9) は1234ABCD を返します。

altova:last-chars(InputString as xs:string, X-Number as xs:integer) asxs:string xps.number as xs:integer) asxs:string

引数として提出された文字列の最後のX-Number 文字を含んている文字列を返します。

■ サンプル

- altova:last-chars("2014-01-15", 5) は01-15 を返します。
- altova:last-chars("USA", 10) はUSA を返します。

pad-string-left [altova:]

 ${\bf altova:pad-string-left(StringToPad} \ \ as \ \ xs:string, \ \ {\bf StringLength} \ \ as \ \ xs:integer,$

PadCharacter as xs:string) をxs:string とする XP3.1 XQ3.1

PadCharacter 引数は1文字です。文字列の左側つがかされ、この数が、StringLength 引数の整数の値と等しくなるよう IStringToPad の文字の数を増やします。StringLength 引数は任意の整数の値(正数ませは負数)を持つことができますが、1分イングは、StringLength の値がStringToPad 内の文字数と比多い場合の場合の場合によす。もし、StringToPad がStringLength の値と比多くの文字数も持つ場合、StringToPad は変更されません。

■ サンプル

- altova:pad-string-left('AP', 1, 'Z') は'AP' を返します。
- altova:pad-string-left('AP', 2, 'Z') は'AP' を返します。
- altova:pad-string-left('AP', 3, 'Z') は'ZAP' を返します。
- altova:pad-string-left('AP', 4, 'Z') は'ZZAP' を返します。
- altova:pad-string-left('AP', -3, 'Z') は'AP' を返します。
- altova:pad-string-left('AP', 3, 'YZ')は[パド文字が張すぎます] エラーを返します。

pad-string-right [altova:]

altova:pad-string-right(StringToPad as xs:string, StringLength as xs:integer,

PadCharacter as xs:string) をxs:string とする XP3.1 XQ3.1

PadCharacter 引数は文字です。文字列の右側つがだされ、この数が、StringLength 引数の整数の値と等しくなるよう InstringToPad の文字の数を増やします。StringLength 引数は任意の整数の値(正数ませは負数)を持つことができますが、パディングは、StringLength の値がStringToPad 内の文字数と比多い場合の場合の分発生します。もし、StringToPad がStringLength の値と比多くの文字数も持つ場合、StringToPad は変更されません。

□ サンプル

- altova:pad-string-right('AP', 1, 'Z') を'AP' を返します。
- altova:pad-string-right('AP', 2, 'Z') を'AP' を返します。
- altova:pad-string-right('AP', 3, 'Z') を'APZ' を返します。
- altova:pad-string-right('AP', 4, 'Z') を'APZZ' を返します。
- altova:pad-string-right('AP', -3, 'Z') を'AP' を返します。
- altova:pad-string-right('AP', 3, 'YZ') は[パパ文字が張すぎます] エラーを返します。

repeat-string [altova:]

altova:repeat-string(InputString as xs:string, Repeats as xs:integer) &xs:string

XP2 XO1 XP3.1 XO3.1

最初のInputString引数により構成される文字列、Repeats 回繰り返してを生成します。

□ サンプル

- altova:repeat-string("Altova #", 3) は"Altova #Altova #Altova #"を返します。
- substring-after-last [altova:]

altova:substring-after-last(MainString as xs:string, CheckString as xs:string) & xs:string

xs

CheckString がMainString 内で検出された場合、MainString 内のCheckString が発生した後のサブ文字列が返されます。MainString 内でCheckString が検出されば、場合、空の文字列が返されます。CheckString が空の文字列の場合、MainString 全体が返されます。一度以上発生する場合、CheckString の最後の発生の後のサブ文字列が返されます。

∃ サンプル

- altova: substring-after-last('ABCDEFGH', 'B') は'CDEFGH' を返します。
- altova:substring-after-last('ABCDEFGH', 'BC') は'DEFGH'を返します。
- altova: substring-after-last('ABCDEFGH', 'BD') は'' を返します。
- altova: substring-after-last('ABCDEFGH', 'Z') は''を返します。
- altova: substring-after-last('ABCDEFGH', '') は'ABCDEFGH' を返します。
- altova: substring-after-last('ABCD-ABCD', 'B') は'CD' を返します。
- altova:substring-after-last('ABCD-ABCD-ABCD', 'BCD') は'' を返します。
- substring-before-last [altova:]

altova:substring-before-last(MainString as xs:string, CheckString as xs:string) をxs:string とする XP3.1 XQ3.1

CheckString がMainString 内で検出された場合、MainString 内のCheckString が発生する前のサブ文字列が返されます。MainString 内でCheckString が一度以上発生する場合、CheckString の最後の発生の前のサブ文字列が返されます。

■ サンプル

- altova: substring-before-last('ABCDEFGH', 'B') は'A' を返します。
- altova: substring-before-last('ABCDEFGH', 'BC') は'A' を返します。
- altova:substring-before-last('ABCDEFGH', 'BD') は''を返します。
- altova: substring-before-last('ABCDEFGH', 'Z') は'' を返します。
- altova:substring-before-last('ABCDEFGH', '') は'' を返します。
- altova:substring-before-last('ABCD-ABCD', 'B') は 'ABCD-A' を返します。
- altova:substring-before-last('ABCD-ABCD-ABCD', 'ABCD') は 'ABCD-ABCD-' を返します。
- substring-pos [altova:]

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string) をxs:integer とする XP3.1 XQ3.1

StringToCheck 内でのStringToFind の最初の発生の文字位置を整数として返します。StringToCheck の最初の文字は、位置 1 にあます。StringToFind がStringToCheck 内で発生しむ場合、整数 0 が返されます。第 2 封出までの後のStringToCheck、発生を確認するによって関数の次の署名を確認してください。

□ サンプル

• altova: substring-pos('Altova', 'to')は3を返します。

- altova: substring-pos('Altova', 'tov')は3を返します。
- altova: substring-pos('Altova', 'tv') は0を返します。
- altova: substring-pos('AltovaAltova', 'to') は3 を返します。

altova:substring-pos(StringToCheck as xs:string, StringToFind as xs:string, Integer as xs:integer とする XP3.1 XQ3.1

StringToCheck 内でのStringToFind の最初の発生の文字位置を返します。Integer 引数により与えられた、文字位置からStringToFind の検索が開始されます。この位置の前の文字サブ文字列は検索されません。しかし、返された整数は、全体文字列 StringToCheck の検索された文字列の位置です。この署名は、StringToCheck 内で複数回発生する、第2 または後の発生を検索する際に役に立ちます。 StringToFind がStringToCheck 内で発生しな場合、整数 0 が返されます。

□ <u>サ</u>ンプル

- altova:substring-pos('Altova', 'to', 1)は3を返します。
- altova: substring-pos('Altova', 'to', 3)は3を返します。
- altova:substring-pos('Altova', 'to', 4)は0を返します。
- altova: substring-pos('Altova-Altova', 'to', 0) は3を返します。
- altova:substring-pos('Altova-Altova', 'to', 4)は10を返します。

trim-string [altova:]

altova:trim-string(InputString as xs:string) をxs:string とする xP3.1 xQ3.1 この関数はxs:string 引数を必要と、先頭おける後続の空白を削除し、トリミングされたxs:string を返します。 サンプル

- altova:trim-string(" Hello World ")) は"Hello World" を返します。
- altova:trim-string("Hello World ")) は"Hello World" を返します。
- altova:trim-string(" Hello World")) は"Hello World" を返します。
- altova:trim-string("Hello World")) は"Hello World" を返します。
- altova:trim-string("Hello World"))は "Hello World" を返します。

trim-string-left [altova:]

altova:trim-string-left(InputString as xs:string) をxs:string とする xP3.1 xQ3.1 この関数はxs:string 引数を必要とし、先頭おける後続の空白を削除し、トリミングされたxs:string を返します。

■ サンプル

- altova:trim-string-left(" Hello World "))は "Hello World " を返します。
- altova:trim-string-left("Hello World ")) は"Hello World " を返します。
- altova:trim-string-left(" Hello World")) は"Hello World"を返します。
- altova:trim-string-left("Hello World")) は"Hello World" を返します。
- altova:trim-string-left("Hello World")) は"Hello World"を返します。

trim-string-right [altova:]

altova:trim-string-right(InputString as xs:string) をxs:string とする xP3.1 xQ3.1 この関数はxs:string 引数を必要と、先頭おける統の空白を削除し、トリミングされたxs:string を返します。

□ サンプル

- altova:trim-string-right(" Hello World "))は" Hello World"を返します。
- altova:trim-string-right("Hello World "))は"Hello World"を返します。
- altova:trim-string-right(" Hello World")) は" Hello World"を返します。
- altova:trim-string-right("Hello World")) は"Hello World"を返します。
- altova:trim-string-right("Hello World")) は"Hello World"を返します。

12.2.1.9 XPath/XQuerv 関数: その他

XPath/XQuery 拡張関数の以下の一般的な目的は現在のStyleVision バージョンでサポートされており(i) XSLT コンテキスト内の XPath 式 ませは(ii) Xguery ドキュメト内のXQuery 式内で使用することができます。。

関数の名前指定と言語の適用性に関するメモ

Altova 拡張関数はXPath/XQuery 式で使用するこかでき、XPath、XQuery、およびXSLT 関数の標準ライブラケで使用可能な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/xslt-extensions に以められており、altova:プレフィケクスが、このセケションでは使用されます。製品の今後のバージョンが拡張機能への継続的サポート、ませは固別の関数の振る舞いは変更する可能性があることに注意してください。Altova 拡張機能へのサポートに関しては、今後のリースのドキュメートを参照してください。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

decode-string [altova:]

altova:decode-string(Input as xs:base64Binary) as xs:string XP3.1 XQ3.1
altova:decode-string(Input as xs:base64Binary, Encoding as xs:string) as xs:string
XP3.1 XQ3.1

指定されたエンコードを使用して送信されたbase64Binary 入力を文字列にデコードします。エンコードが指定されていな場合 UTF-8 エンコードが使用されます。以下のエンコードがサポートされます。US-ASCII, ISO-8859-1, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4

∃ サンプル

- altova:decode-string(\$XML1/MailData/Meta/b64B) はbase64Binary 入力をUTF-8 エンコード済み文字列とて返します。
- altova:decode-string(\$XML1/MailData/Meta/b64B, "UTF-8") はbase64Binary 入力をUTF-8-X分割をUTF-8-X分割をUTF-8-X分割をUTF-8-X分割をUTF-8-X分割をUTF-8-X人力のUTF-8-X人力のUTF-8-X人力をUTF-8-X人力をUTF-8-X人力のUTF-8-X人のUTF-8-X人力のUTF-8-X人力のUTF-8-X人力のUTF-8-X人力のUTF-8-X人のU
- altova:decode-string(\$XML1/MailData/Meta/b64B, "ISO-8859-1") はbase64Binary 入力をISO-8859-1 エロード済み文字列とて返します。

encode-string [altova:]

altova:encode-string(InputString as xs:string) as xs:base64Binaryinteger XP3.1 XQ3.1 altova:encode-string(InputString as xs:string, Encoding as xs:string) as

xs:base64Binaryinteger XP3.1 XQ3.1

与えらえている場合指定されているエンコードを使用して送信された文字列をエンコードします。エンコードが与えられているは場合 UTF-8 エンコードが使用されます。エンコードされた文字列はbase64Binary 文字に変換され、エンコードされた文字列はbase64Binary 文字に変換され、エンコードされた base64Binary 値が返されます。最初にUTF-8 エンコードがサポートされ、サポートは以下のエンコードに拡張されます。US-ASCII、ISO-8859-1、UTF-16LE、UTF-16BE、ISO-10646-UCS2、UTF-32、UTF-32LE、UTF-32BE、ISO-10646-UCS4

■ サンプル

- altova:encode-string("Altova") はUTF-8 エンコード済み文字列 "Altova" の等価である base64Binary を返します。
- altova:encode-string("Altova", "UTF-8") はUTF-8 エンコード済み文字列 "Altova" の等価であるbase64Binary のを返します。
- get-temp-folder [altova:]

altova:get-temp-folder() をxs:string とする XP2 XQ1 XP3.1 XQ3.1 この関数は関数を必要としません。この関数は現在のユーザーの一時的なフォルダーへの いを返します。

- *サンプル*
 - altova:get-temp-folder() は マンシ上で、xs:string として C: \Users\<UserName>\AppData\Local\Temp\ と類似したプを返します。
- generate-guid [altova:]

altova:generate-guid() asxs:string XP2 XQ1 XP3.1 XQ3.1 ユニーグは文字列 GUID 文字列を生成します。

- サンプル
 - altova:generate-guid() ば例えば 85F971DA-17F3-4E4E-994E-99137873ACCD を返します。
- high-res-timer [altova:]

altova:high-res-timer() asxs:double XP3.1 XQ3.1

砂数でシステム・イれぞリューションタイマーの値を返します。ハイれぞリューションタイマーがシステム内で存在すると、必要とされる場合、高度に正確な時間の計算を有効化します(例えば、アニメーションと正確なコード実行の時間の決定などが例として挙げられます)。この関数は、システムのタイマーに精度を与えます。

- ∃ サンプル
 - altova:high-res-timer() は '1.16766146154566E6' などを返します。
- parse-html [altova:]

altova:parse-html (HTMLText as xs:string) asnode() XP3.1 XQ3.1

HTMLText 引数は HTMLドキュメトのテキストを含む文字列です。関数は、文字列からHTML ツーを作成します。提供された文字列は HTML 要素を含む、まけよ 含まむ、場合がかます。しずれの場合でも、ツーのルート要素は、HTML と名付けられます。提出された文字列内のHTML コードが有効な HTMLであることを確認することが奨励されます。.

