

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

はじめに 1 13 1.1 製品機能......15 1.2 Altova 製品における Authentic View......18 1.3 1.4 1.5 1.6 1.7

2 ユーザーインターフェイス

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	<u>`</u>	31
	2.2.1	デザイン概要	33
	2.2.2	スキーマツリー	36
	2.2.3	デザインツリー	39
	2.2.4	スタイルレポジトリ	42
	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

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	63
3.4	コンテンツのフォーマット	68

27

4 使用方法の概要

4.1	SPS &	とソース	
4.2	デザイ	ンの作成	93
4.3	XSLT	とXPath のバージョン	
4.4	Interne	st 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	StyleVision 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

5.1	テキス	トとして XML コンテンツを挿入	
	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	110
	5.1.2	その他のコンテンツ	111
5.2	MS Wo	rd コンテンツの挿入	112
5.3	MS Exc	cel コンテンツの挿入	114
5.4	ューザ	一定義テンプレート	117
5.5	ューザ	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	<i>۱</i> ۷	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	イメージ		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	.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	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	崔	172
6	SPS 4	黄造	175

6.1	スキー	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	Dソースから XML データのマージ	
6.3	モジュ	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	ピートと デザインフラグメント	202
	6.4.1	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 7	シプレート	216
6.6	複数の	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

7.1	自動計	算	228
	7.1.1	自動計算の編集と移動	228
	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	233
	7.2.2	条件の編集	
	7.2.3	条件と自動計算	237
7.3	グルー	プ化	
	7.3.1	サンプル: group-by(Persons.sps)	240
	7.3.2	サンプル: group-by(Scores.sps)	
7.4	並べ替	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	
	7.5.1	ユーザー宣言パラメーター	
	7.5.2	デザインフラグメントに対するパラメーター	
	7.5.3	ソースのための SPS パラメータ	
	7.5.4	変数	
7.6	目次、	参照、ブックマーク	
	7.6.1	目次に含まれるブックマークアイテム	

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

8.1	定義済	fみのフォーマット	291
8.2	出力の)エスケープ	293
8.3	値フォー	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 ス	くタイルを使用する	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	

9 追加機能

 9.2 XSLT、XSL-FO または FO ファイルからの新規作成 9.3 ユーザー定義 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 9.3.3 XPath 関数内のパラメーター 	322
 9.3 ユーザー定義 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 9.3.3 XPath 関数内のパラメーター 	324
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 9.3.3 XPath 関数内のパラメーター	327
9.3.2 ノードを検索するための関数の再利用 9.3.3 XPath 関数内のパラメーター	329
9.3.3 XPath 関数内のパラメーター	331
	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	345
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		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	マイル: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

コマンド	ライン インターフェイス	367
10.1.1	StyleVision	367
10.1.2	StyleVision Server	368
RaptorX	ML の使用	370
10.2.1	PDF 出力	370
FlowFor	ce Server を使用した自動化	372
自動処理	里の方法	374
	コマンド 10.1.1 10.1.2 RaptorX 10.2.1 FlowFor 自動処理	コマンドライン インターフェイス 10.1.1 StyleVision 10.1.2 StyleVision Server RaptorXML の使用 10.2.1 PDF 出力 FlowForce Server を使用した自動化 自動処理の方法

11 メニューコマンドとレファレンス

11.1	デザイン	・ビューのシンボル	. 376
11.2	XPath ī	ちの編集ダイアログ	. 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	ッール	ĩ —	. 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
	1134	デザインフィルター	401
	11.0.1		

375

	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	<u>-</u>	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	<u>-</u>	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	444
	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>	メニュー	473
	11.11.1	スペルチェック	473

474
477
482
482
485
486
486
486
489

12 付録

12.1	XSLT お	らよび 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-	-9	. 594
	12.4.1	OS とメモリ要件	. 594
	12.4.2	Altova XML バリデーター	. 594
	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	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

1 はじめに

Altova StyleVision 2021 Basic Edition は StyleVision Power Stylesheet の編集を視覚的に行うせのアプケーション で。StyleVision®は、プラホフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10 と プラホフォーム更新済み のWindows Server 2008 R2 SP1 おけお以降で作動します。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 おけお以降のアプリ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WYSIWYG ビューの Authentic View にて管理します。Authentic View は、Altova XMLSpy、Altova StyleVision、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ドキュメントをベースはた電子フォームを簡単に作成することができます。
- SPS デザインをベースはたXSLT スタイルシートを生成します(XSLT 1.0、XSLT 2.0、とXSLT 3.0 かりポートされます)。XSLT スタイルシートはStyleVision 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。
- StyleVision 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

1.1 新規機能: バージョン 2021

バージョン2021 リリース2

• <u>検索と置き換え</u>機能がデザインビューイン搭載されました。デザインテキスト、スタイル、プロパティ、XPath 式内で入力するおけよ 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

•

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 に搭載された、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。デザインゴはXML やCSS ファイルといす。、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ドキュメイを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:

- ・ <u>複数のSPS デザイン</u> 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>テ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テ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。
- デザインビューでのマークアップを隠す: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かサポートされます。出力に関連する以下の機能がサポートされます:

- XSLT バージョン 1.0、 2.0、 と3.0 の 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) かりポートされ、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、おけは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 とともこ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 プロジェクト全体を簡単に受け渡すことができ るようこなります。
- 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)が更新されると、XSLTにより(更新箇所を含んだ)XML か動的にフ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>ソースドキュメント</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。

- <u>動的なコンテンソ</u>は、スキーマノースにあるノードをドラッグアンド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。
- <u>動的なコンテンツ</u>はデキスト、おけば、<u>入力フィールド</u>おけは<u>コンボボックス</u>といけうデータ入力デバイス</u>として挿入することができます。
- メインのデンプレートにより、デザインの構造が指定、管理されます。この構造はグロー・シレデンプレートと呼ばれるオプションのデンプレート要素にて修正することができます。
- グロー・バリテンプレート」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。.
- ユーザー定義テンプレート: 指定された XPath 条件式により、複数のアイテムコネル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ませまレードとより、XPath 条件式により、よど詳細なレードの指定を行うことができ、そのセクションコネオす る条件付けやフィルタルングを行うことができるようしています。
- <u>ユーザー定義要素</u>: この機能はデザイン内の任意の場所に(HTML、XSLT、XSL-FO といた)表示言語要素を自由に挿入 するために開発されました。
- <u>ユーザー定義 XML テキストブロック</u> 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。
- デザインフラグメト により SPS 内にあるテンプレートのモジュール化と再利用を行うことができるよう 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 なります(<u>モジュール SPS</u>)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 ケを他のモジュールでも利用することができるよう エジます。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 エジリ、メンテナンスを簡単に行えるよう エンドます。
- XSLT テンプレート: 生成されたスタイルシートへXSLT ファイルをイル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 お使用されます。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、デザイン内部か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 おはXSLFO 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ソやフォーマナを SPS に追加することができます。
- <u>ユーザー定義 XPath 関数</u>: ユーザー定義のXPath 関数を作成し、ドキュメント内でXPath 関数の使用か許されている場所 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い配置することができるブロックです。
- <u>青写真</u>: レイアナンテナー内に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います。
- XMLドキュメトでは、デーダ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ダ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されます。 SPSでは、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<u>動的テーブル</u>とて挿入することできます。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(おけよカラム)を生成することか可能によります。
- 繰り返される要素は、選択された(1つ以上の) ソートキーには述べ替えを行うことができ、並べ替えられた要素セイ 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 できます。
- <u>条件テンプレート</u> 機能により、XML ドキュメトトやン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 ドキュメトトの作成時に、ソーズドキュメト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ことができる条件は、XPath 1.0 まけは XPath 2.0 条件式で扱うことのできる条件に限定されます。

- 自動計算によりノースドキュメン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ーで変更)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には、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 条件式にて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)ませま動的(ソースドキュメト内にあるノードから取得)、ませま静的と動的パトの組み合わせにと、構成することができます。
- 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>静的と動的なリンク</u>をデザイン内へ挿入することができます。ターゲオ URI (講争的(SPS 内で入力) おさま動的(ソースドキュメント内にあるノードから取得)、おさま静的と動的パトトの組み合わせにと、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<u>、ラメータ</u>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 与えられた値によけーバーライドすることができます。
- 入力フォーマナ 入力フォーマナ機能により、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 うっとパンテーマナし直すことができます。入力フォーマナ機能は、<u>自動計算</u>の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 定義済み HTML フォーマナ を利用して、個々のSPS コンポーネ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>から個々のSPS ユポー およトに対して適用することができます。
- 更に、SPSの <u>グロー ッレなレベル</u>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ンティの値を選択し、条件付きでプロンティの値をセントすることができます。

- 関連項目

- <u>ユーザーインターフェイス</u>
- 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

1.4 SPS とは?

StyleVision Power Stylesheet(おけま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 をグラフィカルに作成するために使用されます。

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 おけまXML スキーマ)をベースしており、。SPS のデザイは 柔軟性に富んでおり、動的ならび「静的なコンテンを含めることができます<u>動的なコンテンツ</u> とは XML ドキュメト内にあるデータのことで す。<u>静的コンテンツ</u> とは、SPS 内に直接入力されたコンテンパことです。動的コンテンパよテキスト とて直接、おけは入力フィールドやコン ボドクス テーブルと いたコンポーネトの内部に含めることができます。更に、動的なコンテンンを(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た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を 配置することができ、イメージや マハドーレク、JavaScript 関数といたその他のコンポーネ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 のデザイム党成したら、作成したデザインをベースコンた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 かサポートされており、単一のSPS からHTML、RTF、XSL干O、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干O とW ord 2007 以降はEnterprise Edition でのみサポートされ、 RTF はEnterprise とProfessional Edition で、HTML 出力はBasic Edition でのみサポートされます)。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 をベースコンた XML ドキュメントの変換を行うこともできます。 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たさい。

SPS 出力

StyleVision では出力(Enterprise Edition ではHTML、RTF、XSL=FO、PDF、Professional ならびにBasic Edision では HTML)を直接生成することもできます。出力ビューのダブにより、アクティブな SPS ドキュメナの出力を、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」コマナド おけよ 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。

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 <u>Web サイト</u> から StyleVision をダウムロードした後に、実行ファイル(.exe) をダブルクトックして、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 内部に搭載されており、内部的な全ての変換に使用されます。StyleVision とおりに XSLT エンジンをインストールする必要におりません。。

以下のコンポーネントをインストールする必要があります:

- 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 ページエンテンなど、W ord ドキュメントに貼り付けることので きるエンテングの W ord ドキュメントか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
- 生成されたファイル
- 一般的な使用方法

1.6 用語解説

このセクションでは、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ナ順に表示されます。

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、<u>Altova Webサイト</u>を参照くたさい。
- SPS Style Vision Power Stylesheet を省略したもので、このドキュメンテーションでは Style Vision に お作成された デザインドキュメントのことを指し、.sps とら拡張子に お保存されます。 詳細については SPS とな を参照してくだ さい。

グロー・ VUJソ ファイルのセオ、フォルダーのセオ、そしてデータベースのセオに対するエイリアスです。各リアイアスコは構成セオか含 ース まれており、各構成がリソースにマンピングされます。StyleVision にてグロー・ VUJソースか使用される場合、 StyleVision におけるアクティブな構成を変更することで、割り当てられたリソ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。StyleVision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にある(大文字で表示される) XML スキーマ とは、 W3C の XML スキー マ仕様に準拠したスキーマのことを表します。 XML スキーマは 小文字で表示される) 全スキーマ のサブセナ どより ます
- URI とURL このドキュメンテーションでは、「ロケーター」としてお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、幾つかの用語か変更されました。例えば XPath 1.0 で ルートノードと呼ばれていたものが、XPath 2.0 では<u>ドキュメントノード</u> と名前か変更されています。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いるソノーのルートノードから始まる、マの条件式です。StyleVisionのダイアログにていて条件式を入力する場合、絶対 XPath チェックボックスレチェックを入れることで絶対・マ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 マが入力されます。

コンテキストア コンテキストアイテムとは、条件式が評価されるアイテム(ノードおけは文字列の値)に相対的なアイテムとなります。コンテ イテム / キストノードとは、ノードのコ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、例えば洛ステッ 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。 ード