- サンプル
 - altova:parse-html("<html><head/><body><h1>Header</h1></body></html>") は提供された
 文字列からHTML ツーを作成します。

sleep[altova:]

altova:sleep(Millisecs as xs:integer) asempty-sequence() XP2 XQ1 XP3.1 XQ3.1 Millisecs に以与えられるいかで示される期間のための実行を延期します。

 サンプル

• altova:sleep(1000) は1000 ミトかのナーがの実行を延期します。

[<u>[トップ]</u>]

12.2.2 その他の拡張関数

Java やC# などのプログラング言語 コま XPath 2.0 / XQuery 関数、お出まXSLT 2.0 関数として利用できな、関数が、くつからはす。そのようは関数の良い例として、Java で利用することのできるsin()やcos()という数学関数があります。 XSLT スタイルシートやXQuery のプロルーでこれらの関数が利用できるのであれば、スタイルシートやプロの適用範囲を大幅は拡張することができ、スタイルシート作成タスクの負担が大幅は軽減されます。 Altova 製品で使用されている Altova エンジン(XSLT 1.0、XSLT 2.0、XQuery 1.0)では、Java や.NET および MSXSL scripts for XSLTにおける拡張関数の使用がサポートされます。 このセグションでは、拡張機能および XSLT スタイルシート内で MSXSL スクレプト。を使用する方法について記述します。 使用できる拡張関数は以下のように構成されます:

- Java 拡張関数
- .NET 拡張関数
- XSLT は対するMSXSL スクレプト

記述の中では特に(i)関連するライブラル内の関数がどのように呼ばれるか(ii)関数呼び出しを行う際に入力として使用される引数を変換するのにどのようなルールが適用され、return により値が返される際にどのような変換ルールが適用されるのが(XSLT/XQuery データオープジェクトに対する関数の結果)について説明されます。

必要条件

拡張関数のサポーを有効にするコは、XSLT 変換やXQuery の実行を行うエレニーターにJava Runtime Environment (Java 関数にアクセスする場合) ならびに、NET Framework 2.0 以上 (.NET 関数にアクセスする場合) がインストールされている まけっぱった スできる環境が整っている必要があます。

12.2.2.1 Java 拡張関数

Java 拡張関数は、XPath おさはXQuery 条件式にて使用することができるまか、Java のコンストラクターを呼び出したり、Java のお 的まさはインスタンス)メントドを呼び出すことができます。

Java クラスのフィールドは、引数を持たないメンドとして扱われます。フィールドは静的ませはインスタンスとして存在することができます。フィールドへのアクセス方法については、静的とインスタンスの両方について、以下のサブセグションこで記述されます。

このセグションは以下のサブセグションにより構成されます:

- <u>Java:コレストラクター</u>
- Java:静的メンドと静的フィールド

- Java: インスタンスメナルドとインスタンスフィールド
- データ型: XPath/XQuery から Java へ
- データ型: Java から XPath/XQuery へ

以下の点に注意してください

- Altova デスケップ製品を使用している場合、Altova アプケーションは、Java 仮想マンンへの やを以下の順序で読み取ることには自動的に検知しようとします: (i) Windows レジストリ(ii) JAVA_HOME 環境変数。アプケーションのオプションダイアログトにカスタム やを追加することもできます。このエントリは自動的に検知された他のJava VM パス以上の優先順位を有します。
- Altova サード製品がWindows マンン上で作動している場合、Java 仮想マンンへのいなはWindows レジストリから最初に読み取られます。成功しない場合、JAVA HOME 環境変数が使用されます。
- Altova サード製品を作動する場合、および、Linux おけまmacOS マンン上でサード製品を作動する場合、Java 仮想マシンへの やがJAVA_HOME 環境変数内に保管されていることを確認してください。\bin\server おけま\bin\client デルノトリ内のjvm.dl1 ファイルを指している必要がおます。

拡張関数のフォーム

XPath/XQuery 条件式における拡張関数では、prefix:fname()の形式を取る必要が決ます。

- prefix: 部には拡張関数がJava 関数とて認識されます。java:から始まるURI のスコープ内の名前空間宣言に拡張関数を関連付けることでJava 関数であるとう認識が行われます。名前空間の宣言により、例えばxmlns:myns="java:java.lang.Math"とう Java クラスが特定されます。名前空間の宣言は、(コロン無しの)xmlns:myns="java"とう形式で、Java クラスの識別子を拡張関数にある fname() 部の左型に配置することでも行うことができます。
- fname() 部により、呼び出されている Java メンドが識別され、メンドの別数が提供されます(以下の例を参照ください)。 prefix: 部にて識別された名前空間 URI が Java クラスを識別できなかった場合、 Java クラスの識別はクラスの前にる fname() 部にて行うことしてい、ピリオドによりラスから分離されることしています(以下にある2番目のXSLT サンプルを参照)。

メモ 呼び出されるクラスはエンピューターのクラス、ストにある必要がありあす。

XSLT サンプル

以下に静的メンドを呼び出す2つのサンプルを示します。最初のサンプルでは、クラス名(java.lang.Math)が名前空間 URI に加えられており、fname() へ加えることはできません。2 番目のサンプルでは、prefix: 部にjava: が与えられており、fname() 部にてクラスとメンドが識別されます。

拡張関数内にあるメンバ名(上の例ではcos())は、名前付き Java クラス(上の例ではjava.lang.Math)のpublic な静的メンバの名前に一致する必要が訳ます。

XQuery サンプル

以下にXSLT のサンプルに似たXQuery のサンプルを示します:

```
<cosine xmlns:jMath="java:java.lang.Math">
  {jMath:cos(3.14)}
</cosine>
```

ユーザー定義された Java クラス

独自のJava クラスやメソドを作成した場合、(i) JAR ファイル(ま)は class ファイル)を介してこれらクラスファイルへアクセスしているか (ii) これら(JAR ま)は class) ファイルが、カレントディノオ・リ(XSLT やXQuery ドキュメントが存在するディレクトリ) 「配置されているか」 により、これらクラスの呼び出し方法が変わってきます。 これらファイルの特定方法については、ユーザー定義グラスファイルならびにユーザー定義 JAR ファイルを参照ください。 カレントディレクトリロは無い グラスファイルや JAR ファイルへの りは指定しなければならないとはつ注意してください。

メモ Style Vision を使って SPS から生成された XSLT スタイルシートへ名前空間を加えるコよ SPS がベース している XML スキーマのトップレベル こある schema 要素へ名前空間を追加する必要が決ます。 Style Vision して作成された全ての SPS では、デフォルで xmlns:java="java"とう名前空間宣言が自動的に作成されることに注意してくたさい。

12.2.2.1.1 ユーザー定義のクラスファイル

アクセスかりラスファイルを介したものである場合、4つのケースか考えられます:

- ケラスファイルは、シケージである。XSLT まけはXQuery ファイルがJava パッケージと同じ場所は収められている。(下のサンプルを参照)
- クラスファイルは シケージでは無い。 XSLT ませは XQuery ファイルが Java パッケージと同じ場所に収められている。(下のサンプルを参照)
- クラスファイルは シケージである。XSLT ませま XQuery ファイルがランダムな場所に収められている。(下のサンプルを参照)
- ケラスファイルは シケージである。XSLT ませま XQuery ファイルがランダムな場所に収められている。(下のサンプルを参照)

クラスファイルが シケージではなく、XSLT ませま XQuery ドキュメトと同じ場所に収められているケースを考えてみましょう。この場合、フォルダー内の全ケラスを発見することができるため、ファイルの場所を指定する必要はありません。 クラスの識別を行う構文は以下のようしないます:

java:classname

二で、

java: によりユーザー定義のJava 関数が呼ばれていることが示されます(デフォルトでカレントディンケルリーある Java クラスかロードされます)。

classname は目的となるメントドのケラスか含まれているケラスの名前です。

クラスか名前空間 URI にて識別され、名前空間がシンド呼び出しにて使用されます。

クラスファイルが シケージで、XSLT/XQuery ファイルが Java パッケージと同じ場所に収められている

以下の例では、com.altova.extfunc パッケージにある Car クラスの getVehicle Type () メンゲ が呼び出されて、ます。 com.altova.extfunc パッケージは Java Project と ら名前のフォルダー 電話 かれており、XSLT ファイルも同じフォルダー 電話されて、ます。

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"</pre>
```

クラスファイルか参照され、XSLT/XQuery ファイルかうラスファイルと同じフォルダーコ収められている

下のサンプルではCar クラスのgetVehicleType()メンナが呼び出されます。以下を述べることができます:(i) Car クラスは次のフォルダー内に対はす: JavaProject/com/altova/extfunc。(ii) このフォルダーは下のサンプルの現在のフォルダー内に存在します。XSLT ファイルもフォルダー JavaProject/com/altova/extfunc 内にあります。

クラスファイルが シケージされXSLT/XQuery ファイルかランダムな場所に収められている

以下の例では、com.altova.extfunc パッケージにある Car クラスの getVehicleColor() メンドが呼び出されて、ます。 com.altova.extfunc パッケージは JavaProject と ら名前のフォルダーに置かれており、XSLT ファイルが任意の場所に配置されて、ます。この場合、以下のような構文で、ツケージの場所をクロ文字列として URI 内にて指定する必要があります:

java:classname[?path=uri-of-package]

二で、

java: によりユーザー定義のJava 関数が呼ばれていることを表します。 uri-of-package は Java パッケージの URI です。 classname は目的のメッドが含まれているクラス名です。

クラスは名前空間 URI により特定され、名前空間がシンド呼び出しのプレフィックスで使用されます。以下の例ではカレントディレケトリ以外にあるクラスファイルへのアクセスを行うことができます。

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:car="java:com.altova.extfunc.Car?path=file:///C:/JavaProject/" >
```

クラスファイルが シケージではなく、XSLT/XQuery ファイルかランダムな場所に収められている

以下の例では、com.altova.extfunc パッケージにある Car クラスの getCarColor() メンナが呼び出されて、ます。
com.altova.extfunc パッケージは JavaProject と ら名前のフォルダーに置かれており、XSLT ファイルが任意の場所に配置されて、ます。以下のような構文で、クラスファイルの場所をクエノ文字列として URI 内にて指定する必要があります:

java:classname[?path=uri-of-classfile]

こで

java: によりユーザー定義のJava 関数が呼ばれていることを表します。 uri-of-classfile は Java パッケージのURI です。 classname は目的のメンドが含まれているクラス名です。

クラスは名前空間 URI により特定され、名前空間はメンナド呼び出しのプレフィックスで使用されます。以下の例ではカレントディレクトリ以外にあるクラスファイルへのアクセスを行うことができます。

メモ パスが外部関数により与えられている場合、ClassLoader により、スが追加されます。

12.2.2.1.2 ユーザー定義の JAR ファイル

JAR ファイル経由でアクセスが行われた場合、以下の構文によりJAR ファイルのURI を指定する必要があます:

```
xmlns:classNS="java:classname?path=jar:uri-of-jarfile!/"
```

クラスの識別を行う名前空間 URI のプレフィックスを使用してメンメドが呼び出されます: classNS:method()

上の例に対する説明は以下のとおりです:

java: 関数が呼び出されていることを表します。

```
classname がユーザー定義されたクラスの名前にないます。
? はグラス名と いを分離するために使用されます。
path=jar: により、JAR ファイルへの いか与えられていることを示します。
uri-of-jarfile は JAR ファイルの URI といます。
! / は、終了を表すデにタとなります。
classNS:method() により、メンドの呼び出しが行われます。
```

その他にも、メノンド名とともにクラス名を与えることができます。構文の例を以下に示します:

以下にJAR ファイルを使ったJava 拡張関数を呼び出す XSLT サンプルを記します:

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:car="java?path=jar:file:///C:/test/Car1.jar!/" >
    <xsl:output exclude-result-prefixes="fn car xsl xs"/>

<xsl:template match="/">
        <xsl:variable name="myCar" select="car:Car1.new('red')" />
        <a><xsl:value-of select="car:Car1.getCarColor($myCar)"/></a>
</xsl:template>

<xsl:template match="car"/>
</xsl:stylesheet>
```

メモ: 拡張関数によりなか与えられている場合、ClassLoader になか追加されます。

12.2.2.1.3 Java:コンストラクター

拡張関数を使用することでJava コンストラクターを呼び出すことができます。new() により全てのコンストラクターを呼び出すことができます。

Java エンストラクターの呼び出し結果を、黙示的にXPath/XQuery データ型へ変換できる場合、Java 拡張関数により
XPath/XQuery データ型のシーケンスが返されます。Java エンストラクターの呼び出し結果がXPath/XQuery データ型へ変換できない
場合、値を返すクラス名でラップした Java オブジェケトがエンストラクターにより作成されます。例えば、java.util.Date クラスコ対する
エンストラクターが呼び出された場合(java.util.Date.new())、java.util.Date を持っオブジェケトが返されます。返された
オブジェケトのレギシカルフォーマットは、XPath データ型のレギンカルフォーマットしてッチしたは場合もあり、目的のXPath データ型に対する
レギンカルフォーマットへ値の変換を行い、その後目的のXPath データ型へ変換を行う必要があります。

コンストラクターこと/作成されたJava オブジェクトにより2つのことが行えます:

• 変数への割り当てを行うことができます:

```
<xsl:variable name="currentdate" select="date:new()"
xmlns:date="java:java.util.Date" />
```

拡張関数への受け渡しを行うことができます(インスタンスメソッドならびパニインスタンスフィーハトでを参照くたさい):

```
<xsl:value-of select="date:toString(date:new())"
xmlns:date="java:java.util.Date" />
```

12.2.2.1.4 Java:静的メソッドと静的フィールド

静的メンドは、Java 名ならびにメンドの引数により直接呼び出すことができます。E や PI というに定数の静的フィーリド(引数を持たなし メンド)は 引数を指定することなくアクセスすることができます。

XSLT の例

静的メンドならびにフィールドを呼び出す例を以下に示します:

上の拡張関数は prefix:fname() とう形式を使用していることで注意してください。3つの例にあるプレフィックスは全てjMath: となっており、このプレフィックスは、java:java.lang.Math とう名前空間 URI に関連付けられています。名前空間 URI はjava: で開始しなければなりません。上の例では、クラス名(java.lang.Math)を含むように拡張されています。拡張関数のfname() 部は(java.lang.Math のような)public クラスニマッチする必要があり、その後 public な静的メクドが引数とともに続くかく例: cos(3.14))、public な静的フィールドが続きます(例: PI())。

上の例では、クラス名が名前空間 URI に含まれています。クラス名が名前空間 URI に含まれていない場合、以下の例にあるように、拡張 関数の fname() 部にて追加する必要が決ます:

XQuery の例

XQuery におお似たようなサンプルを以下に示します:

```
<cosine xmlns:jMath="java:java.lang.Math">
   {jMath:cos(3.14)}
</cosine>
```

12.2.2.1.5 Java: インスタンスメソッドとインスタンスフィールド

メンド・呼び出しの第一引数として、一スされる Java オブジェケトが、インスタンスメンド はまうえられています。このような Java オブジェケト は通常、拡張関数を使うことで作成される(例: コンストラクターの呼び出し)か、スタイルシート・プラメーター/変数により作成されます。以下に XSLT サンプルを示します:

```
<xsl:stylesheet version="1.0" exclude-result-prefixes="date"</pre>
```

上の例ではノード enrollment/@type の値が以下のようご作成されます:

- 1. java.util.Date クラスマオするオブジェケトがコンストラクター(date:new() コンストラクター) とともご作成されます。
- 2. jlang.Object.getClass メンドの引数として Java オブジェクトが ~ スされます。
- 3. getClass メノボにより得られたオブジェケトが jlang.Object.toString メノバの引数としてパースされます。

結果(@type の値)は java.util.Date を持った文字列となります。

インスタンスフィールドは、引数としてインスタンスフィールドへ渡される Java オブジェケト ではむという点で、理論的にはインスタンスメンバと 異ないます。パラメーターや変数がその代わりに引数として渡されますが、パラメーター/変数そのものに Java オブジェケトから返された値が含まれている場合もあります。例えば、Current Date パラメーターには 1. Date クラスのエンストラクターから返された値が含まれます。この値は 1数として、date:toString インスタンスメンバーへ渡され、/enrollment/@date の値とて使用されます。

12.2.2.1.6 データ型: XPath/XQuery から Java へ

XPath/XQuery 条件式内部から Java 関数が呼び出された場合、複数ある同名の Java クラスのうち、どのクラスが呼び出されたのか決定するのに、関数へ渡される引数のデータ型が重要になります。

Java では、以下のルールが適用されます:

- 同名のJava メンドが2つ以上あり、それぞれが違う数の引数を受け取る場合、呼び出しこ使用されている引数の数に一番マッチするメンドが選択されます。
- XPath/XQuery の文字列、数値、boolean データ型は、黙示的で対応する Java データ型へ変換されます(以下の)ストを参照)。与えられた XPath/XQuery 型が2つ以上の Java 型へ変換できる場合(例: xs:integer)、選択されたメンドにて宣言されている Java 型が使用されます。例えば、呼び出された Java メンドが fx(decimal) で、与えられた XPath/XQuery データ型がxs:integer の場合、xs:integer が Java の decimal データ型へ変換されます。

以下のテーブル、XPath/XQuery の文字列、数値、boolean 型から Java データ型への黙示的な変換リストを示します。

xs:string	java.lang.String
xs:boolean	boolean (ブミティブ型), java.lang.Boolean
xs:integer	int, long, short, byte, float, double, ならびいに、java.lang.Integer のようなこれらのラッパークラス
xs:float	float (ブミティブ型), java.lang.Float, double (プミティブ型)
xs:double	double (プ ドティブ 型), java.lang.Double

xs:decimal float (プトティブ型), java.lang.Float, double(プチテイブ型), java.lang.Double

上のリストにあるXML スキーマデータ型(ならびにXPath やXQuery で使用されているデータ型)のサブタイプも、対応する相先のサブタイ プとしてJava のデータ型へ変換されます。

場合によっては、与えられた情報から正しいJava メノバを選択することができない場合もあります。例えば、以下のような場合を考えてみま しお:

- 与えられた引数が10 という値を持ってxs:untypedAtomic 型で、mymethod(float) メンゲへ渡されるのを意図してい
- しかし、そのケラスコは別のデータ型を取る mymethod (double) というメントドも存在する。
- メンド名が同じて、与えられた型(xs:untypedAtomic)もfloat とdouble の両方に変換することができるか、 xs:untypedAtomic がfloat ではく double で変換される可能性もある。
- 結果として、意図したメンドは選択されず、予期しない動作結果がを招く可能性がある。この問題を回避するコよ、意図したメ ソゲを使用するユーザー定義のメソゲを別の名前で新たご作成する必要が別ます。

上のJストでか、されていれ、型(例:xs:date)は変換されず、エラーとなります。しかし場合によっては Java コンストラクターを使用し て、目的のJava データ型を作成することが可能が出ってとも留意してください。

12.2.2.1.7 データ型: Java から XPath/XQuery へ

Java メンドによん値が返され、値のデータ型が文字列、数値、まけまboolean 型の場合、対応するXPath/XQuery 型への変換が 行われます。例えば Java のjava.lang.Boolean やboolean データ型はxsd:boolean へ変換されます。

関数から返された一次元配列は、シーケンス(sequence)に展開されます。2次元以上の配列は変換されることが無いかか、ラップして使 用するべきでしょう。

Java オブジェケトや文字列、数値、boolean 以外のデータ型がラップされて返された場合、最初に(例えばtoString といた)Java メ ンゲを使用して Java オブンエケトを文字列へ変換することで、目的の XPath/X Query 型への変換を行います。 XPath/X Query で は、文字列を目的となる型のレキンカルフォーマナトへ変換し、目的の型への変換を(例えば cast as 式を使用することで)行うことができ ます。

12.2.2.2 NET 拡張関数

.NET プラナフォームこて作業を行なっている場合、.NET 言語(例えばC#)で記述された拡張関数を使用することができます。.NET 拡 張関数はXPath やXQuery 条件式内部から使用することができ、NET クラス内部にあるコンストラクターや(static まけはインスタンス 変数)プロノティを呼び出すことができます。

get PropertyName() 構文を使用することにより.NET クラスのプロノティを呼び出すことができます。

このセクションは、以下のサブセクションにより構成されています:

- .NETコンストラクター
- .NET:静的メンドならびご静的フィールド
- .NET: ペンスタンスメン・ドとインスタンスフィールド
- データ型: XPath/XQuery から.NET へ
- 夕型:.NET からXPath/XQuery へ

拡張関数のフォーム

XPath/XQuery 条件式にある拡張関数は、prefix:fname() の形式を取る必要がかます。

- prefix: 部は呼び出されている.NET クラスを特定するURI となります。
- fname () 部により、NET クラス内にあるエレストラクター、プロ 守ィ、まけは、静的まけはインスタンス)メンドが特定され、必要 な場合は引数が与えられます。
- URI はclitype: で開始する必要が決してれてよ関数が、NET 拡張関数であることが認識されます。
- 拡張関数の prefix:fname() 形式は システムクラスならゾコードされたアセンブルともに使用することもできます。しかし ケラスをロードする必要がある場合、必要な情報が含まれる。テメーターが必要になります。

ノプメーター

アセンブルをロードするコお以下のプメーターを使用してくたさい

ロードするアセンブルの名前。 asm

バージョン番号(ピルドにより分離された最大4桁の整数)。 ver

アセンブル厳密名のキートーケン(16新数の数値)。 sn

from ロードするアセンブリ(DLL) の場所を特定するURI。URI が相対パスの場合、XSLT

やXQueryドキュメントノコ対して相対的しています。このプラメーターが指定された場合、

その他のプメーターが無視されます。

partialname アセプ名の一部。Assembly.LoadWith.PartialName() へ渡され、アセブ

リヘのロードが試みられます。 partialname が指定された場合、その他のプメータ

一か無視されます。

loc 例えばen-US とうロケール。デフォルトはneutral です。

アセンブがDLLからロードされる場合、from パラメータが使用して、sn パラメーターは使用しないでください。アセンブルヴローバルアセ ンブルヤシュ(GAC) からロードされる場合、sn ハラメータを使用して from ハラメーターは使用しないでください。

最初のプラメーターの前に疑問符(?)を挿入し、パラメーター同士は七ミコロンで分離する必要が別ます。パラメーター名へ値を受け渡すに は、統合符号(=)を使用しまず以下の例を参照くたさい。

名前空間宣言の例

XSLT において、システムクラス System. Environment を特定する名前空間宣言の例を以下に示します:

xmlns:myns="clitype:System.Environment"

XSLT において、ロードするグラブを Trade.Forward.Scrip として特定する名前空間宣言の例を以下に示します:

xmlns:myns="clitype:Trade.Forward.Scrip?asm=forward;version=10.6.2.1"

XQuery において、システムグラス MyManagedDLL.testClass を特定する名前空間宣言の例を以下に示します。2つのケースが考 えられます:

1. アセンブ」がGAC からロードされた場合:

declare namespace cs="clitype:MyManagedDLL.testClass?asm=MyManagedDLL;

ver=1.2.3.4;loc=neutral;sn=b9f091b72dccfba8";

2. アセンブがDLL からロードされた場合(完全参照と一部の参照):

declare namespace cs="clitype:MyManagedDLL.testClass?from=file:///C:/Altova
Projects/extFunctions/MyManagedDLL.dll;

declare namespace cs="clitype:MyManagedDLL.testClass?from=MyManagedDLL.dll;

XSLT の例

システムクラス System. Math 内の関数を呼び出すための完全な XSLT の例を以下に示します:

math 要素にある名前空間宣言により、math: プレフィックスとclitype:System.Math URI が関連付けられます。URI の最初にある clitype:により、それ以降の記述がシステムケラスまけはロードされたケラスを特定するものであることが示されます。XPath 条件式にある math: プレフィックスにより、拡張関数がURI(そしてケラス)System.Math に関連付けられます。拡張関数により、System.Math ケラス内のメットが特定され、必要な場所に引数が与えられます。

XQuery の例

上のXSLT (式する例と同様のXQuery 例を以下に示します:

```
<math xmlns:math="clitype:System.Math">
    {math:Sqrt(9)}
</math>
```

上のXSLT と同様に、名前空間宣言により、NET クラス(この場合はシステムクラス)が特定されます。XQuery 式により呼び出されるメンドが特定され、引数が与えられます。

12.2.2.2.1 .NET コンストラクター

拡張関数を使用することで、.NET コンストラクターを呼び出すことができます。new() により全てのコンストラクターを呼び出すことができます。ケラス内に2つ以上のコンストラクターがある場合、与えられた引数の数が最もマッチするコンストラクターが選択されます。与えられた引数に対してマッチするコンストラクターが見つかられ場合、"No constructor found!" エラーが返されます。

XPath/XQuery データ型を返すコンストラクター

.NET コンストラクター呼び出しの結果が XPath/XQuery データ型へ黙示的に変換することができる場合、.NET 拡張関数から XPath/XQuery データ型のシーケンスが返されます。

.NET オブジェクトを返すコンストラクター

.NET コンストラクター呼び出しの結果がXPath/XQuery データ型へ適切に変換できない場合、値を返すクラス名でランプした.NET オブ ジェケトがコンストラクターにより作成されます。例えば、System.DateTime クラスのコンストラクターが、System.DateTime.new() により呼ばれた場合 System.DateTime 型を持ったオブジェクトが返されます。

返されたオブジェクトのレキシカルフォーマナトは、目的のXPath データ型と違っている場合があります。その場合、返された値を:(i)目的の XPath データ型のレキシカルフォーマナトへ変換し、(ii) 目的のXPath データ型へキャストする必要があります。

コンストラクターによい作成された.NET オブジェクトに対して3つのことを行うことができます:

変数内で使用することができます。

```
<xsl:variable name="currentdate" select="date:new(2008, 4, 29)"</pre>
xmlns:date="clitype:System.DateTime" />
```

• 拡張関数へ渡すことができます(インスタンスメソッドとインスタンスフィーバル・を参照ください): <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"</pre> xmlns:date="clitype:System.DateTime" />

• 文字列、数値、おけまboolean へ変換することができます:

```
<xsl:value-of select="xs:integer(date:get Month(date:new(2008, 4, 29)))"</pre>
 xmlns:date="clitype:System.DateTime" />
```

.NET: 静的メソッドと静的フィールド 12.2.2.2.2

メンバ名と引数を与えることで、静的メンドを直接呼び出すことができます。呼び出しこ使用される名前は、クラス内にある public static メノ・ドと完全に一致する必要ががます。関数の呼び出しに使用されたメノ・バ名と引数の数にマッチするものがラス内に複数ある場合、 与えられた引数が評価され、最もマッチするものが選択されます。マッチする結果が得られない場合、エラーが返されます。

火モ: .NET クラス内にあるフィーリドは 引数を持たない ツノドとしてみなされます。 プロ ティは get PropertyName() 構文によ り呼び出されます。

1つの引数とともにメンド(System.Math.Sin(arg))を呼び出す XSLT サンプルを以下に示します:

```
<xsl:value-of select="math:Sin(30)" xmlns:math="clitype:System.Math"/>
```

(引数なしのメンドとしてみなされる)フィール・(System.Double.MaxValue()を呼び出すXSLT サンプルを以下に示します:

```
<xsl:value-of select="double:MaxValue()" xmlns:double="clitype:System.Double"/>
```

(get PropertyName() 構文を使って)プロ ティ(System.String()) を呼び出す XSLT サンプルを以下に示します:

```
<xsl:value-of select="string:get_Length('my string')"
xmlns:string="clitype:System.String"/>

1つの別数ともプンド((System.Math.Sin(arg)) を呼び出すXQuery サンプルを以下に示します:

<sin xmlns:math="clitype:System.Math">
{ math:Sin(30) }
</sin>
```

12.2.2.2.3 .NET: インスタンスメソッドとインスタンスフィールド

インスタンスメンバロは、メンバ呼び出しの第一引数として、NET オブジェケトが渡されます。通常この、NET オブジェケトは、拡張関数(例えばエンストラクター呼び出し)まけはスタイルシート・プメーター/変数により作成されます。以下にXSLT の例を示します:

```
<xsl:stylesheet version="2.0"</pre>
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/xpath-functions">
   <xsl:output method="xml" omit-xml-declaration="yes"/>
   <xsl:template match="/">
      <xsl:variable name="releasedate"</pre>
         select="date:new(2008, 4, 29)"
         xmlns:date="clitype:System.DateTime"/>
      <doc>
         <date>
            <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"</pre>
               xmlns:date="clitype:System.DateTime"/>
         </date>
         <date>
            <xsl:value-of select="date:ToString($releasedate)"</pre>
               xmlns:date="clitype:System.DateTime"/>
         </date>
      </doc>
   </xsl:template>
</xsl:stylesheet>
```

上の例では、System.DateTime コントラケー(new(2008, 4, 29)) が、System.DateTime 型の.NET オジンエケの作成に使用されます。このオジンエケーは、最初にreleasedate 変数の値とて、次にSystem.DateTime.ToString() メンドの引数として作成されます。System.DateTime.ToString() インスタンスメンドは、System.DateTime コンストラケーの(new(2008, 4, 29)) における引数として2度呼び出されます。これらインスタンスにおって releasedate 変数が.NET オジンエケを取得するのに使用されます。

インスタンスメン・ドとインスタンスフィールド

インスタンスソッドとインスタンスフィールドの違いは理論的がはのです。インスタンスメッドでは、NET オブジェケトが直接引数に渡され、インスタンスフィールドでは、ウメーターや変数が、NET オブジェケトそのものを含めることはできるものの)代わりに渡されます。例えば上の例では、releasedate 変数に、NET オブジェケトが含まれており、この変数がを番目のdate 要素エンストラクターこで ToString() の引数として渡されます。そのため、最初のdate 要素にある ToString() インスタンスがインスタンスメットドであるのご対し、2番目はインスタンスフィールドとしてみなされます。両方のインスタンスで求められる結果は等価です。

データ型: XPath/XQuery から.NET へ 12.2.2.2.4

.NET 拡張関数がXPath/XQuery 条件式内部で使用された場合、複数ある.NET メンドのうち、どれが呼び出されたのか決定する のは関数の引数に使用されるデータ型が重要になります。

.NET では、以下のルールが適用されます:

- クラス内に司名のメンドかをつ以上ある場合、呼び出しに使用された引数の数がマッチするメンドだけが、呼び出される関数の候 補に狭められます。
- XPath/XQuery の文字列、数値、boolean データ型は黙示的に対応する、NET データ型へ変換されます(以下のリストを参 照)。与えられたXPath/XQuery 型がとつ以上の.NET 型へ変換できる場合(例: xs:integer)、選択されたメンゲに て宣言されている.NET 型が使用されます。例えば、呼び出された.NET メンドがfx(fouble) で、与えられた XPath/XQuery データ型がxs:integer の場合、xs:integer が.NET の double データ型へ変換されます。

以下のテーブルに XPath/XQuery の文字列、数値、boolean 型から、NET データ型へ行われる黙示的な変換リストを示します。

xs:string	StringValue, string
xs:boolean	BooleanValue, bool
xs:integer	<pre>IntegerValue, decimal, long, integer, short, byte, double, float</pre>
xs:float	FloatValue, float, double
xs:double	DoubleValue, double
xs:decimal	DecimalValue, decimal, double, float

上のノストにあるXML スキーマ型(ならびにXPath やXQuery で使用されているデータ型)のサブタイプも、対応する祖先のサブタイプとし て.NET のデータ型へ変換されます。

場合によっては、与えられた情報から正し、NETメンケを選択することができない場合もあります。例えば、以下のような場合を考えてみま しむ

- 与えられた引数が0あるxs:untypedAtomic値で、mymethod(float)メンドへ渡されるのを意図している。
- しかし、そのクラスコは別のデータ型をとる mymethod (double) というメンドも存在する。
- メンド名が同じで、与えられた型 (xs:untypedAtomic) も float と double の両方に変換することができるか。 xs:untypedAtomic がfloat ではくdouble で変換される可能性もある。
- 結果として、意図したメンドは選択されず、予期しない動作結果が対公可能性がある。この問題を回避するコよ、意図したメ ソゲを使用するユーザー定義のメソゲを別の名前で新たご作成する必要が別ます。

上のノストでカドーされていない型(例: xs:date)は変換されずエラーとかります。

データ型:.NET から XPath/XQuery へ 12.2.2.2.5

.NET メンナにより値が返される際に値のデータ型が文字列、数値、ませまboolean 型の場合、対応するXPath/XQuery 型への変 換が行われます。例えば .NET のdecimal データ型はxsd:decimal へ変換されます。

.NET オブジェケトや文字列、数値、boolean 以外のデータ型が返された場合、最初に(例えばSystem.DateTime.ToString() といす).NET メンドを使用して.NET オブジェケトを文字列へ変換します。XPath/XQuery では、文字列を目的となる型のレキシカルフォーマナトへ変換し、目的の型への変換を(例えばcast as 式を使用することで)行うことができます。

12.2.2.3 XSLT に対する MSXSL スクリプト

<msxsl:script> 要素にはユーザー定義の関数や変数が含まれており、XSLT スタイルシート内のXPath 条件式内部がら呼び出しを行うことができます。<msxsl:script> はカップレベル要素で、<msxsl:stylesheet> ままは<msxsl:transform> の子要素である必要があます。

<msxsl:script> 要素はurn:schemas-microsoft-com:xslt 名前空間内に存在する必要がみます(以下を参照なける)。

スクレプト言語と名前空間

ブロック内で使用されるスクレプト言語は<msxsl:script>要素のlanguage 属性に指定され、XPath 条件式におお関数の呼び出して対して使用される名前空間はimplements-prefix 属性にい特定されます(以下を参照)。

```
<msxsl:script language="scripting-language" implements-prefix="user-namespace-prefix">
  function-1 or variable-1
    ...
  function-n or variable-n
</msxsl:script>
```

<msxsl:script> 要素はWindows Scripting Runtime を使すさいとなっている。お使いのコピューターについている。 語が北か<msxsl:script> 要素では使用することができません。MSXSL スクリプトを使用すること、NET Framework 2.0 以上のプラナンオームをインストールする必要があります。結果として<msxsl:script> 言語から、NET スクリプト言語を使用することができます。

HTML の<script> 要素におる language 属性と同じ値が language 属性では受理されます。 language 属性が指定されてしない場合、Microsoft JScript かデンカルとして想定されます。

implements-prefix 属性には名前空間スコープ内で宣言されたプレフトックスか与えられます。通常この名前空間は関数ライブラルかとめて予約されたユーザーの名前空間とかはす。<msxsl:script>要素内で定義された全ての関数ならびに変数は、implements-prefix 属性にて指定されたプレフトックスで特定される名前空間に収められます。XPath 条件式内部から関数が呼ばれる場合、完全修飾関数名が同じ名前空間内に関数として定義されていなければかません。

サンプル

<msxsl:script> 要素内で定義された関数を使用するXSLT スタイパシートの例を以下に示します:

```
' rounded to the nearest cent
    dim\ a\ as\ integer\ =\ 13
    Function AddMargin (WholesalePrice) as integer
     AddMargin = WholesalePrice * 1.2 + a
   End Function
  11>
  </msxsl:script>
  <xsl:template match="/">
    <html>
      <body>
        <p>
          <br/>
Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          <br/>
          <br/>b>Total Wholesale Price =
            $<xsl:value-of select="50"/>
        </p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

データ型

スクリプトブロックとのやルとりに使用される。デメーターの値はXPath データ型に限定されます。スクリプトブロック内にある関数にてやルとだれるデータや変数に、この制限はありません。

アセンブリ

msxs1:assembly 要素を使用することで、アセンブルスクレプト内部ヘインポートすることができます。アセンブルは名前やURI により特定されます。アセンブルのインポートは、コンパル時に行われます。以下に msxsl:assembly 要素の簡単な使用例を示します:

アセンブル名は、以下のような完全な名前でも::

"system.Math, Version=3.1.4500.1 Culture=neutral PublicKeyToken=a46b3f648229c514"