カレントノード カレ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レテキスト ノード と等価にないます。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() 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ドキュメト要素と呼ばれます。ドキュメト要素は上 素 <u>キュメトノード</u>の子どなり、整形式のXMLドキュメト内にはボキュメト要素にはつけた存在することはないます。 GUIにおいて、ドキュメト要素はルート要素と表示されます。
- ドキュメントノードはボキュメト全体が含まれ、ドキュメトのソレ構造におおしトノードとなります。ルトノードは
 イン・
 XPath 条件式では、// にて表されます。StyleVision のスキーマソレーウインドウでは、「/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、レイアウトコンテナー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」なりま ジ す。青写真イメージをベースに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 が動的の場合)ノイノトリンクとして作成することができます。
- グロー・ジル要 スキーマソノーウイドウカーあるグロー・ジル要素リストに表示される要素のことです。XML スキーオにてグロー・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 なるため、グロー・ジル要素のリストにて表示されます。グロー・ジレテンプレート 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・シレテンプレートは、グロー・シレ要素 に対して定義することができます。グロー・シレテンプレートが一旦定義される プレーチ とドキュメント内に出現する要素がどここあろうと、その要素に対してグロー・シレテンプレートが使用されます。 グロー・シレ テンプレートの代わりに、ローカレテンプレートにてグロー・シレ要素を処理することができます。
- レイアオ・コン レイアオ・コンテナーとは、デザイン要素を絶対位置によ、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 ームをベースとている場合、レイアオ・コンテ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・ハイで作成し、ドキュメント内にコンイアオ・コンテナーを配置 することができます。
- □-カルテンプ ローカルテンプレートにより メインのテンプレート内にある(グローマ いおよび)非グローマ いの) 要素がどのよう 2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、 グローマ いテンプレート 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。作業用XMLファイルの ドキュメントノードがメインスキーマと関連付けられ、メインのテンプレート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。StyleVision において、このテンプレートはドキュメト要素 にマッチして、XSLT レート プロセッサーゴン場例に評価されるものとないます。スキーマソレーウィドウでは、ドキュメトノードの子として表示され ます。メインのテンプレート により、基本的ながキュメト構造が定義されるはか、入力ドキュメトカどのように処理され るかか定義されます。その他にも、ローカルテンプレート を含めることができ、グロー・シルテンプレート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ドキュメートを処理することで生成された出力のことです。StyleVision により生成 することのできる出力ファイルはwould be HTML 形式となります。StyleVision により生成された XSLT スタイル シートも出力とは考えられておらず、XSLT スタイルシートとして独立して記述されます。
- *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。静的アイテムコはテキスト、テーブル、リストのほか、 (URI か静的な)イメージやハイハシーレク、ブックマークかあります)。
- SPS コンポー SPS コンポーネントは (i) スキーマノード (例えば要素ノード)、(ii) 自動計算 やテキスト文字列といす。静的な ネント SPS コンポーネント、おけは (iii) SPS にて開始ならびに終了 タブとして表示される 定義済みのフォーマット しないます。

テンプレート ノードやノードのグループを処理するための処理セト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) こ 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、新日ご作成される SPS を伴った XML ドキュメントの開 始データとて 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

マージーに安当でなければなりません。 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。属性には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に対して作成されるテンプレートです。

テンプレート

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

)+ス・20 ク

2

作業用 作業XML/XBRLファイルはStyleVision内のSPS に関連付けられたXMLデータファイルで、StyleVisionに XML/XBRL ファイル 合、StyleVision内のSPS にて動的なXMLデータか処理されることはありません。2つ以上のグロー・シル要素が含 おんるスキーでをSPS がベーズしている場合、どのグロー・シル要素がドキュメント要素になるのかか曖昧になってしま います。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<u>ドキュメント要素</u>が1つしか含まれないため)。XBRLの機能はEnterprise Edition でのみ利用できることに 注意してくたさい。

XML ドキュメ XML ドキュメナト以下のために使用されます:(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に ント 使用されます。

- 三 関連項目
- SPS 심값
- 一般的な使用方法

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ルプシステムとしてご利用 しておれるはか、 <u>Altova Web サイト</u>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とい構成されています

- はどめこ SPS の説明や、StyleVision の主な機能やエノセプトを紹介します。
- <u>ユーザーインターフェイスの説明</u>: StyleVision GUI の概要を記します。
- <u>チュートリアル</u>:StyleVision に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 のデプロイやプロジェ かといた、高度な機能の使用方法について記述します。
- <u>SPS ファイルニンテンツ</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リルと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。
- <u>SPS ファイルの高度な機能</u>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 出力ドキュメン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。
- <u>SPS ファイルプレゼテーション</u> SPS エポーネトのフォーマトや、レイアナにてご説明します。
- SPS ファイル追加編集機能: SPS を更にパフフルニ使用するオーダの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・シリノースや、更なる検証、変数やパラ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や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 StyleVision の概要を知るために、はじめに、ユーザーインターフェイスと使用方法の概要 セクションに関するセクションへ目を通す ことをお勧めします。その後、<u>チュートリアル</u>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(<u>SPS ファイルエンテンソ、SPS ファイル構造、SPS ファイルの高度な機能、SPS ファイルプレゼテーション、SPS ファイル追加</u> 機能)では、StyleVision 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。<u>レファレンス</u>セクションは主全てのソ ールドーアイエン、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ご記述されており、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・タリオ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たさい

• *(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>

 アプリケーション フォルダー: アプリケーション フォルダー・ゴは Altova アプリケーションが存在します。 デフォルトでの、アプリケーション 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32 bit バージョン	C:\Program Files (x86)\Altova\

メモ StyleVision は、プラオフォーム更新済みのWindows Server 2008 R2 SP1 ませお以降 でもサポートされています。

サポート オプション

StyleVision、に関する質問や問題がある場合、以下のサポートオブションかご利用になれます:

- 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します。
- 2. <u>Altova Web サイト</u>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

2 ユーザーインターフェイス

StyleVision GUI は、以下のいいから構成されます(アル表示されているユーザーインターフェイスにはすべてのサイドンド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バーコは最小化、最大化、ドキュメトを閉じるけのがなか表示されます。
- ツール・一領域:様々な<u>ツール・</u>ならびにそれらツール・イン対応したコマイショートカナについては<u>ユーザーリフォンス</u>にて記述されます。
- ダブ化された、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</u> <u>デザインの編集</u> そして <u>XSLT スタイルシート と出力のプレビュー</u>を行うことができます。
- <u>デザインサイドバー</u>: <u>デザイン概要、スキーマソノー、デザインソノー、スタイルルポネリ、スタイル、プロパティ</u>ウイボウが表示され、アプリケーションウイボウドックすることができるおか、ウイボウとして表示することもできます。
- 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



<u>メインウィドウとデザインサイドバーの詳細についてはこのセクションのサブセクションに記述されます。</u>

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

- メニューコマボ セグション
- レファレンスセクション ツールレー

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か表示されます(以下のイメージを参照)。



メインウィンドウ内の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

以下にあるビューインで、キュメノトは表示されます。ビューイは同時につしか選択することができません。

- デザインビュー: 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リックして、デザイン、おさま 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が表示されます。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、メインウィンドウ下部に表示されるタブをクリックすることで切り替えることができます。選択されたビューのタブが、イライトされます。

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で行います。デザインビューでは、())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エンテンン作成ならびに編集機能を使用、まけよ(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ナ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。デザインビューイは搭載されたこれらの機能について以下で説明します。

3		
	メインウィン	ドウ
7	ククティブなドキュメン	Ь Ø
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ククティブなビュー	クリックして JavaScript エディターへ切り替え
デザイン ▼H	TML	
Doc1.sps	[Doc2.sps]	40

デザイビューは、<u>JavaScript エディター</u>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 <u>JavaScript 関数</u> を作成することができます。 <u>JavaScript エディター</u> 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ムボタンをク 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ムボタ ンをクトックして、ドロップダウンメニューからデザインを選択します。

デザインビューでは、メインテンプレート、グロー・シリテンプレート、ページレイアナトテンプレート、そしてデザインフラグメーというを複数のテンプレートをSPS内は収めることができます。ツールドーアイエンとして表示されているテンプレートデザインフィルターを使用することで、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

マークアップアイコンを使用することで、デザインビュートこあるマークアップタグの表示を切り替えることができます(以下を参照)。

\triangleleft

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クアップ(名前なしのタク)を表示、(ii) 大きなデザイン マークアップ(名前付きタグ)の表示を表します。 小さなマーク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。 HTML 出力ビュータブニあるドロップダウオオジェをクリックし、XSLT オプションまけは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。

нтмに出力	ידאכ	メインウイン タブがアクティブ	ドウ
XSLI X91	1	P2237	クリックして HTML
		/	★ フレビューへ切り替え
デザイン	XSL	T-HTML ▼	
Doc1.s	ps]	Doc2.sps	4۵

XSLT ビュー

ビューーコよ 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 されるオンパニスタイルシートは新たっこ生成されます。

出力ビューダ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 ビュービ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)、クリップボードへのコ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、<u>作業用 XML ファイル</u>がHTML プレビュータブ がフレクされるたびに、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こて編集される一時的なファイルではよい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 プレビューカ表示された場合作業用 XML ファイルを 割り当てるようこ促されます。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 ファイルが生成 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(おさまサイドバーウィンドウはSPSのデザインを行うさめのGUIコンポーネントで、アクティブなビューイ工関する情報を提供します。以下に示される各サイドバーについては、このセクションのサブセクションにて解説されます。

- <u>デザイン概要</u>
- <u>x+-vy</u>
- <u>デザインソト</u>
- <u>スタイルポネリ</u>
- <u>2911</u>
- <u>৴৸ ঈ</u>
- <u>メッセージ</u>
- 検索と置換

ビューのレイアウト

ビューのレイアナトはよそのビューでどのサイドバーが利用可能なのかと、これらサイドバーがGUI にてどのようご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対してカスタマイズすることができ、カスタマイズは、(i) ビューにおける個々のサイドバーの表示(表示、メニュー、ませはサイドバーのタイトルバーを右クリックして、「隠す」を選択する)と、(ii) 必要に応じて GUI にてサイドバ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バーを除く全てのサイドバーが表示されていた場合、次回デザインビューレイアナトが変更されるまで、このレイアナか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、W ord 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。

サイドバーウィンドウアッキングとフロート

サイドバーウイドウは Style Vision GUI に対してドックするか、画面上にフロート状態にすることができます。ウイドウをドックするコおクト ルーをドラッグし、ドラッグする際に表示される矢印にドラッグしてくたさい。内側に表示される矢印ヘドロップすることで、矢印の表示された ウイドウの内側に対してドラッグされたウイドウを配置することができます。外側にある矢印ヘウイドウをドロップすることで、矢印の表示され ているウイドウの縁にドッキングが行われます。ウイドウをフロート状態にするコよ(i)タイトルーをダブルクトックするか、(ii)タイトルーを ドラッグして、画面上に表示される矢印以外の場所でドロップしてくたさい。

その地にも、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ゴよ、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リックして、フロートを選択します。このメニューへは、ドックされたウインドウのタイトルバーを右クリックすることでもアクセスする ことができます。

2,94,11			🚽 🕁 🗙
La 🗜 圮		70−ŀ(£)	Ś
2.947.JU :	~	ドッキング(<u>D</u>)	
■ 1 テキスト		タブ化されたドキ:	שאטאנ ד
[1 段洛		自動的(認す(<u>A</u>)
		隠す(<u>H</u>)	
		🕀 UI	
		🕀 XSL-FO	
		田 テキスト	
		⊡ テーブル	
		田 フォント	-

フロート状態のフィンドウをドッキングするコまそのフィンドウのタイトルレーを右クリックして、表示されるメニューからドッキングを選択します。最後にドッキングが行われた場所にフィンドウルドッキングされます。

デザインサイドバーウインドウを自動的に隠す

ドックされたウイボウは自動的に隠すこともできます。サイドバーウイボウか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、メイノウイボウェウイボウか、懐示されます。以下にあるスクノーンショナでは、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ダインウイボウ上に表示されます。



マカスカーノルを表示されたウイドウの外側へ、おけおブとおりの場所に移動させると、そのウイドウがダブとして GUI の端に再度収まます。

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、必要な時にさきたクセスするようよ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ごな ります。

ウイドウを自動的に隠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隠すボタン(ピンのかたちをしたアイコン)をクトックします。 それ以外にも、ウイドウのタイトルドーを右クトックするか、ドッキングされたウイドウのタイトルドーニあるメニューボタンをクト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を選択して、この機能を有効にすることができます。



ウイドウが自動的に隠されるよう」なります。

特定のウイドウニオして自動的に隠す機能を無効にするこは、マウスカーノルをそのタブへ移動してウイドウが表示された状態で、自動的 に隠すポタンをクリックします。それ以外にも、ウイドウのタイトルドーを右クリックするか、ドッキングされたウイドウのタイトルドーこあるメニュー パタンをクリ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。

スタイルリポジトリ	▼ = ×
4∎ 6₽	XB
©¶0'0−バル	
田 セレクター:.info	
…田 セレクター:.explanation	

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。

サイドバーウィンドウを隠す(閉じる)

サイドバーウィンドウが隠されると、GUI 上から直接確認することはできません。サイドバーウィンドウは最大化された状態(ドッキングまとはスロート状態)か、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トックします。それ以外にも、ウィンドウのタイトルドーを右クトックするか、ドッキングされたウインドウのタイトルドースあるメニューボタンをクト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)状態のフィンドウを再度表示するコよ、表示 メニューか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ンドウが、隠された時の位置と状態(ドッキングまナコンロート状態)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 に対してスキーマノース、グロー・ゾレ ラメーター、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。これらコレポーネントをまとめて表示することで、コンポーネントの集中管理を行うことができます。

テザイン概要	×
▽ ソース	
🚰 \$XML (メイン)	0
スキーマ	Data.xsd 💽
作業用XML	Data.xml 💽
テンプレートXML	0
▽ モジュール	
ᡖ TextState.sps	0
新しいモジュール	<u>,を追加</u>
▽ CSSファイル	
🛃 Samples.css	0
<u>新しいCSSファイ</u>	<u>ルを追加</u>
▽ パラメーター	
😫 \$Year	2012 O
新しいパラメータ	<u>ーを追加</u>
🔻 XSLT ファイノ	l I
🚮 NameAttrUpper	.xslt 📀
<u>新規XSLTファイ</u>)	<u>レを追加</u>

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ドをクリックすることで、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、 おっぱ ユーザー定義スキーマ

t 🔁	XML スキーマ/DTD/XML を追加(S)
-8	ユーザー定義スキーマを追加(世)

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 ファイルセクションの下部にある、新たに追加するコマメドをクリックすることで、対応するセクションに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 にてアクティブになっている
 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 複数のモジュール か表示されており、そのうち1つのモジュールが選択されて、る際に有効にないます。選択したコマンドにより、選択されて、るモジュールの位置を変更することができます。

ソース

ソースセクションのJストはよ スキーマと、SPS をベースはしている作業用 XML ファイル か表示されます。 (右クトック、おけよしテキストメニューアイコン 2 をクト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コは、アクティブな SPS により使用されている <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 ドをクトックし、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>モンユールは表示される順序</u> はは意味があるさめ、複数のモジュールは表示されている場合、(モジ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ジ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トゴよ、StyleVision にて選択されたコマンドを用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クト</u> (モジュール内部のオブジェクト)は<u>デザインソレーサイドバー</u>に表示さ れます。

CSS ファイル

CSS ファイルセクションコよ アクティブな SPS により使用されている CSS ファイルあまっされます。新しい CSS ファイルを追加コ マボをクトックして、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 ファイルあ表示される順 序 には意味かある」が、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 ヘノ下 ヘコマボ が選 択可能しなります。目的のコマボをクトックすることで、選択された 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イコよ、選択さ れたモンュールを XMLSpy にて開くためのコマボ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 CSS ルールは、スタイルルポネリサイドバーに表示されます。

パラメータ

ノ ラメーターセクションコは SPS 内にあるグローノ ジレ ラメーターカ表示されます。セク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、新たなノ ラメーターを追加することができます。ノ ラメーター名や値をダブルクトックすることで、それそれ名前や値を編集 することができます。ノ ラメーターを削除するコよ、目的の ラメーターを選択して、コンテキストメニューカら削除コマイドをクトックする必要かあ ります。

XSLT ファイル

XSLT ファイルセクションゴよ SPS 内にインポートされた XSLT ファイルは装示されます。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・シ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: <u>XSLT テンプレート</u>を参照く ださい。

- 151-9
- <u>______</u>
- EVI-IUSPS
- <u>デザインプラグメト</u>
- スクノプトの使用

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ン):

- 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、デザインへドロップします。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 にて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。



ル十要素

\$XML 以下にある各スキーオマ対して、選択された<u>ルーキ要素</u>(おけよ<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 オニ関連付けられている全てのルート要素が表示されます。各ルート要素を展開すると、モデルソノーが表示されます。これらルート要素内に 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リーポイントはメインスキーマのドキュメントノードで、ド キュメントノードはい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)。
サイドバー

37

スキーマインサレ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 いう ポタンをクルクします。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マル要素からルート要素を選択することでもあてきます。 複数のルート要素を選択することで利用可能になるオプションの説明については、<u>SPS構造 スキーマノース</u>を参照くたさい。

ルート要素の選択
スキーマで利用可能な利用可能な全てのルート要素が以下のリストに表示されます。 要素の隣にあるチェックマークは現在スキーマソースツリー内で可視化されている示しています。 ゲレーになっている要素はデザインで使われており、チェックマークを削除することはできません。
スキーマソースシリー内で可視化したい要素をチェックしてください。 ▼ 約1:Department n1:Desc n1:EMail n1:OrgChart n1:Person n1:para
<u>ОК</u> <u></u> +юслел

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

グロー・ジレ要素とグロー・ジレ型

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・シレテンプレートを作成するために使用されます。 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デザインフラグメント 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。

デザインフラグメトを扱うための情報については、デザインフラグメ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 E をクルクすることで、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 関数をダブルクルクすることで、その関数がダイアログにて開かれ、編集を行うこ とができるようゴムルます。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たとい。

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に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)デフォルトで新たに作成された 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 エトリーの値フィールドをダブ ルクトックして、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。

ツールシーとスキーマソノーアイコン

以下のソールドーアイエイは、一般的なスキーマソノーサイドバーコマンドに対するショートカナトとむます。

スキーマソノーで使用されるシンボル

スキーマソトーンおおシンボルを以下に示します。

- Name 要素
- = Mgr

属性



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



グロー・ ジレの型は型を単純型ませま複合型となります。 複合型はシアン色のアイエインと表示され、単純型はプラナン色のアイエインと表示されます。

- 🗉 関連項目
- <u>SPS 構造の作成 | スキーマノース</u>

2.2.3 デザインツリー

デザインツリーサイドバーコは SPS デザインの概要が表示されます。

デザインツリー ×
🕭 📲 📉 🖊 🛤 🕍
🚯 AutoCalc.sps
田 🗟 スクリプト
田 🗮 メインテンプレート
🐘 🖏 グローバルテンプレート 👥 📗
🔤 デザインフラグメント 🛨
📆 XSLT テンプレート
🛄 XPath 関数群 🗗

デザインソーのルートにはSPSの名前が表示され、マケ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は、以下のカテゴリーに分類されます:

- スクレル: Style Vision の Java Script エディターにより SPS に対して定義された全ての Java Script 関数が表示されます。
- メインのテンプレート: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・シレデンプレート:現在のSPS内にあるグロー・シレデンプレートや、インクルレードされた全てのSPSモジュールにあるグロー ノシレテンプレートか表示されます。
- <u>デザインプラグメト</u>:デザイン内にある全てのデザインプラグメントが表示され、デザインプラグメントの作成、編集、名前変更、そして削除を行うことができます。
- <u>XSLT テンプレート:</u> インポートされた XSLT ファイル内にある XSLT テンプレート か表示されます。
- <u>ユーザー定義 XPath 関数</u> 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。

ツールドーアイコン

以下のソールドーアイコンはスキーマソノーサイドバーにある一般的なコマイのショートカナノされます。

- デザインズオしてデザインフラグメト、メインテンプレート、おゴコレイアナアイテムを追加します。アイエンの左側をクリックすることで、デザインフラグメトトが追加されます。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すっかのリストカ表示されます。
- 選択されたアイテムを削除します。グロー・シレテンプレートませるレイア・オサブソレ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。
- 同期ソーを切い替えます。有効」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してなっている場合、デザインソノー内の対応するノードが選択されます。アイエンが無効してなっている場合、デザイン内の対応するノードが選択されます。アイエンが無効してなっている場合、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。
- Image: 同期オプタンが有効になっている場合、スキーマソレーにあるその他のアイテムが自動的に縮退され、デザイノニてアイテムが選択されます。このオプタンはソレーを同期オプションが有効になっている場合には引援択可能にないます。

デザインソーの表示を変更する

デザインソー内にあるアイテム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、デザインソーの表示を変更することができます(以下のスクノーシンヨットを参照)。

×	削除(P)
₽	デザイン フラグメントを作成(E)
	全て展開⊗

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ら削除します。デザインフラグメントを作成により、デザイン内に にデザインフラグメントを作成し、そのアイテムをデザインソノード追加します。全て展開コマイドにより、デザインソノ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

スクリプトはよ、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リプトが表示されます。メインテンプレートはおイン テンプレートのソレーが表示されます。アイテムを右クリックして、削除を選択することで、ツレー内にあるアイテムを削除することができます。

グロー・ジレテンプレート

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 にて定義された・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て定義された・グロー・シレテンプレートは大クレーで表示されます。各グロー・シレテンプレートはよ、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・シレテンプレートは大方に行った。



現在のSPS内にあるグロー・シレテンプレートをクリックレ、ツールレートこある削除ボタンをクリックするか、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

<u>デザインフラグメト</u>アイテムこより、現在のSPS ならびに追加された SPS モジュール内にあるデザインフラグメトカ表示されます。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 ンフラグメトロよ、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、削除コマイをクリックするか、コンテキストメニューから削除コマイドを選択することで、そのデザインフラ グメートを削除することができます。コンポーネントがデザインとソレ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・・・・をクトックすることで、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メリ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・・の削除パタンをク 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)。

デザインツリー	x
🚇 • 🔛 🚰	2
🚯 DB.sps	
🔢 🔁 スクリプト	
🔢 🗰 メインテンプレート	
🖽 🖽 グローバルテンプレート 🛨	
田 🛱 デザインフラグメント 🛨	
📃 🛱 XSLT テンプレート	
match - /	
match – italic	
match – strong	
name - BytesToKBytes	
<u>爲</u> XPath 関数群 →	
スキーマツリー デザインツリー	

イポーされたXSLT テンプレートには2種類あます: (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、これシ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ついては、XSLT テンプレートを参照くたさい。

- 🖃 関連項目
- <u>デザインプラグメント</u>

• <u>スケノプトの使用</u>

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・シルな 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ルールとグロー・シルニ定義された 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。



スタイルレポネリサイドバーコオグローバルと外部とら名前で、2つのソレカ表示されます。外部ソレーコは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かきまれます。グロー・シルソレーコはSPSと関連付けられた全てのグロー・シレスタイルかきまれます。

スタイルルポネリの表示構造は以下のようてなっています:

```
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)
- セレクター-1
- プロパティ-1
- ...
- プロパティ-N
- ...
- セレクター-N
+ ...
+ CSS-N.css
グローバル
- セレクター-1
+ セレクター-1 プロパティ
```

+ セレクター-N

スタイルレールの優先順位

グロー リ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トを表すセンクターが 指定されている場合、グロー・ジ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・ジ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レ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ー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ルールが適用されます。

スタイルルポントリニてスタイルを管理する

スタイルポネリサイドバーでは、ツールバー上にあるアイコンを使用する、おけコン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 5 をクリックすると、選択されているソノー(外部おさなグローノ ジレ)により、新たな外部スタイルシートエトリーを外部 ツーへ追加し、新たなグロー・シルスタイルエナ・リーをグロー・シルソーへ追加することができます。新規エナ・リーが、ツー内にて既に存在す るリストへ追加されます。追加コマンドはコ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・バルスタイルを使 用するす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。

挿入: 挿入アイコン 「「」により、(外部ツノー内にて)選択されたかい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・ジレンノー内にて)選択されたグロー・ジレスタイルエントリーの上部へ新たなグロー・ジレスタイルエントリーを挿入することができます。 コマトレコ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ッシレスタイルを使用する詳細な情報についてしま、次を 参照してくたさい 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、選択されたか部スタイルシートやグローノ ジルスタイルを、ツノー内部にてそ れぞれ上や下へ移動することができます。これらのコマンドを使用することにより、外部スタイルシートやグロー・シル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ボリ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・ジルに定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 📴 、全て展開 🛂 、全て縮退 🏥 .コマボがソールドーとコンテ キストメニューコマンドにて利用することができ、空でないリストオプショ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レティイゴが 表示されるよう」さいます。このオプションが選択されていない場合、全てのスタイルプロレティの表示されます。全て展開ならびに全て縮退設 定を、空でない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 とをクリックすることで、選択された CSS ルールマシして important とら CSS 値をセナします。

すべてを再ロード: すべてを再ロード アイコン 1 により、全ての外部 CSS スタイルシートを再ロードします。

リケット・リセットアイコン と により、選択されケット部スタイルシートおけよグローノ ジレスタイルシートを削除することができます。

スタイルレポントリ内にある CSS スタイルの編集

スタイルルポネリでは、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くたい。

• 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ご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、デザインビューーこで選択されたSPS コンポーネントに対して、CSS スタイルをローカルにて定義することができます。これはスタイルレポントリサイドバーーこでグローバルニ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つのいやいから構成されます:

- ・ 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 OSS 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、CSS スタイル値の設定を参照くたさい。XPath アイエン ピー 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、一のポタンとエンテキスト メニューから選択することができます。

- 空でないリスト
 し
 このオデ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コンポーネントにおいて、値が与えられているプロンティだけがアルファベット順で表示されるようゴンはます。オプションが無効になっている場合、すべてのプロンティが表示されます。このオプションには、特定種類のコンポーネントに対してどのようなプロン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。選択されたコンポーネント型に対して新たなプロンティを定義するコよこのオプションを無効にした状態で、空のプロンティーヘアクセスする必要があります。
- すべて展開 2: 左側で選択されたコンポーネント に対して、すべてのプロノテルを右側のペイノに表示します。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 左側で選択されたエンポーネント に対して、すべてのプロ・ティを縮退された状態で右側のペイムに表示します。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レーアイコンのノセット

重要フラグの切り替えアイコン Eをクトックすることで、選択された CSS ルールに対して Important とら CSS 値をセオします。

- 三 関連項目
- CSS スタイルと作業
- · スタイルポトリ
- 定義済みのフォーマト

2.2.6 プロパティ

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

プロパティ		×	
님 🕂 📲	х ратн ×		
プロパティ:	属性	値	1
条件分岐	📮 when		
テンプレート	XPath	(contains(string(@	
			l

プロ・ティサイドバーは、以下にある2つのいいから構成されます。

- 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、次のセグショ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、以下の次のセクシンを参照してください、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、ツールドーのパ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 しき: このオブ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ロ、ティオさけがアルファベント順で表示されるようごないます。オブションが無効になっている場合、すべてのプロ、ティカ表示されます。このオブションにより、特定種類のエンポーネントに対してどのようごプロ、ティか定義されているか、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、ティを定義すること、このオブションを無効にした状態で、空のプロ、ティーペアクセスする必要があります。

_H

- すべて展開 ៉ 左側で選択されたコンポーネント に対して、すべてのプロ・ 守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 🔤: 左側で選択されたコンポーネント I つうして、すべてのプロ・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。

ツールドーアイコンのルセナ

リセットアイコン と をクリックすることで、選択されているプロン ディの値をデフォリトに戻します。

コンポーネントとプロレティグループ

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・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・ティグループを示します。.