名前空間

msxsl:using 要素により名前空間の宣言を行うことができます。これにより、スクレル内において名前空間無しでアセンブクラスを使用することができ、タイピングの手間を軽減することができます。以下に msxsl:using 要素の簡単な使用例を示します:

[&]quot;myAssembly.Draw" のような短い名前でも指定できます。

</msxsl:script>

namespace 属性の値は名前空間の名前といます。

12.3 DB により生成された XML スキーマ内のデータ型

XML スキーマがデータベース(DB) から生成される場合、そのDB 固有のデータ型は、XML スキーマデータ型に変換されます。DB データ型のXML スキーマデータ型へのマピングは、下に挙げられるDB で使用されます。下のJストから使用するDB を選択してください。

- ADO
- MS Access
- MS SQL Server
- MySQL
- ODBC
- Oracle
- Sybase

12.3.1 ADO

XML スキーマがADO データベース(DB) から生成される場合、ADO DB データ型は下のテーブルの通りXML スキーマデータ型に変換されます。

ADO データ型	XML スキーマデータ型
adGUID	xs:ID
adChar	xs:string
adWChar	xs:string
adVarChar	xs:string
adWVarChar	xs:string
adLongVarChar	xs:string
adWLongVarChar	xs:string
adVarWChar	xs:string
adBoolean	xs:boolean
adSingle	xs:float
adDouble	xs:double
adNumeric	xs:decimal
adCurrency	xs:decimal
adDBTimeStamp	xs:dateTime
adDate	xs:date
adBinary	xs:base64Binary
adVarBinary	xs:base64Binary
adLongVarBinary	xs:base64Binary
adInteger	xs:Integer

adUnsignedInt	xs:unsignedInt
adSmallInt	xs:short
adUnsignedSmallInt	xs:unsignedShort
adBigInt	xs:long
adUnsignedBigInt	xs:unsignedLong
adTinyInt	xs:byte
adUnsignedTinyInt	xs:unsignedByte

12.3.2 MS Access

XML スキーマがMS Access データベース(DB) から生成される場合、MS Access DB データ型は下のテーブルの通りXML スキーマ データ型に変換されます。

MS Access データ型	XML スキーマデータ型
GUID	xs:ID
char	xs:string
varchar	xs:string
memo	xs:string
bit	xs:boolean
Number(single)	xs:float
Number(double)	xs:double
Decimal	xs:decimal
Currency	xs:decimal
Date/Time	xs:dateTime
Number(Long Integer)	xs:integer
Number(Integer)	xs:short
Number(Byte)	xs:byte
OLE Object	xs:base64Binary

12.3.3 MS SQL Server

XML スキーマがMS SQL Server データベース(DB) から生成される場合、MS SQL Server DB データ型は下のテーブルの通り XML スキーマデータ型に変換されます。

MS SQL Server データ型	XML スキーマデータ型
uniqueidentifier	xs:ID

char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string
ntext	xs:string
sysname	xs:string
bit	xs:boolean
real	xs:float
float	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
binary	xs:base64Binary
varbinary	xs:base64Binary
image	xs:base64Binary
integer	xs:integer
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

12.3.4 MySQL

XML スキーマがMySQL データベース(DB) から生成される場合、MySQL DB データ型は下のテーブルの通りXML スキーマデータ型に変換されます。

MySQL データ型	XML スキーマデータ型
char	xs:string
varchar	xs:string
text	xs:string
tinytext	xs:string
mediumtext	xs:string
longtext	xs:string

tinyint(1)	xs:boolean
float	xs:float
double	xs:double
decimal	xs:decimal
datetime	xs:dateTime
blob	xs:base64Binary
tinyblob	xs:base64Binary
mediumblob	xs:base64Binary
longblob	xs:base64Binary
smallint	xs:short
bigint	xs:long
tinyint	xs:byte

12.3.5 **ODBC**

XML スキーマがODBC データベース(DB) から生成される場合、ODBC DB データ型は下のテーブルの通りXML スキーマデータ型に 変換されます。

ODBC データ型	XML スキーマデータ型
SQL_GUID	xs:ID
SQL_CHAR	xs:string
SQL_VARCHAR	xs:string
SQL_LONGVARCHAR	xs:string
SQL_BIT	xs:boolean
SQL_REAL	xs:float
SQL_DOUBLE	xs:double
SQL_DECIMAL	xs:decimal
SQL_TIMESTAMP	xs:dateTime
SQL_DATE	xs:date
SQL_BINARY	xs:base64Binary
SQL_VARBINARY	xs:base64Binary
SQL_LONGVARBINARY	xs:base64Binary
SQL_INTEGER	xs:integer
SQL_SMALLINT	xs:short
SQL_BIGINT	xs:long

SQL_TINYINT	xs:byte
-------------	---------

12.3.6 Oracle

XML スキーマがOracle データベース(DB) から生成される場合、Oracle DB データ型は下のテーブルの通りXML スキーマデータ型に変換されます。

Oracle データ型	XML スキーマデータ型
ROWID	xs:ID
CHAR	xs:string
NCHAR	xs:string
VARCHAR2	xs:string
NVARCHAR2	xs:string
CLOB	xs:string
NCLOB	xs:string
NUMBER (with check constraint applied)*	xs:boolean
NUMBER	xs:decimal
FLOAT	xs:double
DATE	xs:dateTime
INTERVAL YEAR TO MONTH	xs:gYearMonth
BLOB	xs:base64Binary

* データ型 NUMBER の列にチェック制約が適用されている場合、またチェック制約が0 まけま1 の値をチェックする場合、この列の NUMBER データ型は、xs:boolean のXML スキーマデータ型に変換されます。このメカニズムは、生成された XML スキーマ 内にxs:boolean データ型を生成する際に役に立ちます。

12.3.7 Sybase

XML スキーマがSybase データベース(DB) から生成される場合、Sybase DB データ型は下のテーブルの通りXML スキーマデータ型 に変換されます。

Sybase データ型	XML スキーマデータ型
char	xs:string
nchar	xs:string
varchar	xs:string
nvarchar	xs:string
text	xs:string

sysname-varchar(30)	xs:string
bit	xs:boolean
real	xs:float
float	xs:float
double	xs:double
decimal	xs:decimal
money	xs:decimal
smallmoney	xs:decimal
datetime	xs:dateTime
smalldatetime	xs:dateTime
timestamp	xs:dateTime
binary<=255	xs:base64Binary
varbinary<=255	xs:base64Binary
image	xs:base64Binary
integer	xs:integer
smallint	xs:short
tinyint	xs:byte

594 付録 技術データ

12.4 技術データ

このセグションは、ソフトウェアの技術面に関する役に立つ背景情報を含んでいます。以下のよう、整理されています:
OS とメモリ要件Altova XML バリデーターAltova XSLT とXQuery エンジ Unicode のサポート インターネナの使用

12.4.1 OS とメモリ要件

オペレーティングシステム

Altova ソフトウェアアプリケーションは 以下のプラナフォームでご使用いただます:

プラナフォーム更新済みのW indows 7 SP1、W indows 8、W indows 10プラナフォーム更新済みのW indows Server 2008 R2 SP1 ませは以降 メモリ

ソナウェアがC++ で書かれているため、Java Runtime Environment をダウルードする必要はなく、Java ベースのアプリケーションに 比べ、通常少ないソモル必要とします。しかしながら、各ドキュメントは完全に解析するため、また、ビューと編集の速度を向上するためにメモ リにダウルードされます。メモルの要件は、ドキュメントのサイズを増やします。

メモノ要件は、制限のない、元に戻す」履歴により影響を受けます。 大きなドキュメントの大きなセグションの切り取り、 貼り付け操作を繰り返し行うと、 使用できるメモルすぐに消費されます。

12.4.2 Altova XML バリデーター

XML ドキュメトを開くと、アプケーションは、内蔵のXML バリデーターを使用して、指定されている場合、スキーマニ対して整形式をチェックし、ツノーとインフォセナを作成します。 XML バリデーターは、ドキュメントを編集する際にインテルジェントな編集ペルプを提供し、発生する検証エラーを表示するがは使用されます。

内蔵のXML バリデーターは W 3C のXML スキーマ1.0 と1.1 仕様の最終勧告を実装しています。. New developments recommended by the W 3C XML スキーマ作業グループにお働告される新しい項目は、XML バリデーターに継続的に組み込まれるため、Altova 製品は最高水準の開発環境を届けることができます。

12.4.3 Altova XSLT と XQuery エンジン

Altova 製品は、Altova XSLT 1.0、2.0、および3.0 エンジンとAltova XQuery 1.0 と3.1 エンジンを使用しています。各エンジンのオータのドキュメートと実装に固有の振る舞いに関しては、製品で使用されるエンジンの各ドキュメートの付属書(エンジン情報)で確認することができます。

メモ Altova MapForce は、XSLT 1.0、2.0 および XQuery 1.0 エンジンを使用したコードを生成します。

12.4.4 Unicode のサポート

Altova XML 製品は、Unicode を完全にサポートします。XML ドキュメートを編集するコよ ドキュメート内で使用されている Unicode 文字をサポート するフォント か必要です。

フォントの多くは、Unicode 範囲全体の特定のサブセトを含む場合がが、このため、通常は対応する表記システムをターゲトとします。 テキストの一部が、文字化けして表示された場合、理由としては、選択されたフォントが必要とする字形を含まない場合があれられます。です 付録 技術データ 595

から、特に、異なる言語、ませま、異なる言語システムのXMLドキュメトを編集する場合、範囲全体をカドーするフォントを使用することが 役にするます。典型的なUnicode フォントは、Windows PC のArial Unicode MS で確認することができます。

アプケーションフォルダーの/Examples フォルダー内で、異なる言語システムで表記された次の文章を含むUnicodeUTF-8.html とらXHTML ファイルを確認してくたさい

When the world wants to talk, it speaks Unicode Wenn die Welt miteinander spricht, spricht sie Unicode 世界的に話すなら、Unicode です。) XHTML ファイルを開くと、Unicode の可能性を確認することがき、使用中のPC の使用することのできるフォントによりサポートされている表記システムが表示されます。

12.4.5 インターネットの使用

Altova アプリケーションは、次の状況でインターネット 接続を開始します:

- 登録ダイアログ(「ヘルプ | ソフトウェアのライセンス認証」)内の「評価キーコードをリクエスト」をクリックした場合、登録ダイアロ グボックス内の3つのフィールドが通常のhttp (ポート 80)接続を使用し、サーバーは送され、無料の評価キーが顧客に通常の SMTP 電子メールを使用して送り返されます。
- Altova 製品の一部では、インターネナからファイルを開くことができます(「ファイル | 開く | URL に切り替える」)。この場合、ドキュメントは、次のプロトコルダンドと接続の1つを使用して取得されます:HTTP(通常、ポート 80)、FTP(通常、ポート 20/21)、HTTPS(通常、ポート 443)。HTTP サーバーをポート 8080 で作動することもできます(URL ダイアログ内で、サーバー名とコロンの後にポートを指定します)。
- XML スキーマ、まけよ、DTD を参照する XML ドキュメト、と、URL により指定されているドキュメトを開くと参照されている スキーマドキュメトは、HTTP 接続(ポート 80) まけま URL により指定されている他のプロトコール上のポイト 2 参照) により 抽出されます。 XML ファイルが検証されている場合、スキーマドキュメントも抽出されます(オプションダイアログのファイルタブ内の(「ツール | オプション」))。アプリケーションに命令している場合、ドキュメントが開かれると検証が自動的に行われる場合もあります。
- W SDL とSOAP を使用する Altova アプケーションでは、W eb サービスを使用する接続は、W SDL ドキュメイにいた 義されています。
- XMLSpy 内で、「電子メールで送信」コマイを使用する場合、(「ファイル | 電子メールで送信」) 現在選択されている範囲、ませま、ファイルは、ユーザーのマシンにインストールされている MAPI コンプライア・ト、電子メールプログラムにより送信されます。
- ソナウェアの荒いセス認証 とLiveUpdate の一部として、Altova ソナウェア使用許諾書内で更に詳し、説明を確認することができます。

596 付録 ライセンス情報

12.5 ライセンス情報

このセグションコお以下の内容が含まれています:

- ソストウェアの配布 (関する情報)
- ソフトウェアのアクティベーションとライセンスの計測
- ソナウェアの使用に関する使用許諾契約書

本製品を使用する前に、上記の情報をよくお読みください。ソフトウェアのインストール時に上記のすべての条件に同意したとみなされ、お客様は上記の条件に拘束されることを同意したとみなされます。