コンポーネント	プロパティグループ
コンテンツ	コンテンソ、共通、イベト
7777	テキスト; 共通: イベト
自動計算	自動計算 共通 イベナ
条件分岐	When
データ入力デバイス	共通: データ入力デ・イス: イベント; HTML
イメージ	イメージ、共通、イベト; 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
ודדעדי ו	ップブフ, 共通: イベト; HTML

コンポーネ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。

各プロハティグループの説明を以下のテーブルニテします。

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ないます。値フォーマットプロ・ディ では、数値や日付データ型といた、自動計算の書式を指定することができます。XPath プロパ ティでは、 <u>自動計算</u> 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・ディグループは、テンプレートと自動計算コンポーネントを除く全てのエンポーネントに て利用可能です。class(クラス名)、dir(書き方の方向)、id(ユニークID)、lang(言語)、title (名前)プロ・ディをコンポーネント(こ対して定義することができます。
<i>₸-\$入力₸›¥</i> ₮	コンボボックス、チェックボックス、そしてラジオボタイニおける値の範囲を指定します。このプロ・ティ グループは入力フィールドとボタイニ対して適用されない点に注意してくたさい。
1124	以下のデイアナサイドのHTML イベトロ対して、 <u>JavaScript 関数</u> を有効します: onclick, ondblclick, onkeydown, onkeypressed, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup.
HTML	<u>データ入力デバイス、イメージ、リンク、テーブル、パラグラフ</u> のエンポーネント型 >さして利用す ることができます。 <u>データ入力デバイスとパラグラフ</u> は複数の種類があり、 <u>テーブル</u> はよサブユ ンポーネントが備えられている点に注意してください。これらプロン 守ィは対応する HTML 要素 (img、table、p、div など) [>さ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ン 守ィ は、選択されたエンポーネントにより変化します。これらプロン 守ィー対応する値を 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、ノラグラフと他の定義済みのフォーマトと条件分岐に固有のプロレティが用意されています。これらプロレティについては、それぞれのセグションで記述されます。

プロ・ティの値の設定

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。
- プロパティの値カラムの右側にある編集ボタン ・・・・ を使用する。編集ボタンをクリックすることで、そのプロパティのダイアログが表示されます。

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

コン の有効/無効を切り替えることで、XPath 条件式をプロレデ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 アイコンを有効にすると、このプロレディに対して XPath 条件式を入力し、その XPath 条件式の返り値を、プロレディの値として使用することができます。 これにより、プロレディの値として XML ドキュメント 内にある値を、 ランタイムにて取得することができます。 ようようようます。 XPath アイコンカ無効になっている場合、プロレデ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レディを参照くたさい。

プロ、守ィ値の修正と削除

プロ、ティ値を修正し、これまで記述された方法を使用するコは、プロ、ティの値の設定してくたさい。プロ、ティ値を削除するコは、プロ、ティを選択し、プロ、ティサイドバーのソール・ドーンあるノセントアイコンズをクレックしてくたさい。

- 🗉 関連項目
- <u>2911</u>

2.2.7 メッセージ

StyleVision か最初に開かれる場合、メッセージサイドバー(アのスクリーンションオ) 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は、表示 | メッセージをクリックします。



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 ドキュメトル 理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、または、 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、プロパティ、XPath 式内で直接入力するおけば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<u>[編集 | 検索</u>]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、長示 | 検索と置換]をクリックします。

Find & Replace	×
Shares ~	Find
×	Replace
▼▲CGG ۹.9.8 X	
E Text <u>main template '/'>template (source) 'XML'>template (element) 'n1:Ord</u> <u>main template '/'>template (source) 'XML'>template (element) 'n1:Ord</u> Properties Found 14 match(es)	<u>aChart'</u> >: a <u>Chart'</u> >:
	•

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

検索

「検索対象」フィールドに検索する用語を挿入します。検索ボタンのドロップダウン矢印をクトックして検索オア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>サイドバーニて入力される) スタイル内 (iii) (<u>プロ/ ディ</u>サイドバーニて入力される) プロ/ ディ内。 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に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・ティと(iv)Xpath 式のオメのグループに整理されます。(上のスクレーンショント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オータリン構造されています。検索されたアイテムまだは都層内のレ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。

結果ペインは以下の左側からのコマドのナダのアイコンを持っています:次のおよび前のアイテムこ移動する。クル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 ボ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 を作成するこかさい注意すべき点を簡単に解説して、きます。最初に SPS の作成とセイアップに ついて説明し、その後 SPS 内にコンテンンを挿入する方法について、SPS コンポーネントのフォーマナト方法、そして、自動計算や条件という、SPS に搭載されている2つのパフフ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、おこ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、(マイ)ドキュメントフォルダー 以下のC:\Documents and Settings¥<username>¥My Documents¥Altova¥StyleVision2021¥StyleVisionExamples¥チュートリアル \QuickStart 「コロめられてします:

- 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すしておくことで、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には、ファイルの保存を忘れないようしてくたさい。

- = リンク
- 次のステップ新たな SPS の作成とセメアップ

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

- <u>新規 SPS ドキュメントの作成方法</u>。
- SPS 1文化てスキーマノースを追加する。
- <u>SPSのXSLT バージョンを追加する</u>。
- <u>作業用 XML ファイル害り当て</u>。
- 出力のエンコードの指定。
- SPS ドキュメトの保存。

このセクション内のファイル

このセクション内で参照されてしるファイルは以下の場所で見つわことができます: マイドキュメトフォルダーC:\Documents and Settings\<username>\My

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

- QuickStart.xsd はSPS が、一スにされているXML スキーマファイルです。
- QuickStart.xml は出カプビュ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 ドキュメントの作成

新規 SPS ドキュメーを作成するコは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、アプリケーションソール ドー ある 新規アイコン む をりいりします。

新規デザインを作成ダイアログボックスが表示され、(i)フレーッンドのドキュメートデザイン、ませま(ii)フォームをベースリュナジキュメートデザイン(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。

新規デザインの	2作成
StyleVisionは 23 つのデザイン種類は ほぼすべてのデザイン	種類のドキュメント デザインをサポートします: フリーハンドとフォームを基にしたものです。以下の説明を読んで 1 つ選んでください。2 ドキュメントの同一ページ内でも混合することができます。 両方のデザインにて、テキスト、テーブル、イメージ、編集フィールドなど、 ンアイテムが使用可能です。
◎ フリーハンドの)ドキュメントを作成
	フリーハンドデザインは本、新聞、ドキュメンテーション、レポートのデザインを行う際に有効です。テキスト、イメージ、テーブルなどを ページ内で温合させることができ、個々のサイズに従って自動的に配置されます。ページは自動的に生成され、テキストはページ に収まるようにラップされ、テーブルの自動幅設定を使うことでページの大きさに合うようにテーブルの大きさを変更することができま す。
◎ フォームを基(こしたドキュメントを作成(絶対位置)
Haar Ca Ayr Ca Haar Ca Haar Ca	フォームを基にしたデザインは、絶対位置で指定されたページにて編集コントロール、ラベル、イメージまたはテーブルを含む入力フ ォームを作成するのに最適です。通常すべてのページが個別にデザインされます。 用紙上に既存のフォームをお持ちの場合は、その用紙をスキャンしてデザインの春写真イメージ ファイルとして使用することができま す。この方法を使えば春写真イメージにある位置に従ってデザイン アイテムを配置することができ、オリジナルのフォーム位置に一 致させることができます。
	青写真イメージを選択 参照(B)
	OK(0) キャンセル(C)

フレーットがのドキュメントデザインでは、ドキュメン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)にフィットするようコンイアウトされます。ドキュメントコンテンン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。

フォームをベースコンナギュメントロン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よ、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ロジキュメントコンテングを配置 することしています。既にあるフォームのデザインをSPS にて再現したし、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。

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ルクして、「OK」をクルクしてく ださい。

SPS1.sps とう名前の新規ドキュメントが作成され、デザインビューに表示されます(以下のスクレンショナを参照)。



<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>スキーマソレー</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 I文北て QuickStart.xsd スキーでを使用します。以下の操作により、スキーでをスキーマノースとして追加することができます:

 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。



- ファイルを開くダイアログボックスが表示されるので、(マイ)ドキュメトフォルダー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。
- 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、(マク)ドキュシトフォルダー 内のC:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\QuickStar t.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たさい(i) デザイン概要内にある \$XML エントリーに、スキーマや <u>作業用 XML ファイル</u>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<u>グロー シル</u> 要素 選択した、<u>ルト要素(ドキュメト要素)</u>が表示されます。このスキーマの場合、グロー<u>シル要素</u>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 <u>グロー シル要素</u> か<u>ずべてのプロー シル要</u> 素ツノーに表示されます。

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 デジョンを指定する コよ アプリケーションソール デーから 300 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せま変更

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コま、デザイン概要サイ ドバー内にある 作業用 XML ファイル の行を右クルク(おけま右側にあるエンテキストメニューアイエン の をクルク)して、エンテキストメニュ 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か行われ、デザイン概要し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) ドキュントフォルダー 内にある C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\QuickStart.xml が 作業用 XML ファイルとしておという。

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 のセオアプを終えた後に、MyQuickStart.sps (マイ)ドキュメオフォルダーC:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart フォルダーにて MyQuickStart.sps とう名前でドキュメオを保存してください。メニューコマナ 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
- 次のセクション 動的なコンテンソの挿入
- <u>チュートリアルスタートページ</u>

3.2 (XML ソースの)動的なコンテンツの挿入

このセクションでは、XMLドキュメン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 しま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。(*下のスクノーンション*).



- 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>ベドラッグして、矢印カーノルが挿入状態に変わオ・状態でメインテンプレートベ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、<u>HTML 出力のプレビュー</u>(*下のスクリーンショ・*/)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 <u>デザイン要素の挿入 ツール・</u>ツール・→ こ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ゴンリます。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。(アのスクリーンショント).



4. HTML プレビュー へ切塔えて、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. <u>スキーマソノーサイドバー</u>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. <u>スキーマソノーサイドバー</u>内にある dateline/place 要素の city 属性を選択します(アのスクノーンショオ)。



- @city 属性ノードを <u>デザインビュー</u> ペドラッグして、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。
 @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
- 8. @country ノートを <u>プラインビュー</u> ヘトロソルし、@city 属性の終了タク 直後にトロツノし、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になっているはずです:



HTML プレニ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に注目してくたさい。

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。このセクションを終了する前に、以下のスクノーンシ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に注意してくたさい(スキーマ内の順番と異なっています)。synopsis とsource 要素をデザイムに追加した後に、HTML プレビューから出力を確認してくたさい。メインのテ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(ドキュメントの構造と出力ドキュメントに関する詳細については、デンプレートとデザインフラグメントを参照ください)。

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たまか、これら全てのノードはデキスト文

字列とて出力されます。
<u>条件を使用する</u>セクションでは、コンボボックスと、てノードを作成する方法について学習し、
<u>グロー・シレテンプレート</u> <u>とその他のコンテンソの使用</u> 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。

次のセクションを始める前に、ファイルを保存するようしてくたさい。

- = リンク
- 次のセクション、静的コンテンソの挿入
- 前のセクション 新規 SPS の作成とセナアップ
- <u>チュートリアルスタートページ</u>
- <u>SPS ファイル構造</u>

3.3 静的コンテンツの挿入

静的コンテンソ出ま、XML ソースから得られるコンテンソ出は違い、デザイン内に直接入力ませま挿入されるコンテンソのことです。 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:

- <u>イメージ</u>
- <u>水平</u>線
- <u>7777</u>

静的イメージを挿入する

挿入する静的イメージは、(マイ)ドキュメトフォルダー以下の: C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\QuickStart\NewsItems.BMP で、ドキュメトのへダーとて使用されます。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)。

イニシャルドキュメントセ <u>プロパテ液編</u> クション <u>集…</u>	
●\$XML Opresswatch Onewsitems Onewsitem Oheadline (コンテンツ) Oheadline	

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)。

イメージの挿入	×
「静的」「動的」「静的ならびに動的」インライン データー	
アドレス(E):C:¥Altova¥StyleVision2012¥StyleVisionExamples¥Tutorial¥QuickStart¥NewsItems.bm	P
 単純な指 単純な指 上のフィン 	争りアドレスの入力に、 ↓ −ルドを使用してくださ ↓
<u></u>	キャンセル

- 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、ドキュメトのヘッダービキュメ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。

水平線のプロレ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・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. <u>デザインビュー</u>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、プロノティのボックスグル ープを開きます。margin-botton プロノティの値に12ptをセットします。
- 4. <u>HTML プレビュー</u>Iこて出力を確認します。

各ニュースアイテムの最後に水平線を挿入することはます。newsitem要素にある終了タグ直前にカーノルを配置することで、各 newsitem要素の終端にラインの表示されるようはいます。ラインの size プロ 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 ティー・イン・イニてラインを選択し、sizeの値に例えば3 を入力してみてください)。

静的なテキストを挿入する

実は(XML ソースから)動的なコンテンン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します。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、以下のようこなります:

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 sandfiltering 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 ならいにplace/@country 属性のエレテンが、空白スペース無しこ出力されている点に注目してくたさい。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。デザインにて、date 要素の直後にカーノルを配置して、コロンとスペースを入力します。その後、@city 属性の直後にコレスを入力します(以下のスクレーンションを参照)。

<u>Odateline</u> Odate (コンテンツ) □ Odate :	Oplace = city (コンテンツ) (= city), = country
(コンテンツ) = country Oplace Odateline	

出力は以下のようています



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。デザイノこて、 Source: "とうテキストをsource 要素の開始タグ直前に入力します。

<mark>0 synopsis</mark> (コンテンツ) <mark>0 synopsis</mark> Source: <mark>0 source</mark> (コンテンツ) <mark>0 source</mark>

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トするコよ、目的のテキストをハイライトして、ローカルのスタイルプロ・ティを指定します。 デザイノこて入力した "Source: ~というテキストをハイライトします。 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。 以下のスクレーシショント にあるように、プロッティの font グループを展開して、font-style プロッティーこで italic オプションを選択します。 (*下のスクレーンション*)。

スタイル			×
la 🕂 📲 🛛			
2.94N : .	属性	値	
■ 1 テキスト	 ⊕ effects 		
	·⊞ IE		
	🕀 numbering		
	🕀 paged media		
	•⊞ UI		
	-⊞ XSL-FO		
	田 テキスト		
	田 テーブル		
	中 フォント		
	⊖ font		
	family		•
	size		
	style	italic	
	······ variant		
	weight		•
	田 ボックス		-

静的テキスト(ここでは "Source: ") かデザイノこてイタトック状態によいます:

Osynopsis (コンテンツ) Osynopsis Source: Osource (\Box) ノテンツ)く Osource

HTML プレビューの出力は以下のようてなります::



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、デザイノ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。

操作が終わす後に、ファイルの保存を行います。

このセグヨンでは、静的コンテンソの挿入とフォーマナ方法について学習しました。次のセグションではCSSを使用することにより、デザインコンポーネン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 コンテンソのフォーマト
- 前のセクション 動的なコンテンソの挿入
- <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 をベースは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レポーネントのフォーマッ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。
- 定義済みのフォーマナ

 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、コンポーネトのブループに対するラットとして 使用されます。これら定義済みフォーマナ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よ り、オーィ・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たのです。
- 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、<u>外部</u> おま<u>グロー・シル</u> 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)
 <u>グローブルスタイルレール</u>と(iii) <u>ローカルスタイルレール</u>となります。

このセクションでは、以下のことからを学習します。

- 定義済みのフォーマトを割り当てる
- コンポーネントにクラス属性を割り当てる
- <u>外部 CSS スタイルシートにてスタイルを定義して</u> そのスタイルシートを SPS のスタイルルポトリへ追加する
- 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レス対してローカルスタイルを定義する
- 単一のエレポーネントに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

<u>定義済みのフォーマト</u>を割り当てることで定義済みのフォーマトのスタイルを、継承とらかたちでコンポーネントへ与えることができます。デ ザイノこれで headline 要素を選択し、メニューオブションから「タグで囲む | 特殊な段落 | 見出し3 (h3)」を選択します。定義済 みのフォーマトタグが headline 要素の割りご作成されます (*下のスクリーンション*ト)。

<sup>
§</sup>h3 <mark>
<sup>
0</sup>headline</mark> (コンテンツ)
<sup>
0</sup>headline
<sup>
§</sup>h3
<sup>
</sup>

コンテンソのフォントプロノティが変化して、定義済みフォーマトの上と下に空白が追加されます。プロノティ値は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、ブロック内にあるコンポーネントをグループとしてデザイム、それらコンポーネン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 コンポーネントを割り当てます。デザインは以下にあるスクレーシンヨナのよう してひます。静的テキストの "Source: " が source 要素を含む div コンポーネント内に含まれるようになり、 newsitem 要素全体が div コン ポーネント内に含まれます。

§div Onewsitem
<mark> [§]div Odateline Odate</mark> (コンテンツ) <mark>■ Odate</mark> : Oplace = city (コンテンツ) = city , = country (コンテンツ) = country Oplace Odateline §div
<mark> §div</mark> <mark> Osynopsis</mark> (コンテンツ) Osynopsis §div
<u> Ŝdiv</u> Source: ○source (コンテンツ) ○source (Ŝdiv
Onewsitem \$div

div ブロック内とともに、コンポーネントがジループ化されました。このセクションの終盤 では、グループ化されたコンポーネントのブロックに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

スタイルはことポーネトのクラスに対して定義することができます。例えば、全てのヘッダーに対して(フォントファミリーや色といすご)共通したプロンティを定義することができます。定義を行うには、(i) 共通したプロンティを含んだことポーネントを割り当て、(ii) そのクラスに対してスタイルプロンティを割り当てる、という操作を行う必要があります。

デザイノにて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/ ディグループを選択します。共通プロ/ ディグル 一を展開している状態で、class プロ/デ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。

スタイル				X		
la 🕄 🟭	PATH	×		_		
スタイル: スタイル:	⊕ UI			▲		
■ 1 段落	⊕ XSI	L-FO				
E 1 コンテンツ	田 テキ					
	⊞ テーブル					
	田 フォント					
	田 ボックス					
	日共	<u> </u>				
	Ь	ottom				
	c	lass	header			
	c	lear				
	d	irection		•		
	d	isplay		•		
	fl	loat		•		
	le	eft				
	P	osition				
	ri	ight				
	te	ор		• •		

h3 フォーマトのインスタンスがheader とう名前のクラスへ割り当てられます。header クラスマオレてスタイルプロ ティを定義した場合、これ らのプロノティがheader クラスを持っている SPS 内の全てのエレポーネトレマオレて適用されます。

外部 CSS スタイルシートをスタイルレポントリへ追加する

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ためコよ、スタイルシートをスタイルルポントリへ追加することができます。 スタイルレポントリサイドバー (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. <u>スタイルポネリサイドバーのソールドー</u>コあ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¥Al tova¥Styl eVi si on2021¥Styl eVi si onExampl es¥チュートリアル \QuickStart\QuickStart.css を開きます。

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

.header	{	<pre>font-family: "Arial", sans-serif; font-weight: bold; color: red; }</pre>
h3		{ font-size: 12pt; }

header クラスとh3 要素のスタイルリ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 sandfiltering 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 センクターを使用することで、<u>グローイ シレスタイノリレール</u>をSPS 全体に対して定義することができます。 ルールは <u>スタイルルポネリサイ</u> ドバー内で直接定義します。 header クラスに対して、以下のようこグローイ シレスタイノリレールを作成します:

- 1. デザインビューがアクティブコンコンしる状態で、スタイルポントリサイドバー内のグローノシ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)。

スタイルリポジトリ	x
월 🖆 스 🗸 🔓 📲 👪 🚦 🏵 🛛	
田倡外部	-
= ○ ブローバル	
ロセレクター: .header	
-⊞ effects	
IE	
-⊞ numbering	
·⊞ paged media	
- ⊕ UI	
-⊞ XSL-FO	
·田 テキスト	
…田 テーブル	
─── フォント	
田 ボックス	
田 共通	
- 戶 色	
-⊞ background	
color 🛛 🔹 💮	
…⊕ 詳細	
	7

外部スタイルシートでも定義されているプロ・ティーンオレてグローッシレスタイルレートの定義を行うと、グローッシレレールのプロ・ティ値が優先的 に使用されます。そのため、以下のHTML プレビューではペッドラインのエレテンンが緑色で表示されます。外部スタイルシートにて定義された その他のプロ・ティーム、(グロー・シレ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 sandfiltering 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に対してローカルスタイルを定義する

複数のコ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 コンポーネントをクトックして、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 に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。 デザイノこて、dateline、synopsis、sourceノードを囲んでしる3つのdivコンポーネンを(Shift キーを押下した状態で各 div コンポーネントをクリンクすることで)選択します。 スタイルサイドバー にてび3 段落」を選択し、プロンティのフォントグループにて、font-sizeを10ptに指定します(h3 コンポーネントは必須項目のfont-size 12pt どなってしるため、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・ティのフォントグループにて、font-weight をbold にセントして、font-style をitalic に指定します。プロ・ティの色グループにて、color をgray 「指定します。 dateline の出力は以下のようてなります: 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・ティのフォントグループにて、font-weight をbold にセントして、font-style をitalic に指定しま す。プロ・ティの色グループにて、color をgray 「指定します。 dateline の出力は以下のようてなります:

2006-04-01: Boston, USA

div コンポーネトーマ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ーマ対しても適用されます。

単一の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ーンオーて定義されたローカルスタイルは、SPS内で定義されたとい高次のエンポーネトをオー・トライドします。デザイン ーて headline 要素を選択し、(スタイルプロ・ティの色グループ内にある color プロ・ティーンオルて) navy 色を割り当てます。ローカルで定義されたプロ・ティ(color:navy) により、header クラスマオルて定義されたグロー・シルスタイル(color:green) がオー・トライドされます。

source 要素を囲んでいる div コンポーネントを選択します。スタイルサイドバー にて「1 段落」アイテムを選択し、(スタイルプロ ティの色グ ループ ある) color プロ ティーこて gray をセルします。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 れらの値は精的なテキストに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:" に対して適用され たのを思い出してくたさい。新たなプロ ティ(font-weight:bold とcolor:gray) が font-style:italic プロ ティと 胡川に追加されます。

デザインビュートコン source 要素の(コンテン)パレースオルダを選択します。スタイルカラムコン「1 コンテンソ」を選択し、(スタイルパロレ 守ィの 色グループニある) color プロレ 守ィーこ black をセナーします。その後、プロレ ディのフォントグループ こある font-weight に対して normal を選択 します。新たコンプロ ディが source 要素のノードに対応したコンテンソパレースオ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レ ディがオー・バーライドされます(以下のスクノーンションを参照)。

フォーマトを完成する

このセクタインCフォーマトを完成させるこなさり、synopsis 要素のdiv コンポーネトを選択し、ツールレーの定義済みのフォーマト ボボックスこて 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スマオして適用されます。HTML プレビューイお以下のよう しています:

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

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

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オメクのパフフルなメカニズムです。このセクションでは、デザイメニネして、 XMLドキュメン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こよりカバーされた期間という、2つの情報を追加します。どちらの情報も XMLドキュメン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

デザイノこて、以下を行います

 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)。

Summary of News Items
Total number of items:
§div <mark>Onewsitem</mark>

- 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ル</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。<u>XPath 式 の編集ダイアログ</u> ダイアログが表示されも明日(以下のスクノーンショナを参照)。デザインを右クリックして、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。

<mark>と)</mark> XPath 条件式の編集				×
XPath 2 条件式:				
count(newsitem)				
構築 評価				
	演算子/式の選択	₹ 8	関数の選択	≡
322177 \$XML/presswatch/newsitems	□ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●		Accessors	
	田比較		⊞ Boolean ⊞ Constructors	
	⊡ ≂i⊤ ⊡ Quantified/for		田 コンテキスト	
□ [] \$XML (3√2)	田論理			
ロノルート要素	田数値			
O presswatch	田シーケンス		⊞ Miscellaneous	
			田数値	
			⊞ QName ⊡ Si⇔ http://	
			田文字列	
		-	⊞ URI	v
Ⅰ 希色对 XPath (<u>A</u>)	1	▶	<u>र</u>	Þ
Javaの使用Netの使用			OK (<u>0</u>)	キャンセル(<u>C</u>)

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ヘ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、次のセグションを参照し てくたざい XPath 式を編集。

) XPath 条件式の編集		
KPath 2 条件式:		
count(newsitem)		
構築 評価		
評価コンテキスト	結果	
(? (!- = Abc	• xs:integer 4	*
<pre>() presswatch = xsirnoNamespaceSchemaLocation="QuickStart) () selection () newsitems () newsitem [1] () newsitem [2] () newsitem [3] () newsitem [4] () newsitem [4] </pre>		*
選択されたコンテキストを記憶(R)	☑入力時とコンテキストの変更時(ご評価(T) 今すぐ評価(E)	
Javaの使用 Netの使用	OK(0)	ンセル(C)

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より自動計算をフォーマトすることができます。

Summary of News Items
Total number of items:=(自動計算)
§div Onewsitem

HTML 出力 は以下のようてなります:



ニュースアイテムこよりカバーされた期間を表示する

ニュースアイテムことりか、そ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と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 ノードのコンテンンが単純に出 力され、2番目の条件式により date ノードにある年や月151か出力されるよう」されます。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)))

デザイノこて、以下に示されるよう、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。

- ヨ リンク
- 次のセクション 条件を使用する
- 前のセグション、コンテンソのフォーマト

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 ノードには 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とらノードかあります。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 コンテンクを含むrelevance ノードを持すと全てのニュースアイテムが表示されます。条件により(ノードをルックアップすることで)byrelevance ノ ードのコンテンガケストされ、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なります。このセクションでは、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。
- 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。
 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。

ユニークなノード値を選択するコンボボックスを作成する

XMLドキュメイでは ユーザーの選択が/presswatch/selection/byrelevance 内に含まれます。コンボボックスとてこのノー 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。