Altova ライセンスの内容を確認するコよ Altova Web サイト のAltova法的な情報のページに移動してください。

12.5.1 電子的なソフトウェアの配布

この製品は電子的なソストウェアの配布により利用することが可能で、この配布方法により、以下のユニーグなメリメトが別ます:

- 購入を決定する前に、無料でノストウェアを試用することができます。(Note: Altova Mobile Together Designer に対して ライセンスを無料で割り当てることができます)。
- Once ソフトウェアの購入を決定した際には、Altova Webサイトにて注文を行います。すぐにライセンス登録された製品の使用を開始することができます。
- オンライノこで注文を行うと、常に最新のノフトウェアをご利用いけがます。
- 製品パケージコは包括的なヘルプンステムが画面上に表示されます。最新バージョンのユーザーマニュアルは
 https://www.altova.com/ja/上におり、(i) HTML フォーマナによる閲覧、ならびに(ii) PDF フォーマナのダウンロードと
 印刷に対応しております。

30日間の評価期間

この製品をダウンロードした後は、最大で30日の間無料で製品の評価を行うことができます。20日間を超えたは引から、製品からイセンス登録されていないことがノオ・ウェアにより表示されます。このメッセージはアプリケーションが起動されるたむに表示され、30日間を超えてプログラムを使用するには、キーコードを含むライセンスファイルから提供される製品のライセンスを購入します。ライセンスファイルを製品のノオ・ウェアアクティベーションダイアログにアップロードして、製品をアンロックします。

https://shop.altova.com/ でライセンスを購入することができます

組織内でノフトウェアの評価を行う

評価版のノストウェアを組織内のネ・トワークにて配布したい場合、まけはインターネットに接続されていないコンピューターにてノストウェアを使用する場合、どのような状態でも改変さていていないことを条件に、セットアッププログラムさけの配布を行うことが可能です。ソフトウェアインストーラーヘアクセスした人は、例外なく30日間の評価ライセンスキーコードをリクエストして、試用期間が経過した後は、製品を使い続けるためにライセンスの購入を行う必要があります。

12.5.2 ソフトウェアのアクティベーションとライセンスの計測

Altova のノストウェアアクティベーションの一部として、ソストウェアにより内部ネナワークませはインターネナトへの接続を行い、インストール時、登録時、Altova により使用されるライセンスサードーの更新やライセンスの正当性を検証することで、ソストウェアの不正な使用を妨ぎ、顧客サービスを向上するため、ライセンスに関する情報を送信することがあります。アクティベーションにより、オペレーティングシステムやIP アドレス

付録 ライセンス情報 597

日付/時刻、ソフトウェアのバージョン、コンピュータの名前などのライセンスに関する情報が、お使いのコンピューターとAltova ライセンスサーバー間にてや、取りされます。

お使いのAltova 製品にはライセンス計測モジュールが内蔵されており、エイューザー使用許諾契約書の意図しない違反を防ぎます。お使いの製品はシングルユーザーおけまマルチューザーとしてインストールされており、ライセンス計測モジュールにより、ライセンスされている数を超えたユーザーが同時に製品を使用することが無いことが保証されます。

このライセンス計測技術により、ローカルエリア接続(LAN)において、別々のエレピューター間で動作しているアプリケーションインスタンス間の通信が行われます。

シングルライセンス

ライセンス計測プロセスの一部としてアプケーションが起動すると、ソフトウェアにお短いデータグラムがブロードキャストにお送信され、同一のネトワークセグメントにある他のエンピューター(こてプログラムが動作していないかのチェックが行われます。 応答が無い場合は、アプケーションの他インスタンスから送信される信号に応えるため、ポートが開かれます。

マルチューザーライセンス

同一のLAN 内にて2つ以上のアプケーションインスタンスが使用された場合、スタートアプ時に、これらインスタンス間において通信が行われます。これらのインスタンス間にてキーコードのやりとめでわれ、購入された数のライセンスを超えてインスタンスが起動しないように保証することができます。このようなライセンス計測システムはUNIX やデータベース開発ソールにて広く使用されているもので、Altova ユーザーはリーズナブルな価格にて同時使用マルチユーザーライセンスを購入することができます。

弊社はアプケーションのデザインも行っており、少数の小さなネトワーク。かトを送信することで、ネトワークコオする負荷を最小限に抑えております。Altova により使用される2799番 TCP/IP ポートはIANA により公式登録されており、詳細は(IANA Web サイト (http://www.iana.org/) を参照ください、弊社のライセンス計測モジューリも既にテストされたものです。

ファイヤーウォールを使用している場合、2799番ポートにて Altova 製品が動作しているコンピューター同士が通信しているのに気づかれるかも知れません。その他の手段によりテイセンス使用許諾書の内容が守られることを保証できる限り、組織間の異なるグループにおいてこのようなトラフィックをブロックすることは勿論可能です。

証明書は関するメモ

Altova アプケーションはHTTPS を介して Altova ライセンスサーバー(link.altova.com) 「通信します。この通信のなめに Altova は登録済みのSSL 証明書を使用します。(例えば、社内 IT 部署まりは外部エージェンシーによりこの証明書が置き換えられている場合、使用中の Altova アプケーションは接続が安全ではいとを警告します。 Altova アプケーションを開始するから計算の証明書を使用することができますが、自己責任で行ってくたさい。 安全ではない 接続の警告メッセージが表示されると、証明書の発行元を確認して(Altova 証明書の代替証明書の使用の継続まりは停止を決定することができる社内 IT チームと相談してください。

(例えば、クライア・トマンへの、まけは、クライア・トマンへの通信を監視するける) 自身の証明書の使用が必要な場合 Altova の無料管理ノト・ウェアである Altova LicenseServer を使用中のネ・トワーグにインストールすることが奨励されます。このセトアップでは、Altova LicenseServer は Altova との通信のために Altova 証明書の使用を許可しつングライア・トマン・か所属機関の証明書の使用を継続することができます。

12.5.3 エンドユーザー使用許諾契約書

- Altova エイユーザー使用許諾契約書: http://www.altova.com/ja/legal/eula
- Altova プラインシーポシー: http://www.altova.com/ja/privacy

598 インデックス

インデックス

(

(テーブルの)セル, 左右に分割,466上下に分割,466(テーブルの)行, HTML 出力内の展開/折りたたみ,137(テーブルの)列, HTML 出力の非表示,137

.docx (Enterprise Edition のみ), 15, 29 .NET 拡張関数,

XPath 条件式内, 380, 391
XSLT と XQuery, 579
インスタンスメソッド、インスタンスフィールド, 583
コンストラクター, 581
サポート、Authentic View 内, 380, 391
データ型変換、.NET から XPath/XQuery へ, 584
データ型変換、XPath/XQuery から .NET へ, 584
概要, 579
静的メソッド、静的フィールド, 582

NET 内のXSLT と XQuery のための拡張機能, NET 拡張関数を参照する, 579

A

Altova XML パーサー, (について, 594 Altova ウェブサイト, 489 Altova エンジン, Altova 製品内で, 594 Altova ソフトウェアの注文, 486 Altova 拡張関数, チャート関数 (チャート関数を参照), 499 AltovaXML, とFOP, 370

ASP.NET アプリケーション, 356

ASPX web アプリケーション, 356 Authentic Browser, 18 Authentic Desktop, 18 Authentic View, Altova 製品内の, 18

C

CALS/HTML テーブル, 138, 467 CDATA セクション, 108 Core Catalog.xml, 103 CSS スタイル, クイックスタートチュートリアル内, 68 スタイルも参照, 44 モジュール SPS 内, 193 CSS スタイルシート, スタイルも参照, 305 に適用されたメディア, 305 外部スタイルシート, 305 外部の優先順序のインポート, 305 CSS ファイル, デザイン 概要 サイドバー内の管理, 33 CustomCatalog.xml, 103

D

DB パラメーター, 作成と編集, 432 DPI, 424 DTD, SPS ソースとして, 178 パースされていないエントリの宣言, 322 DTD とカタログ, 102

E

Excel テーブル コンテンツ, デザインにコピーして張り付ける, 112 インデックス 599

F

FO プロセッサー (Enterprise Edition), セットアップ, 21 FO 変換, 370

G

GUI.

ドキュメントビュー内, 28 メイン ウィンドウ の, 28 詳細, 27 複数の ドキュメント内, 28

H

HTML インポート、349 HTML テーブルの、353 HTML リストの、353 SPS からファイルの作成、355 SPS デザイン、351 スキーマ 構造、351 新規の SPS の作成、349 HTML から XML への変換、349 HTML テーブル、138、467 HTML ページ コンテンツ、 デザインにコピーして張り付ける、112 HTML 出力、97 とイメージサポート、151

Ī

IE 9.

Internet Explorer 互換性を参照, 95 Internet Explorer 互換性, 95

J

Java と .NET 関数 (Enterprise Edition のみ),

自動計算内, 228

Java 拡張関数,

XPath 条件式, 380, 391

XPath 条件式 ダイアログの編集, 380, 391

XPath 条件式内, 380, 391

XSLT & XQuery, 571

インスタンスメソッド、インスタンスフィールド, 577

コンストラクター, 576

サポート、Authentic View 内, 380, 391

データ型変換、Java から XPath/XQuery へ, 579

データ型変換、XPath/XQuery から Java へ, 578

ユーザー定義の JAR ファイル, 575

ユーザー定義のクラスファイル, 573

概要. 571

静的メソッド、静的フィールド,577

JavaScript,

スクリプトを参照, 344

JavaScript Editor, 345

JavaScript エディター, 344

デザインビュー内, 29

Line,

レイアウト コンテナー内, 453

M

Microsoft Office 2007 (Enterprise Edition $\mathcal{O}\mathcal{A}$), 15, 29

MobileTogether デザイン,

エクスポート, 421

MS Word ドキュメントコンテンツ,

デザインにコピーして張り付ける, 112

MSXSL スクリプト内の拡張関数, 585

msxsl:script, 585

0

Office Open XML (Enterprise Edition $\mathcal{O}\mathcal{A}$), 15, 29 OOXML (Enterprise Edition $\mathcal{O}\mathcal{A}$), 15, 29 OS.

Altova 製品のための, 594

600 インデックス

Otherwise 条件 分岐, 233

Р

PDF 出力 (Enterprise Edition), 97 とイメージサポート, 151 PXF ファイル, 359 デプロイ, 364 作成, 359 編集, 363 名前をつけて保存, 420

R

RaptorXML, 366 と FOP, 370 RaptorXML 内のカタログ, 103 RootCatalog.xml, 103 RTF 出力 (Enterprise Edition), 97 とイメージサポート, 151

S

SPS,

と StyleVision, 19 と XSLT スタイルシート, 19 開く, 410 概要, 19 再ロード, 410 閉じる, 410 SPS テーブル, 静的なテーブルも参照, 123 動的なテーブルも参照, 123 SPS デザインの構造, 175 SPS デザイン概要, 93 SPS のためのソース ファイル, 92 SPS ファイル 構造, 175 SPS 内の XPath バージョン, 94

SPS 内の XSLT バージョン, 94

ユーザー マニュアル, 13

製品の機能,15

Style Vision,

Ł Authentic View (Enterprise Ł Professional Edition), 19

StyleVision Power Stylesheet, SPS を参照, 13 StyleVision について, 489 StyleVision に関する良くある質問, 489 StyleVision のセットアップ, 21 StyleVision のためのサポート, 489 StyleVision のための技術サポート, 489 StyleVisionBatch, 21, 366

T

Toolbar buttons, adding and removing, 396

U

Unicode のサポート, Altova 製品内で, 594 URI, パースされていないエントリの保持, 322

W

Windows,

Altova 製品のためのサポート, 594
Word 2007 (Enterprise Edition のみ), 15, 29
Word ドキュメントコンテンツ,
デザインにコピーして張り付ける, 112
WordML (Enterprise Edition のみ), 15, 29



XML, デザインに挿入, 121 XML スキーマ と DTD, SPS ソースとして, 178 XML データ, SPS デザイン内に挿入, 108 複数のソースからマージ, 186 XML テーブル (Enterprise と Professional Edition), 123 XML ドキュメントコンテンツ, XML ドキュメントコンテンツ, デザインビュー内のシンボル, 376

XML ドキュメントノード,

デザインビュー内のシンボル, 376

XML パーサー,

について, 594

XMLSpy, 18

XPath.

複数のドキュメント内でのノードの検索, 255

XPath 1.0,

と日付,341

XPath 2.0.

と日付,341

XPath ダイアログ.