🛐 コンボボックスの編集	×
以下の要素に対して:	
○ スキーマ列単を使用	
○ 値のリストを使用	
表示されるエントリー XML 値	
● XPath 条件式を使用	
● 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	
○ XML 値ならびに閲覧可能なエントリーに対して異なる XPathを使用	
XML 値ならびに開覧可能なエントリーに対する XPath distinct-values(//relevance)	
編集	
■ Authentic 内の値を並べ替え	
OK +>>セル	

- 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:

Total number of item Period covered by ne	s:=(自動 ws items	計算) x =(自動計)	〕)	
Select by relevance:	🖾 \$XML	Opresswatch	Oselection	O byrelevance
=(×Pəth) 🔽 🚺	byrelevance	Oselection	Opresswatc	h 🕒 \$XML

6. <u>HTML プレニー</u> ペ小 潜えます。コンボドックスの矢印をクックすると、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。

Total number of iten Period covered by ne	ns: 4 ews items: 4/2006 to 5	/2006
Select by relevance:	NanoPower 🔹	
	NanoPower	
	NanoNull	
	Stockmarket	
NanoNull Inc Laur	NanoDiamonds	of Nano

条件を挿入することで選択された relevance し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 下のようで条件を挿入します:

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)。

§div Onewsitem
^{Sh3} Oheadline (コンテンツ) Oheadline Sh3
Sdiv Odate のdate ロンテンツ<
[§] div <mark>Osynopsis (コンテンツ) Osynopsis</mark> [§] div
§div <i>Source</i> : <mark>O</mark> source (ロンテンツ) <mark>O</mark> source §div
Onewsitem §div

- 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。
- 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)。

§div Onewsitem ? 1
[§] h3 <mark>Oheadline (コンテンツ) Oheadline</mark> §h3
Sdiv Odate Odate Odate Country Country <td< td=""></td<>
[§] div <mark>Osynopsis (コンテンツ) Osynopsis</mark> [§] div
§div <i>Source</i> : <mark>0</mark> source (ロンテンツ) <mark>0</mark> source (§div
Onewsitem Sdiv

この段階では条件分岐がり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。このセクションの後半では、2番目の条件分岐を追加します。

relevance ノードをリストとて挿入する

各 newsitem 要素のrelevance ノードを表示するためコよ 以下の操作によしれらノードをデザイン内に挿入します(以下のスクノーンショナを参照):

- 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. <u>スキーマソノーサイドバー</u>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。
- 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:

§div Source: <mark>()</mark> source (ロンテンツ) <mark>()</mark> source (§div
Sdiv Relevance:
Onelevance Orelevance Orelevance Onelevance
Onewsitem Onewsitems Opresswatch

ここでHTML プレビューーこて、コンボボッ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ドキュメノト内のbyrelevanceノードの値を変更する。(ii) XMLドキュメノトを保存する。(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をハイラールすることで、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。
- =/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リックします。
- 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た条件が作成されました: relevance のエレテンソ が/presswatch/selection/byrelevance のコンテンソニマッチすればrelevance のコン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トでコンテンツが表示されます。HTML プビューインで動作を確認してくたさい。

コンボボックスを修正して、2番目の条件分岐を挿入する

コンボドックス内の、全てのエ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. コンボドクスの編集 ダイアログにて、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)。(アのスクリーンショット).

Total number of new Period covered by ne	rs items: 4 ews items: 4/2006 t	o 5/2006
Select by relevance:	All 🔹	
	NanoPower NanoNull Stockmarket NanoDiamonds All	

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17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ノードにて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た場合に、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たさい

- 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. <u>XPath 式 の編集ダイアログ</u>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が含まれます。 byrelevance ノードのエンテンがAII の場合、2番目の分岐にはエ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。

- = リンク
- 次のセクション. グロー・ジレテンプレートとその他のエレテンソの使用
- 前のセクション 自動計算の使用
- <u>チュートリアルスタートページ</u>
- <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

<u>グロー・シレテンプレート</u>を使用することで、グロー・シレな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・シレテンプレートを、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こさります。 グロー・シレテンプレート 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・シリテンプレートを呼び出すコス 2つのメカニズムかあります:

- 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。

このセグションでは、その他のコンテンツ処理命令とグローバリ要素のmember しつするグローバ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 スキーマの大おな構造を以下のスクレーシショナに示します。



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います。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 要素だけが処理されます。selection な らいこ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 ならいにteam 要素が処理されるよう」ないます。 newsitemsとpresswatchの終了タク間にカーノルを配置し、メニューコマドまけコンテキストメニューから挿入 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コンテンン処理命令が挿入されます。その他のコンテンソプレースやいダが挿入されます(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(以下のスクレーンションを参照):

AllAndrewBentincka.bentinck@nanonull.comNadiaEdwardsn.edwar

内蔵のテンプレートを selection ならびに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が出力されます。 selection/byrelevance かみられたAll とう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 キュメント内の順序にと出力されます。member のid 属性は出力されて、ないてとご注目してくたさい(属性は member の子とは見なされ ません)。

selection に対してグロー・ ジレテンプレートを作成する

selection のコンテンルを出力に含める必要は無しため、selection に対して空のグロー・シリテンプレートを作成し、そのコンテンツが処理され ないようします。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 W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)。

selection (単一)
0selection (コンテンツ) 0selection

- 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
- 3. フロ ハルアンツ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・シレテンプレートが空になっているため、selectionの子要素が処理されなくなります。

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イ シリテンプレート内にテーブルを作成します。以下の 操作を行なってください

- 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
- team 安素に対するグローバルテンプレートを作成します(スキーマグリーサイトバーにのる主 Cのグローバル安 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)。
 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。
 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)。

動的テーブルの作成	×
member	の各アイテムに対して
テーブル拡張方向 上がら下へ(王) 左がら右へ(L) 	 ヘッダークッター ✓ ヘッダーを作成(日) □ フッターを作成(F) ☑ 数値フィールドの要約(8)
セルの表示方法 ③ コンデンツ ③ コンポポックス ④ 編集フィールド ④ 複数行入力フィールド ③ チェックポックス ③ ラジオポタン	行力力な選択 <member> @id first last department email telephone</member>
	ОК 4 +>ンセル

- 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、プロ・ティサイドバーのテーブルが選択されている状態で、以下のスクレーン ショナに示されているようにテーブルのプロ・テクを変更します。

プロパティ		×
La 🗜 🖑	PATH ×	
プロパティ:	属性	[值] 🔼
テーブル	田 テーブル	
	🕀 Authentic	
	HTML	
	align	
	bgcolor	💌 😲
	border	1
	cellpadding	5
	cellspacing	0
	······ datafld	
	dataformatas	•
	datasrc	
	······ frame	•
	······ height	
	······ rules	V
	summary	
	width	
	田 イベント	
	田 インタラクティブ	
	田 共通	
		*

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、ヘッダー行ヘ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

- = リンク
- 次のセクション、おま!
- 前のセクション 条件を使用する
- <u>チュートリアルスタートページ</u>
- 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しています

- ドキュメントの構造を作成する方法、(メインのデンプレートとグロー・シリテンプレート)。
- <u>動的</u>と<u>静的</u> SPS コンポーネントを使用して、動的ならび、静的なコンテンンをデザイン内部へ挿入する方法。
- <u>外部スタイルシート、グロー・シルスタイルレール、とローカルスタイルレール</u>にて、CSS スタイルを使用する方法。
- <u>自動計算</u>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クや、XML から得られたデータン従った出力を得る方法。
- <u>グロー・ ジレデンプレート</u> と<u>その他のエレテンツ</u>を使用する方法。

これら機能の詳細な説明については、以下にあるセクションを参照ください

- <u>SPS ファイル: エレテンソ</u>
- <u>SPS ファイル 構造</u>
- <u>SPS ファイル: 高度な機能</u>
- <u>SPS ファイル プレゼテーション</u>
- <u>SPS ファイル 追加機能</u>

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

- 次のセクション コレテンソのフォーマト
- 前のセクション 動的なコンテンソの挿入
- <u>チュートリアルスタートページ</u>

4 使用方法の概要

目的

StyleVision で作成されたSPS ドキュメン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 ドキュメン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:

- 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。スキーでおよ(i) スキーマアイル(DTD おけまXMLス キーマ)、(ii) DB から生成されたXMLスキーマ(Enterprise ならび)こProfessional Edition のみ)、(iii) XBRL タケノミ をベースコンたスキーマ(Enterprise Edition のみ)、(iv) ユーザ 定義スキーマ(StyleVision にて直接作成)の、ずれかけな ります。この操作はデザク、概要サイドバートコントできます。おけよ、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 <u>使用するXSLT の ジンを選択します</u>。
- 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 のデジョンに合致するようします。
- 5. <u>デザインビュー</u>では、様々なデザインコンポーネントにより SPS ドキュメントをデザインすることができます。<u>デザインの作業</u> はギキュ メント構造の作成と、<u>表示プロンティ</u>の定義から構成されます。
- 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が必要とされる場合、それらのファイルを生成します。
- 三 関連項目
- <u>SPS ファイル構造</u>

4.1 SPS とソース

新規 SPS ファイルを作成する

SPS ドキュメトを新たった成するコま「ファイル 新規作成 (Ctrl+N) 」コマド以下のオプタンを選択するか、標準ツール くに

ある新規デザインアイコン します。新規ドキュメト には、SPSX.sps とう仮の名前が与えられ、X にはアプリケーションの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 ドキュメン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かあります。

SPS I文化で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ノースファイルは以下の2 種類あます:

- <u>_____</u>
- <u>作業用 XML ファイル</u>

これらソースファイルの割り当ては、<u>デザイン概要サイドバー</u>にて行うことができます。割り当てを行う方法についてしよ、<u>デザイン概要</u>のセク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

デザイボキュメトの構造を作成するコよ SPS ヘ少なくとも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 <u>デザイン概要サイドバー</u>内で割り当てられます。スキーマは、XML スキーマファイル(.xsd ファイル)、XML ファイルより生成された XML スキーマ、DTD、おゴよ、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、<u>作業用 XML ファイル</u>を割り当てることができます。 <u>作業用 XML ファイル</u>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。 <u>作業</u> <u>用 XML ファイル</u>は、<u>デザイン概要サイドバー</u>にて割り当てられます。

- 関連項目
- <u>______</u>

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>エレテノソ</u> (XML ドキュメントまたは DB から得られた)のた どれが出力されるか。更に、エレテンンを直接 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. 出力の様々なパーソス対して適用される表示(フォーマナ)プロ、ティの内容です。

出力のエンテンソ

出力のコンテンソお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 スキーマソノーサイドバー内の関連するスキーマソノーから目的の XML データノードをド ラッグし、SPS 内の目的の場所にドロップすることで、XML ドキュメント から得られたコンテンンが SPS に含まれます。
- 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。
- 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 自動計算によ行われます。
- 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<u>出力の構造</u>を指定することができます: (i) XML ドキュメトの構造とば独立した入力レ <u>ベルテンプレート</u> (Style Vision の<u>メインのテンプレート</u>) にて出力構造を指定、(ii) (Style Vision の<u>グロー・シレテンプレート</u> 等の) <u>様々なノードに対してテンプレートルールを宣言し</u>(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<u>メインのテンプレート</u> と <u>グロー・シレテンプレート</u> 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 <u>モンュールSPS</u> と<u>デザ</u> <u>インテグメトト</u> 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(フォーマナ)

デザインビューでは CSS スタイルにとし表示プロ・ティカデザインコンポーネントへ適用されます。スタイルの定義はコンポーネントに交してロー 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 セノクターに マガして行うことができます。 更に 定義済みのフォーマナ を使用することで、特定の HTML 要素をコンポーネントに 文刊して適用す 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、<u>表示の方法</u>のセクションを参照ください。

- □ 関連項目
- <u>SPS ファイル構造</u>
- <u>デザイビュー</u>

4.3 XSLT と XPath のバージョン

SPS とはXSLT スタイルシートと同様の機能を持っています。各 SPS に対して 1.0、2.0、おは3.0のXSLT バージョンを指定する 必要があり、適切なソール デーアイコンの 10 おは 10 おけま 10 をクリックするこ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、おは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 が使用されなか、(XSLT 1.0 で使用される)XPath 1.0 おは(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) *ヘターフェー*スて <u>出力ビュー</u>を生成するため、そして(ii) <u>ヘクターフェイス内部</u>と <u>StyleVision Server</u> からHTML)の<u>出力ファイルの生成と保存</u>を行うために使用されます。変換に使用されるXSLT エンジン(Altova XSLT 1.0、2.0、おけは3.0 エンジン)の いジョンは、SPS で選択されたの いうコンには決定されます。

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 のビ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 StyleVision 内の様々な場所 -特に自動計算 と<u>条件テンプレート</u>ー 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 のビ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しておします。

- 三 関連項目
- <u>生成されたファイル</u>

4.4 Internet Explorer の互換性

SPS デザインを(デザインビューーこて)正確に表示し、プレビューを(HTML プレビュー)出力するコよ Style Vision をお使いのエレピューター 上にInternet Explorer (IE) をインストールする必要があます。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 おけお以降。
- 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-StyleVision 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、<u>イメージ型 と出力</u>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 出力に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、<u>ユーザー定義要素</u>と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

プロハティダイアログにて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。

 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、定義済みのフォーマナ おはユーザー定義要素 とて入力 することができます。HTML 要素は

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」コマンドおけま 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。
- <u>作業用 XML ファイル</u>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。