デバッグ式, 384

式のテスト, 381

詳細, 380, 391

XPath フィルター、

グローバル テンプレート上, 203

XPath 演算子,

XPath ダイアログ内, 380, 391

XPath 関数,

XPath ダイアログ内, 380, 391

ユーザー定義, 327

XPath 条件式 ダイアログの編集,

XPath ダイアロ を参照, 391

XPath ダイアログを参照, 380

XQuery,

拡張関数, 571

XQuery プロセッサー,

Altova 製品内で, 594

XSLT.

デザインにコードフラグメントを挿入, 121

拡張関数. 571

XSLT インポート, 324

XSLT から SPS は. 324

XSLT から新規の, 324

XSLT スタイルシート プレビュー,

出力ビュー内, 29

XSLT テンプレート, 39

SPS ヘインポートする, 216

デザイン 概要 サイドバー内の管理, 33

XSLT テンプレートのインポート.

SPS ~, 216

XSLT と XQuery のためのJava 拡張関数,

Java 拡張関数を参照する, 571

XSLT と XQuery のための拡張関数, 571

XSLT のパースされていないエントリの URI 関数, 322

XSLT バージョン,

SPS のための設定, 402

XSLT プロセッサー.

Altova 製品内で, 594

XSLT 変換, 370

XSLT 要素,

デザインにコードとして挿入する, 120

XSLT/XQuery 内のスクリプト,

拡張関数で参照する,571

Z

イベントハンドラー,

関数を割り当てる.346

イメージ,

SPS 内の使用の概要. 149

URI の指定, 149

イメージも参照, 149

クイックスタートチュートリアル内, 63

サポートされている型, 151

サンプル ファイル, 153

デザインビュー内のシンボル,376

デザイン内に挿入のためのコマンド, 441

とパースされていないエンティティ URI, 149

パースされていないエントリを使用して検索,322

出力の表示の評価. 149

インターネットの使用,

Altova 製品内で, 595

インターフェイス,

GUI を参照, 27

ウィンドウ メニュー, 485

エンコード、

出力ファイルのための、482

エンコード コマンド, 424

エンティティ 参照,

と出力 エスケープ, 293

エンドユーザー使用許諾契約書, 596, 597

エントリ,

URI ホルダーとして使用, 322

パースされていない,322

カスタム化 Style Vision, 477

カスタム辞書.

SPS スペルチェックのための, 474

カタログ ファイル, 98

カタログと環境変数, 105

カタログのカスタム化, 104

カタログファイル, 102 カタログメカニズムの概要, 102 カタログ内で使用される環境変数, 103 キーコード,

Altova ソフトウェアのための, 486

キーボードショートカット, コマンドのカスタム化, 477

クイックスタートチュートリアル,

CSS スタイル, 68

testing Authentic View (Enterprise & Professional Edition), 90

XSLT スタイルシートの生成,90

イメージ, 63

クラス 属性,68

グローバル テンプレート, 85

コンテンツ プレースホルダ, 56

コンボボックス, 78

その他のコンテンツ,85

はじめに,51

リスト, 78

作成 新規の SPS ドキュメント, 52

自動計算,74

条件, 78

新規の SPS ドキュメントのセットアップ、52

水平線. 63

静的コンテンツ, 63

静的テキスト, 63

定義済みフォーマット, 68

動的コンテンツ, 56

必要な ファイル, 51

クラス 属性,

クイックスタートチュートリアル内, 68

グループ化, 238

サンプルによりグループ化 (Scores.sps), 242 サンプルによりグループ化(Persons.sps), 240

グローバル スタイル,

スタイルを参照,308

グローバル テンプレート, 202, 203 クイックスタートチュートリアル内, 85

その他のコンテンツへの効果, 111

グローバル型,

テンプレート内, 203

コピー コマンド, 427

コマンド.

カスタム化, 477

コマンド ラインのユーティリティ, 21

コマンドライン, 366 とパラメーター, 251 と計画されたタスク、374

コンテキストノード,

XPath ダイアログ内, 380, 391

コンテンツ.

デザイン内に挿入のためのコマンド, 437

コンテンツ プレースホルダ,

クイックスタートチュートリアル内, 56 コンテンツとしてノードを挿入. 108

コンテンツ 編集 プロシージャ, 107

コントロールから,

挿入のためのメニューコマンド, 438

コンポーネントダウンロードセンター, Altova ウェブサイトで, 489

コンボボックス, 157

クイックスタートチュートリアル内, 78

サポートオプション. 25

ショートカット.

キーボードのためのカスタム化, 477

シンプルな グローバル テンプレート, 203

スキーマ,

SPS ソースとして, 178

カタログを介したルックアップ, 104

ユーザー定義, 183

スキーマソース, 92, 404

SPS のための選択, 177

サイドバー ウィンドウ, 36

ソースの変更, 255

デザイン 概要 サイドバー内の管理, 33

と ルート 要素 (ドキュメント 要素), 177

の概要, 39

複数のSPS (Enterprise Edition), 177

複数のソース と XPath, 255

複数のソース とノードの検索, 255

スキーマソース ウィンドウ.

デザイン入力ヘルパーも参照, 31

スキーマツリーオプション, 482

スキーマとカタログ, 102

スキーマの追加,404

スクリーン上の ヘルプ.

のインデックス, 486

検索, 486

目次, 486

スクリプト.

JavaScript 関数, 346

SPS 内の使用, 344

デザイン ツリー内, 344

と JavaScript 関数, 344

の概要,39

スクリプト.

定義 JavaScript 関数, 345

スクロール ボタン.

メイン ウィンドウ内, 28

スタイル.

CSS ルールを合成, 317

SPS に CSS スタイルシートを割り当てる,305

SPS 内の定義 グローバル スタイル, 308

StyleVision 内で作業, 304

XML データから, 315

XPath 条件式, 315

XPath 条件式を使用して, 315

カスケードの順序, 304

サイドバー ウィンドウ. 44

スタイルの優先順位, 308

デザイン入力ヘルパーも参照, 31

と スタイル, 315

とプロパティグループ, 44

の優先順位,42

の用語,304

割り当てられた外部スタイルシートのためのメディア,305

定義,44

定義 ローカル スタイル, 310

複数を合成, 317

スタイル レポジトリ.

サイドバー ウィンドウ. 42

デザイン入力ヘルパーも参照, 31

と グローバル スタイル, 308

と外部CSS スタイルシート, 305

スタイルシート

CSS スタイルシートを参照, 305

XSLT スタイルシートを参照, 305

ステータスバー, 434

スペルチェック.

StyleVision 内, 473

スペルチェック オプション.

SPS のための, 474

セル,

SPS テーブル内の結合, 465

セルコンテンツの上下揃え,

SPS 内, 468

セルの結合.

SPS テーブル内, 465

その他のコンテンツ, 111

クイックスタートチュートリアル内、85

デザイン内に挿入のためのコマンド, 438

と グローバル テンプレート, 203

ソフトウェアの比較,

ダウンロードのための、489

ソフトウェアライセンス認証, 486

ソフトウェア製品ライセンス, 597

ダイアログのカスタマイズ.

StyleVision のためのカスタマイズ, 434

タグ.

展開と縮退,432

タグで囲むメニュー, 456

チェックボックス, 155

ツールバー, 395

GUI の配置, 395

カスタマイズ, 434

テーブル ツールバー, 397

フォーマット ツールバー. 397

移動, 27

再設定, 395

追加/移動アイコン, 395

標準 ツールバー, 402

表示のオンとオフ、395

表示のオンとオフの切り替え, 434

ツールバーとウィンドウの復元, 482

ツールバーを使用してデザイン要素を挿入する、

デザイン要素 ツールバーの挿入, 399

ツールメニュー, 473

データ-エントリデバイス、154

デザインビュー内のシンボル, 376

挿入のためのメニューコマンド、438

データベース.

編集のためのツールバー ボタン, 402

データベース (Enterprise と Professional Edition),

DB を参照 , 13

テーブル.

HTML ドキュメントからインポートされた, 353

navigating, 463

PDF 内のヘッダー と フッター, 133

SPS 内の削除, 463

Style Vision 内での境界の表示非表示, 466

から行を削除, 465

から列を削除, 465

セルコンテンツ, 463

セルコンテンツの上下揃え, 468

セルの結合, 465

フォーマット、133

ヘッダー と フッター, 463

ヘッダー とフッターの追加, 464

概要, 123

空の列の非表示, 137

行の挿入,464

データ-エントリデバイスの, 376

ハイパーリンクの、376 テーブル. 行の追加. 464 ブックマーク (アンカー)の, 376 行の展開/折りたたみ, 137 自動計算の,376 作成, 444 条件付きのテンプレートの, 376 条件付きの処理, 130 定義済みフォーマットの, 376 静的な テーブルの挿入. 463 デザインビュー内のテーブル, タグで囲まれた と 削除 テンプレート, 131 静的なテーブル, 124 代替の行のためのスタイル. 315 動的なテーブルの作成、126 デザインビュー内のマークアップタグ、29 編集 プロパティの. 467 デザインビュー内の入力へルパー, 列の挿入,464 表示のオンとオフの切り替え, 435 列の追加,464 デザインフラグメント、212 列を非表示にするボタン, 137 デザイン入力ヘルパー. 自動的に非表示. 31 テーブル (SPS). プロパティの編集,397 詳細.31 編集のためのツールバー ボタン, 397 非表示, 31 テーブル セルコンテンツの配置、 表示のオンとオフの切り替え, 435 SPS 内. 468 デザイン入力ヘルパー ウィンドウ. テーブル メニュー, 463 ドッキング, 31 フロート, 31 テーブルセルの分割, 左右, 466 デザイン要素, 399 上下, 466 デフォルトの ユーザー 辞書, SPS スペルチェックのための, 474 テキスト参照, 282 デザイン コマンドの保存, 415 テンプレート. デザイン ツリー. タグで囲まれた,456 サイドバー ウィンドウ, 39 テーブル 行と列の移動. 131 テーブル 行と列をタグで囲む, 131 と モジュール SPS, 193 デザイン テンプレートのためのフィルター, と一致するノードの変更, 172 スイッチオン とオフ,435 のツリー, 39 ビューのオンとオフの切り替え, 435 デザイン フィルター, 插入, 450 スイッチオン と オフ, 435 デザイン フラグメント. テンプレート XML ファイル (Enterprise と Professional Edition), 92 挿入, 453 定義, 22 デザイン 概要. テンプレート フィルター, 401 サイドバー ウィンドウ, 33 テンプレートー致 コマンドの編集, 117 デザイン 構造, 175 ドキュメント. デザインツリー, 開くと閉じる,28 デザイン入力ヘルパーも参照, 31 概要. 25 デザインビュー. ドキュメント スタイル, 225 SPS デザイン内のシンボル, 376 ドキュメント プロパティ, 225 と JavaScript エディター, 29 ドキュメント 要素 (ルート 要素参照), 177 マークアップの表示, 29 ドキュメントノード. 詳細, 29 定義, 22 デザインビュー内のシンボル. ドキュメントビュー, XML ドキュメントコンテンツの, 376 GUI 内, 28 XML ドキュメントノードの, 376 ドキュメント要素, イメージの、376 定義, 22

ドッキング.

デザイン入力ヘルパー ウィンドウ, 31

ノード,

として作成されているノードの変更, 172

ノードテンプレート,

と XPath フィルター, 209

と グローバル テンプレート, 209

ユーザー定義, 117

子テンプレートに変換, 209

操作, 209

ノードテンプレート上の XPath フィルター, 209

ノードテンプレート上のフィルター, 209

ノードのためのテンプレート.

ノードテンプレートを参照. 209

ノードの更新 (Enterprise と Professional Edition),

自動計算結果を使用して, 228

バーコード、159

パーサー.

Altova 製品に内蔵の, 594

ハイパーリンク, 158, 284

タグで囲まれた, 459

デザインビュー内のシンボル, 376

デザイン内に挿入のためのコマンド, 447

とパースされていないエントリ, 286

ハイパーリンクを使用して検索, 322

ブックマークにリンク. 286

移動 と削除, 286

外部リソースにリンク, 286

作成と編集, 286

バッチ ファイル,

と計画されたタスク,374

パラメーター, 251

SPS 内, 251

スキーマソースのため、255

デザイン 概要 サイドバー内の管理, 33

デザインフラグメントのための, 252

لط Authentic View, 251

とコマンドライン, 251

ユーザー定義 パラメーターの概要,39

一般的な説明, 251

作成と編集, 432

複数の ドキュメント内でのノードの検索, 255

パラメーター ダイアログの編集, 432

バリデーター.

Altova 製品内で, 594

ピクセル,

と 印刷メディアの長さ, 424

とスクリーンの解像度, 424

ビュー,

GUI レイアウト, 31

ファイル.