出力のマークアップは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。マークアップとデ ータは XSLT スタイルシートによと結び付けられます。StyleVision の<u>出力ビュー</u>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: 🔗 <u>XML レポート</u>

生成されたギュ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 とイアナ は Style Vision 内部で確認す ることのできる ほぼ同一のもの (ないます。(2) HTML ファイル内のデータ入力デ・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おりません。これらデータ入力デ・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。

三 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ナ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 (や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、(ネトワークに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の変更に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 におっちかのグ機能は以下のよう、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か含まれる RootCatalog.xml といろ名前のファイルかロードされます。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 ログファイルにて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%変数には識別されたフォルダー以下のSchemasならいにXBRLフォルダー内に catalog.xml とう名前のカタログファイルが存在する点に注目してください、%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 タクノノミの公開ならび、ミンステム識別子から、ローカルに保存された対応するスキーマのコピーか指し示されます。これらスキーマは、StyleVision のインストール時にAltova Common フォルダーヘインストールされます。エラーを起こす可能性かあるため、これらファイル内にあるマノピングを複製してれいうご言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 にて、CoreCatalog.xml ならびにCustomCatalog.xml か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 ならいこCoreCatalog.xml ファイルはStyleVision アプリケーションフォルダーヘ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 ならいこ%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より、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ケトップフォルダーへのフルマス
%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な
%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ペ
%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ペ
%TemplateFolder%	現在のユーザーのテンプレートフォルダーへのフルペ
%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マス
%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ペ
% CommonAppDataFolde r%	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な
%FavoritesFolder%	現在のユーザーのお気に入りフォルダーへのフルマス
%PersonalFolder%	現在のユーザーの個人用フォルダーへのフルマ
%SendToFolder%	現在のユーザーのSendTo フォルダーへのフルス
%FontsFolder%	システムフォントフォルダーへのフルクス
%ProgramFilesFolder%	現在のユーザーのProgram Files フォルダーへのフル &
%CommonFilesFolder%	現在のユーザーのCommon Files フォルダーへのフルタ
%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な
%LocalAppDataFolder%	ローカル(非ローミング)アプリケーションコンオするデータレポネリとして機能するシステムフォルダーへのフルマ
%MyPicturesFolder%	ピクチャライブラリ、マイピクチャシフォルダーへのフルダ

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 ヘリダイレクトされます。カタログファイルにより、公開ませるよン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、公開ませるよステム識別子 が、カタログファイルによりローカルリノ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ピングすることさけが必要にないます。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か読み取られます。例えば、以下にある SVG ファイルを考えてみましょう

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

```
<svg width="20" height="20" xml:space="preserve">
    <g style="fill:red; stroke:#000000">
        <rect x="0" y="0" width="15" height="15"/>
        <rect x="5" y="5" width="15" height="15"/>
        </g>
</svg>
```

このドキュメン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は以下のエントリーか含まれます:

<catalog>

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

</catalog>

この場合、PUBLIC 識別子に対応するマッチが存在するため、SVG DTD へのルックアップがschemas/svg/svg11.dtd URI ペダイレナ され(この なはカタ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よ場合、XMLドキュメント内の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ドキュメナイのシップレベンドキュメナー要素内にある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、(上の列にて太字で示されている)属性値のURI には、実際に存在するスキーマの位置か記述されます。カタログによりスキーマか参照された場合、実際に存在するXML スキーマを指し示す必要に進しものの、xsi:schemaLocation 属性の妥当性を満たすためにつらかの値を入力する必要があります。属性のURI 部分には、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(上の列における http://www.altova.com/schemas/orgchart)により、カタログ内にスキー マを配置することができます。カタログでは、以下のエン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ナ

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)にてエト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」を使用するようし、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。よぼ細な解説に

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ない場合、system要素によりシステム識別子をURLへ直接マンングすること ができます。更にuri要素を使用することで、URIを他のURIへマンピングすることもできます。rewriteURIならひにrewritsSystem要素に より、それぞれURIやシステム識別子の書き換えを行うことが可能しなります。この機能により、使用されるファイルレジを変更し、ターゲナの フォルダーを変更することが可能しなります。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。

- 三 関連項目
- <u>生成されたファイル</u>

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てい、一変更のインクニクタログファイルURI のみか必要 ーコン るイン ハー処理 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オーダンフタログメカニズムを使用 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、オニューザーには「拡張可能です。セクション <u>Style Vision 内のカタログ構造</u>と力タ ログのカスタム化内で説明されています。セクション <u>Windows システムロケーションのオーダの変数</u> は共通のシステムのロケーションのオーダに 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オーダンニエれらの変数に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

- <u>カタログの仕組み</u>
- <u>StyleVision 内のカタログ構造</u>
- <u>カタログのカスタム化</u>
- Windows システムロケーションのための変数

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 スキーマをリ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エ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 にダイレクトするオムタン使用されます。これはマンピング、カタログファイル、ローカルURI によ り必要とされる、ブリックます」おシ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、ハブリックます」よン 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ンピングによりロケートします。

人気のあるスキーマイ理しては PUBLIC 識別子は通常定義済みです。このナダカタログファイル内のURI のみか正し、ローカルコピーイス対して PUBLIC 識別子 をマップします。XML ドキュメト か解析されると内部の PUBLIC 識別子が読み取られます。この識別子がカ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 がリックアップされ、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 StyleVision 内で開かれるとします。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="20" height="20" xml:space="preserve">
<g style="fill:red; stroke:#000000">
<frect x="0" y="0" width="15" height="15"/>
<rect x="5" y="5" width="15" height="15"/>
</g>
<//svg>
```

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ントレを含んでいます:

<catalog>

</catalog>

この場合 PUBLIC 識別子のかめに一致するものかみます。この結果 SVG DTD のかめのルッグアップがカタログファイルは相対的な) URL schemas/svg/svg11.dtd にダイレオされます。これは SVG ファイルのかめに DTD とて使用されるローカルファイルです。 カタログ内で Public ID のかめのマピングが存在しない場合、XMLドキュメント内の URL が使用されます(上の SVG ファイルのサン プルの場合、これはインターネット 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リの要素の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属性の値はおつの部分が決ます:(上で緑色で表示されている名前空間のペー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ソースをマップするためにカタログ内で使用されています。例えば以下のカタログエストリは上のスキー マレファレンスを代替のロケーションのスキーマイニリダイレクトします。

<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 やです。しかしなから、スキーマかガ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ります。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)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(W3CXMLスキーマとSVGスキーマう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ルを参照します

• CustomCatalog.xml は (変数 % Personal Folder % によったートされる) 個人用フォルダー にあます。自身のマピ

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により指されていない必要とするスキーマのため にマピングを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(次のセクションを参照)。

- (共通に使用されるスキーマのセルを含む変数 %CommonSchemasFolder%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コピーをポイントするURI に対する プリックおよび/まけよン ステム識別子をマップする catalog.xml ファイルです。
- CoreCatalog.xml はStyleVision アプリ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 を生成するために使用されるStyleVision パワ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ために使用されます。

<u>ロケーション変数</u>

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、上のバスティング:

%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

カタログファイル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アプリケーションフォルダー内におます。
- CustomCatalog.xml はMyDocuments\Altova\StyleVision フルダー内はあます。
- Catalog.xml 入るよそれそ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(おけましにいいの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たさい。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) を参照してくたさい。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/ ジリック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ませよステム識別子の開始の部分をそれそ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マの開始部分が置き換えられこの結果他のディノオリをターゲナ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。

リース2014 以降StyleVision はXML Catalogs specification (OASIS Standard V1.1, 7 October 2005) 仕様を厳密 は順守して、ます。この壮様は厳密に、「ジトクID お子はシステムID) 外部識別子ルッグアップを(、ジトクID お子はシステムID を 持たされ、URI) URI ルッグアップと区別して、ます。名前空間 URI は単純に、ジトクID お子はシステムID ではなく URI と考えられ外 部識別子ルッグアップではなく URI ルッグアップとして使用されるへきです。、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> マメニングを使用して訳されてきました。2014 以降から <uri>マメニングが使用されて、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 か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ドキュメント内で参照されるスキーマ(以下に表示されています)。 xsi:schemaLocation 属性の値ははつの部分があります。(緑色で表示されている名前空間のパートといううたされている) URI パート。

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

以下はStyleVision か参照されたスキーマを検索するステップです。スキーマは最初のステップでロードされます。

- 1. xsi:schemaLocation 値のURI 部分のオダ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が 検索されるとスキーマのロードのオックに結果のURI を使用します。
- 2. xsi:schemaLocation値の名前空間部分のナダ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 グが検索されるとスキーマのロードのナダン結果の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StyleVision ビントされておりXML スキーマ(.xsd)ドキュメ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 スキーマドキュメト内では、このためレフォンスはXML スキーマ仕様を定義するスキーマイズってしないません。

%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 へのレフルシスを含んできます。XML スキーマドキュメートをこれらのスキーマイズれて検証してはひょうません。参照されるファ イルはドキュメー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 に入力ヘルレー情報を与えるセッジに含まれていま す。

4.7.4 Windows システムロケーションのための変数

nextCatalog 要素ではシェル環境変数が使用され、システム、関係する場所へのやか参照されます(上のRootCatalog.xmlののの容を参照)。以下のシェル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 次 例えばC:\Users\<name>\Documents