最後に開かれたファイルを開く、426

ファイル メニュー, 404

コマンドの終了, 426

ファイル | エンコード, 424

ファイル | デザインの保存, 415

ファイル | 印刷, 425

ファイル | 印刷プレビュー, 425

ファイル | 開く, 410

ファイル | 新規の, 404

ファイル | 生成された ファイルの保存, 421

ファイル | 閉じる. 410

ファイル | 名前をつけて保存, 420

ファイル 変更の通知,

モジュール SPS 内, 193

ファイルコマンド, 427

フィルター,

テンプレート選択して確認するための, 401

フォーマット.

ツールバー ボタン, 397

テーブルのための、133

テキスト プロパティ, 397

テキストの配置, 397

プロシージャの概要. 290

リスト, 397

数字フィールドの, 295

挿入時のノード, 110

定義済みの HTML フォーマット, 397

表示も参照, 290

フォーマット 設定.

入力 フォーマットのための定義, 471

フォーマット 番号.

自動番号, 279

フォーム-ベースの デザイン, 163, 404

ブックマーク, 158, 284

タグで囲まれた, 459

デザイン内に挿入のためのコマンド, 446

作成と編集, 284

削除, 284

ブックマーク (アンカー),

デザインビュー内のシンボル、376

フッター,

テーブル内, 133

テーブル内に追加, 464

プラットフォーム.

Altova 製品のための, 594

プレビューの問題, 21 フロート. デザイン入力ヘルパー ウィンドウs. 31 モジュール. プロジェクト オプション, 482 プロセッサー. ダウンロードのための, 489 プロパティ. SPS テーブルの. 397. 467 サイドバー ウィンドウ, 45 デザイン入力ヘルパーも参照,31 とプロパティグループ, 45 定義, 45 プロパティメニュー, 470 箇条書きと段落番号, 470 プロパティ入カヘルパー. イベントグループ, 346 ベース年度. 入力 フォーマット内, 295 ヘッダー. テーブル内, 133 テーブル内に追加, 464 ヘルプ. スクリーン上のヘルプ参照, 486 ヘルプ メニュー, 486 ボタン. 157 マークアップの非表示, 29 マークアップの表示, 29 メイン スキーマ, 202 メイン スキーマ (Enterprise Edition のみ), 36 メイン テンプレート, 202 定義, 22 メッセージサイドバー, 49 メニュー, カスタム化, 477 メニュー バー, 移動. 27 メニューの挿入, 437 箇条書きと段落番号, 444 挿入 | イメージ, 441 挿入 | コンテンツ, 437 挿入 | その他のコンテンツ, 438 挿入 | デザイン フラグメント, 453 挿入 | ハイパーリンク、447 挿入 | ブックマーク, 446 插入 | 自動計算, 439 插入 | 条件, 449 插入 | 水平線, 443 插入 | 段落, 440 リスト (静的 と動的), 444

插入 | 特殊な 段落 440 メモリ要件, 594 デザイン 概要 サイドバー内の管理, 33 モジュール SPS. SPS モジュールに追加する, 193 サンプル プロジェクト, 197 と CSS スタイル, 190, 193 とスキーマソース, 190, 193 と スクリプト, 190 と テンプレート XML ファイル, 190 とファイル 変更の通知, 193 と モジュールオブジェクト, 190 と 作業用 XML ファイル, 190 と名前空間宣言, 190 と作業, 193 概要, 189 作成. 193 追加する SPS モジュール, 193 優先順位の順序の効果, 193 有効化と無効化, 193 やり直しコマンド, 427 ユーザー インターフェイス. GUI を参照, 27 ユーザー マニュアル. スクリーン上のヘルプも参照, 486 ユーザー参照, 375 ユーザー定義 XML テキストブロック, 121 ユーザー定義 XPath 関数, 327 ユーザー定義 スキーマ, 183 ユーザー定義 テキストブロック, 454 ユーザー定義 テンプレート、117 タグで囲まれた, 457 插入. 451 ユーザー定義 要素, 120, 454, 462 ユーザー定義テキストブロック, 120 ユーザー定義要素, 120 ライセンス, 597 Altova ソフトウェアのための, 486 情報,596 ライセンス計測. Altova 製品にて, 596 ラジオ ボタン、157 リスト, 144, 458 HTML ドキュメントからインポートされた, 353 クイックスタートチュートリアル内, 78 タグで囲まれた,458

リスト プロパティ, 470

リンク,

ハイパーリンクを参照, 158, 284

ルート 要素, 36

ルート 要素 (ドキュメント 要素参照),

スキーマのために選択, 177 とスキーマソース, 177

レイアウト.

GUI 内のビュー, 31

レイアウト コンテナー, 163, 453

レイアウト コンテナー と 要素, 399

レイアウト ボックス, 453

レイアウト モジュール、

作成のためのステップ, 163

レイアウトのための青写真, 163

レイアウトボックス、166

ローカル スタイル,

スタイルを参照, 310

ローカル テンプレート, 202

一時的な 出力 ドキュメント, 21

一般的な使用方法,91

印刷 コマンド, 425

印刷プレビュー コマンド, 425

箇条書きと段落番号, 144, 146, 444, 470

タグで囲まれた,458

開く.

最後に開かれた ファイル, 426

開く(SPS) コマンド, 410

環境変数, 105

関数,

in XPath 内でユーザーにより定義されている, 327

機能.

StyleVision, 15

技術データ,594

境界.

SPS テーブルの, 466

型-ベースの テンプレート, 203

計画されたタスク,

Style Vision Batch コマンドの作成, 374

StyleVisionBatch バッチ ファイル, 374

検索, 49

正規表現の使用, 427

元に戻すコマンド, 427

行,

SPS テーブルに挿入, 464

SPS テーブルに追加, 464

SPS 内のテーブルから削除, 465

合成 スタイル, 317

最後に開かれた ファイル, 426

作業用 XML ファイル, 36, 92

と 出力ビュー, 29

印刷, 425

印刷プレビュー, 425

定義. 22

作成 新規の SPS ドキュメント.

クイックスタートチュートリアル内, 52

削除.

SPS 内のテーブル, 463

SPS 内のテーブルから行を削除, 465

SPS 内のテーブルから列を削除, 465

削除 コマンド, 427

使用方法,91

試用期間.

Altova ソフトウェア製品の試用, 596

次を検索コマンド, 427

自動化された 処理,366

自動計算, 228

Java と:NET 関数 (Enterprise Edition のみ), 228

クイックスタートチュートリアル内, 74

サンプル, 242

サンプル ファイル, 230

デザインビュー内のシンボル, 376

デザイン内に挿入のためのコマンド, 439

と出力 エスケープ. 293

と条件, 237

ノードを値で更新する,439

移動, 228

作成、編集、フォーマット, 228

使用方法, 228

日付結果のフォーマット, 341

自動的なノードの番号付け、279

自動的に非表示.

デザイン入力ヘルパーの機能, 31

自動番号, 279

辞書,

SPS スペルチェックのための, 474

終了 コマンド, 426

出力 エスケープ, 293

出力 エンコード, 424

出力 ドキュメントのエンコード, 424

出力 ドキュメントのスタイル, 225

出力 ドキュメントのプロパティ, 225

出カ-エスケープの無効化, 293

出力ビュー,

詳細. 29

出力ファイル,

出力ファイル,

生成,97

処理ユニットとしての型,

グローバル テンプレート内, 203

小数.

のフォーマット, 295

省略.

ユーザーマニュアル内で使用されている, 25

上下のテキスト.

テーブル セル内, 133 レイアウトボックス内, 166

条件.

クイックスタートチュートリアル内, 78

セットアップ、233

タグで囲まれた,460

デザイン内に挿入のためのコマンド, 449

と自動計算,237

編集, 236

条件付きのテンプレート, 449

デザインビュー内のシンボル, 376

条件を参照, 233

新規の SPS ドキュメントのセットアップ,

クイックスタートチュートリアル内, 52

新規の コマンド, 404

新規の ドキュメント テンプレート, 218

URL. 220

とデザイン 構造, 220

挿入, 219

新規機能,14

水平線.

クイックスタートチュートリアル内, 63 デザイン内に挿入のためのコマンド, 443

数字,

のフォーマット、295

数字フィールド,

フォーマットの, 295

整数,

のフォーマット, 295

正規表現.

検索と置換の条件式, 427

生成された ファイルの保存コマンド, 421

生成されたファイル, 97

製品の機能.

リスト, 15

静的コンテンツ.

クイックスタートチュートリアル内, 63

静的テキスト.

クイックスタートチュートリアル内, 63

と出力 エスケープ 293

静的な テーブル,

SPS 内に挿入, 397

挿入, 463

編集のためのツールバー ボタン, 397

静的な リスト、144、444

静的なテーブル, 123

SPS テーブルも参照してください。, 124

テーブルも参照, 133

切り取りコマンド, 427

全て選択コマンド, 427

挿入.

SPS テーブル内の行, 464

SPS テーブル内の列, 464

相互参照, 282

段落.

タグで囲まれた、458

デザイン内に挿入のためのコマンド, 440

値フォーマット, 295

置換, 49

正規表現の使用, 427

置換コマンド (Enterprise と Professional エディション),

427

著作権に関する情報,596

追加,

SPS 内のテーブルの行, 464

SPS 内のテーブルの列, 464

追加編集 プロシージャ, 321

定義済みの フォーマット 設定,

入力 フォーマットのための, 471

定義済みフォーマット,

デザインビュー内のシンボル,376

デザイン内に挿入のためのコマンド, 440

ノードの挿入, 110

定義済みフォーマットの割り当て,

クイックスタートチュートリアル内, 68

貼り付けコマンド, 427

動的コンテンツ,

クイックスタートチュートリアル内, 56

動的な テーブル.

編集のためのツールバー ボタン, 397

動的な リスト、144、146、444

動的なテーブル, 123

SPS テーブルも参照, 126

テーブルも参照, 133

と グローバル テンプレート, 126

ネスとされた動的なテーブル, 126

ヘッダー と フッター, 126

動的なテーブル、123

追加された/挿入された行との差異, 126

特殊な 段落.

タグで囲まれた,458

デザイン内に挿入のためのコマンド, 440

日付.

SPS 内での使用, 341

XPath 2.0 を使用したデータ操作のサンプル, 341

のフォーマット, 295

フォーマットの, 341

入力 フィールド, 155

入力 フォーマット.

定義 フォーマット 設定のための, 471

日付の, 341

背景情報, 594

配布,

Altova ソフトウェア製品, 596

Altova ソフトウェア製品の配布, 596

非表示,

デザイン入力ヘルパーの機能, 31

表示.

フォーマット、フォーマットも参照, 290

プロシージャの概要, 290

表示メニュー, 434

評価キー

Altova ソフトウェアのための, 486

評価機関.

Altova ソフトウェア製品, 596

付録, 491

複合型の グローバル テンプレート, 203

複数の 出力-ドキュメント, 218

と 出力 プレビュー, 223

ファイルの場所, 223

リンクする、220

複数のドキュメントの出力, 454

複数の入力 フィールド, 155

文字 参照,

と出力 エスケープ, 293

並べ替え, 245

グループとグループ内, 238, 240, 242

サンプル ファイル, 247

並べ替え メカニズム, 245

並べ替えキー, 245

並べ替えキー, 245

閉じる (SPS) コマンド, 410

変更コマンド, 172

変数, 251, 255

変数 テンプレート, 209

タグで囲まれた、457

挿入, 452

編集メニュー, 427

法的な情報,596

名前が付けられたテンプレート, 202

名前空間.

SPS に追加, 36, 97

SPS へ追加, 92, 177

SPS 内の. 36

の概要, 39

目次, 258

サンプル, シンプルな, 272

サンプル、階層ならびに連続的な, 275

メニュー コマンド, 453

含まれるアイテムのためのマーク付け, 260

使用方法の概要, 258

目次 を参照, 258

目次 アイテム,

フォーマット, 271

構築, 271

目次 テンプレート.

フォーマット, 271

レベル参照, 269

構成, 269

作成と編集, 267

参照レベル, 269

目次 ブックマーク, 260

タグで囲まれた,461

とレベル, 265

のためのウィザード, 265

作成, 265

目次 レベル、260、262

タグで囲まれた,461

目次参照, 270

目次参照を参照, 270

優先順位.

のスタイル, 42

用語.

StyleVision 内で使用されている, 22

要素.

ユーザー定義, 120

要素 テンプレート,

ユーザー定義, 120

列,

SPS テーブルに追加, 464

SPS テーブル内に挿入, 464

SPS 内のテーブルから削除, 465

連続した マークアップ, 29