```
%
CommonSchemasFol
der% C:\ProgramData\Altova\Common2021\Schemas
%
ApplicationWrita
bleDataFolder% C:\ProgramData\Altova
```

olo	
AltovaCommonFold er%	C:\Program Files\Altova\Common2021
%DesktopFolder%	現在のユーザーのデスケトップフォルダーへのフルタ。
% ProgramMenuFolde r%	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マ。
%StartMenuFolder %	現在のユーザーの開始メニューフォルダーへのフルマ。
%StartUpFolder%	現在のユーザーのスタートフォルダーへのフルタ。
%TemplateFolder%	現在のユーザーのテンプレートフォルダーへのフルマ。
8 Nelmin Maala Dalalan	
%	現在のユーザーの管理ツ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クス。
%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レス。
% CommonAppDataFol der%	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ークのフルマ。
%FavoritesFolder %	現在のユーザーのお気に入りフォルダへフルマ。
%PersonalFolder%	現在のユーザーの個人フォルダーへのフルマ。
%SendToFolder%	現在のユーザーの送信フォルダーへのフレッマ。
%FontsFolder%	システムフォントフォルダーへのフルマ。
% ProgramFilesFold er%	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マ。
8	
CommonFilesFolde r%	現在のユーザーの共通ファイルフォルダーへのフルペ。
%WindowsFolder%	現在のユーザーのWindows フォルダーへのフル や。
%SystemFolder%	現在のユーザーのシステムフォルダーへのフルマ。
% LocalAppDataFold er%	ローカル(非ローミング)アプリケーションのオーダ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レクトリ へのフルマ、
% MyPicturesFolder 2	⊬ ² 7∓+77+11&ั→~ก\7111 งัว
U	

5 SPS コンテンツ

このセクションでは、XML データエンテンツのデザインを構成する SPS ドキュメントコンポーネントの作成ならびに編集を行う方法について説明します。様々な操作方法が以下のとお記述されており、それそれのセクション以下にあるサブセクションにご詳細が記述されます。これらメ カニズムはメイン、グローバル、おけは名前付きテンプレートをデザインするために使用されます。

- <u>テキストとて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(その他のコンテンツ)として作成します。
- MS W ord エレテンソの挿入
- <u>ユーザー定義テンプレート</u>
- <u>ユーザー定義要素, XML テキストプロック</u>
- <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ドキュメン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、ターゲオ 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 デザインへ挿入することができます。
- データ入力デバイス(おコュフォームエトロールの使用:入力フィールドやエレボボックスといオーデータ入力デバイスにより、 Authentic View ICT XML データを入力することができます。この機能により、ユーザーへのへい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ノードをデータ入力デバイスとして作成することができます。
- リンク
- レイアナ モジュール
- <u>変更機能</u>: 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、別のエ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
- <u>SPS ファイルの高度な機能</u>
- <u>SPS ファイル追加機能</u>

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レーウィンドウカシドラッグし、デザインコン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、デザインは挿入することができます。スキーマノードをデザインインドロップすると、デザインコンこのノードをどのようご作成するのか選択す るけっかのメニューカ表示されます(以下のスクレーシショントを参照)。

コンテンツの作成(C)
テンプレートの作成(E) ^{KS}
段落の作成(P)
テーブルの作成(工)
CALS/HTML テーブルの作成
箇条書きの作成(№)
バーコードの作成
チャートの作成…
イメージの作成(<u>G</u>)…
入力フィールドの作成の
複数行フィールドの作成(<u>M</u>)
リッチエディットの作成(<u>R</u>)
コンボ ボックスの作成(<u>O</u>)…
チェック ボックスの作成(出)
ラジオ ボタンの作成(<u>R</u>)
ボタンの作成(B)

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ヘドロップすることか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、ずれかけないます。

コンテンソの挿入ツールレーアイコンを使用する

<u>デザイン要素の挿入 ツール デ</u>アイエンを使用することでも、デザイン内にあるノードのエンテンンを挿入することができます。以下の方法により エンテンン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・*大)。
| 🚯 デザイン要素 コンテンツ を挿入 | | |
|---|--|--|
| ■ 新規デザイン要素 コンテンツ のソースとして使われることになる XML または DB ノー
修選択 | | |
| ソースデータを後で選択したい場合は、このチェックポックスのチェックを外して、適切な時に「テンプレートで囲む」メニューを使用してください。 | | |
| XPathコンテキスト: \$XML/presswatch/newsitems/newsitem/metainfo/r | | |
| スキーマの属性または要素を選択: | | |
| ● ソース
□ () \$\$\$ \$\$\$ \$\$\$ \$\$\$\$ \$\$\$\$
□ () presswatch
□ () selection
□ () newsitems
□ () newsitem
○ headline
□ () headline
□ () dateline
□ () source
○ synopsis
□ () metainfo
□ () team | | |
| ■ 絶対 XPath OK キャンセル | | |

- 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。
 「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
- 「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:



上のスクレーシショナでは、Desc 要素がシンテンソとて作成されました。出力ではDesc のテキストコンテンパ、振示されます。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パ、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サム、属性ノードのコンテンパ、Desc コンテンパの一部として出力されることはありません。属性内のコンテンツ 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、それらが出力になります。.

fl cData This is CDATA Text. **fl 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

このセグョンのサブセグョンでは、XML コンテング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は値接マークアップする。
- 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デンツ.
- 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。

三 関連項目

- デザインビューシンボル
- <u>テノプレートとデザインプラグメント</u>

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の定義済みのマークアップとともに、直接挿入することができます。スキーマソノーウィバウから目的の場所にノードをドラッグアイドロップして、表示されるメニューから段落の作成を選択します(以下のスクノーンショナを参照)。

1	コンテンツの作成(<u>C</u>)	
	テンプレートの作成(E)	
	段落の作成(P)	
	ーー・レート テーブルの作成①…	

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、おゴオメニューコマボの挿入 | 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。

書式		▼ ×
標準	- B I <u>U</u> S	• • • • • • • • • • • • • • • • • • •

プロ・ティウイドウ内にある段落グループの段落タイププロ・ティの値を変更、おゴコレードテンプレートの<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トを変更することができます。

各種の段落には、それそ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。

三 関連項目

- デザインビューシンボル
- 定義済みのフォーマト

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市り刻ま、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 扫が処理されます。child2 ならびにchild3 子要素が処理されることはかません。parent の要素に対してその 他のエンテンソプレースポ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 扫すでは 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・シレテンプレートが定義されている場合、グロー バリテンプレートが使用されます。そうでは、場合、(要素、属性、テキストノードに対して使用される) 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 その他のコンテンソニオしてどのノードが選択されているか留意する必要が決ます。

- 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、その他のエレテンソプレース市ルダにより選択されます(XML ドキュメン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・ジレ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、その他のエンテンソプレースオルダーにより子要素も選択されること」ないます。しかし、子要素のグロー・ジレテンプレートが「ローカルのエピー」となっている場合、ローカルで使用されたことしていく、子要素がその他のエンテンメにより選択されることはありません。
- **メモ 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。

三 関連項目

- <u>テキストとしてXML コンテンンを挿入</u>
- <u>デザイビューシンボル</u>
- <u>テンプレートとデザインプラグメント</u>

5.2 MS Word コンテンツの挿入

Microsoft W ord 2007+ がマシイニインストールされていると、コンテンンをW ord ドキュメトから、デザインに静的コンテンツとして貼り つけることができます。W ord コンテンソス、適切に対応するデザインコンポーネント内挿入され、テキスト書式プロパティは、W ord コンテ ンソから移動されます。ます。例えば、W ord 段落プロック内のテキストコンテンソス、<u>パラグラフのコンポーネント</u>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ント参照*)。

Accelerate XML Development

Today, <u>eXtensible</u> 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



メモ Word コンテンソコルえ、Word ドキュメトとして貼り付けることのできるコンテンソよ、Style Vision デザイン内は比けけることができます。これは、MS Excel テーブルとHTML ページコンテンツを含みます。

メモ W ord ドキュメト全体が静的コンテンを含む 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 コンテンガデザイノニピーして貼り付けられた場合サポートされます

- · フォーマトされたテキスト
 - 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
- ひページ
- 水平線
- ハイパートンク
- ブックマーク
- テーブル
 - 行の範囲、列の範囲
 - フォーマナ済み//ッチコンテンツ
 - ネストされたテーブル
 - · ヘッダー、フッター
- リスト、サブリスト
 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+ がマンイニインストールされている場合、コンテンンを Excel ドキュメトからデザイン「静的コンテンソと」て貼り 付けることができます。 Excel コンテンソよ 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 とて挿入されます。書式 プロパ ティイは保管されます、 下のスクレーンショント参照。。各 Excel シートは、個別の静的テーブルとして挿入されます。

	А	В	
1	09.03	Euclid's Elements	
2	11.09	glish 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)ED	
9	07.89	owler's Modern English Usage	
10	05.30	xford Guide to Plain English	
11	21.88	lediscover Grammar	
12	11.97	low to Solve It	
13	21.54	Advanced Learner's Grammar	
14	19.02	Aacmillan 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 ジート

	-	▼
Þ	09.03	Euclid's Elements
Þ	11.09	English Phrasal Verbs in Use
Þ	08.86	Code Book
Þ	09.80	Foundations and Fundamental Concepts of Mathematics
Þ	18.38	Style
Þ	08.72	The English Language
Þ	18.36	History of Mathematics
Þ	06.09	QED
	07.89	Fowler's Modern English Usage
Þ	05.30	Oxford Guide to Plain English
Þ	21.88	Rediscover Grammar
Þ	11.97	How to Solve It
Þ	21.54	Advanced Learner's Grammar
Þ	19.02	Macmillan English Grammar in Context
Þ	22.57	Oxford Style Manual
Þ	16.44	Proofreading
Þ	20.51	Taschenbuch Mathematischer Formeln und Moderner Verfahren
Þ	97.50	Oxford Companion to the Book

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 コンテンソコルえ、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、MSW ord コンテンツとHTML ページコンテンツを含みます。
- メモ Excel ドキュメント全体から静的コンテンンを含む SPS を作成する コス ファイル 新規作成 | Excel 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ニ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
 - フォーマオ済み/リッチコンテンツ
 - ネストされたテーブル
 - 0 **^~/&____ 7/&__**
- · 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

•

- ハイパートンク
 - リスト、サブリスト ○ *箇条書き: 異なるスタイル* ○ *列挙: 異なるスタイル*
- イメージ

5.4 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に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おとし、一ドとなります。以下のスクリーンショナトでは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か示されます。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。

▲ /n1.tName (コンテンツ) ▲
👼 for \$i in n1:Office[Loc=NY] return \$i[1]
▲ <u>1,2,3</u> →=(自動計算) ▲

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交化て定義された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、全てのコ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にないます。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ム、表示されます。
- XPath 2.0 とXPath 3.0 のfor コンストラ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 内でエンテンソプレース市以多を使用することはできません。これは、(エンテンソプレース市以外には少生成される)xsl:apply-template 処理命 令が、(原子型の値ではなく)ノードアイテムはしか適用できないためです。そのかわり、箇条書きといたデザイン要素と生に、自動計算を 使用することは可能で、以下のスクノーシショント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、そうでない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)には選択することで、XPathのパセレ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になり、ノードの選択に対してフィルタレクや条件をセルすることが可能にないます。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にないます。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 お が 選択されます。 このセクション内にあるその他の例も参照くたさい。

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 スペステップ内に活弧が使用されているかがブ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 select="Dept"> </xsl:for-each> </xsl:for-each> </xsl:for-each></pre>	各 Office 要素には独自の Dept が含まれ、グループ化や 並び替えしよ 各 Office 内部で 行われる。
有り	<xsl:for-each select="/Org/Office/Dept"> </xsl:for-each 	Dept 要素が 全てのOffice ならびこ 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

- デザイン要素の挿入ツール・トーンあるユーザー定義テンプレートを挿入アイエンをクリックし、デザイン内のテンプレートを挿入する場所を クリックします。まけは、テンプレートを挿入する場所を右クリックし、コンテキストメニューカ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 をクリックます。 XPath 条件式のエンテキストノードは、クリックされたノードになる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、ターゲル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) 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、(結合まだはよう割により) 返された両者のノードセナトに違いながりません。 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、: ノードテンプレートオペレーションを参照してくたさい。

テンプレートマッチの編集

任意の、ユーザー定義まけは通常の)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よ、ノードテンプレートを右クリックレ、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カ表示されるので、新たなノードを選択するXPath条件式を入力し、「OK」をクリックします。

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LVードおけおノ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、

スキーマソレ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、新たなノードのエレ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)相対パマで)参照されるのか、おさま(ii)スキーマノースのルートからのフルル マにより参照されるのか決定 するよう促されます(以下のスクレーンションを参照)。

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なっています。「ソール」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: エレテンソ
- <u>ノードテンプレートオペレーション</u>
- <u>ユーザー定義要素</u>

ユーザー定義要素、XML テキストブロック 5.5

ユーザー定義要素とユーザー定義 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 シナカら得られた 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しなり、(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 オブシェケ)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 かみます。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。コンテ 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 メモ 載された Authentic View でのみサポートされます。

□ 関連項目

- <u>ユーザ-定義要素</u>
- 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は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XSLFO)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。

灹 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

- 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。
 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。
- 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」を選択します。
 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。

ユーザー定義要素の編集		
要素の名前と、二重引用符(ダブルウォート)で囲まれた属性の繰り返しを入力してください。		
例えば、HTMLにて50%の幅を持った「水平線」要素を入力するには:		
hr width="50%"		
動的属性の値に対しては、括弧で囲まれた XPath 条件式を入力します:		
hr width="{concat(width-in-pct, '%')}"		
fo:leader xmlns:fo="http://www.w3.org/1999/XSL/Formal" leader-length="100%" rule="		
<fo:leader.xmlns:fo="http: 1999="" <="" formal"="" leader-length="100%" td="" www.w3.org="" xsl=""></fo:leader.xmlns:fo="http:>		
OK ++>>セル		

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)。

fo:leader lead / fo:leader lead

- 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。
- メモ 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。

三 関連項目

- <u>ユーザー定義 XML テキスト ブロック</u>
- <u>ユーザー定義テンプレート</u>

5.5.2 ユーザー定義 XML テキストブロック

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 アラグメト 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た時に使用されます。

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う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。 灹 この機能は、Enterprise エディションの Altova 製品 (Style Vision、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:

- 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。
 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。
 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。

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。
例:
<ol list-style-type:="" lower-alpha"="" style="list-style-type: lower-alpha>
リストアイテム
</td></tr><tr><td colspan=3>1 Col style="> 2 List Item 3
ОК + +>ンセル

- 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。
- バモ 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。
- 三 関連項目
- <u>ユーザー定義要素</u>
- 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 テーブルとら、2種類のテーブルを使用することができます。このセグタンではこれらのテーブルを使用する方法にていて説明します。

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/フォーマナトされます。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 テーブルはデーダを表示するためだりれて使用されるもので、出力フォーマ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レテンソは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:

- 静的テーブルはよ SPS のデザイナーによれた成されるテーブルのことです。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。セルのエンテンソはスキーマソノーこ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してとご注意してくたさい。繰り返しが行われないすより、テーブルは静的と呼ばれており、 XML エンテングからは影響を受けれい固定された構造とないます。
- 動的テーブルはよ 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 ー少な くとも1つの子属性おけま子要素 - 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か複 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、全ておけまー部 の子要素や属性がテーブルのカラム」てなります。そのため、動的テーブルの構造は XML ファイルのコンテンメンは衣存し、コンテンメこ より動的に変化します。

CALS/HTML テーブル

CALS テーブルおけまHTML 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ドキュメトにて(CALS おけまHTML の壮様に従うかたちで)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ナトにおけるテーブルマークアップが生成されます。 CALS/HTML テーブルのフォーマナトは XML インスタンスドキュメント や SPS (おけまそれら両方)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

上のテーブルマ対応するHTMLコードのフラグメートを以下に示します:

</テーブル>

オジナルのXMLドキュメトは以下のようてなります

XMLドキュメイ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たさい。テーブルの構造は HTML おけは CALS テーブ ルモデルに対応している必要が設ます。以下を参照くたさい

- XML 要素だがHTML カラム要素のくtd/> しう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、デーダをテーブルセルに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て使用されます。
- CALS/HTML テーブルのフォーマトプロ ディはXML ドキュメトから得られる おか、SPS デザイノ に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。

三 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」<u>または「挿入 | テーブルを挿入」</u>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 Ш 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が表示されます(以下のスクノーシンコントを参照)。

StyleVi	sion
?	固定された数の行/カラムの静的テーブル、または入力データをベースにした動的テーブルを作成しますか?
	静的テーブル 動的テーブル キャンセル

静的テーブルをクリックします。

3. テーブルの挿入ダイアログか表示されるので、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。

方」ム:	ОК *юンセル

4. 「OK」をクルクします。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、テーブルのセルニコンテンツを入力することができます。直接入力され たデキストや、スキーマソノーからドラッグされた要素、おけよイメージなどのオブジェクトや入れ子になったテーブルをテーブルに含めるこ とかできます。入れ子構造になったーブルの例を以下に示します。

Person Telephone		Fax	
	Office Home	Office Home	

静的な SPS テーブルよ スキーマ構造の、サニるところに散りばかられた XML デーダを表示し、静的なコンテンツ(XML ソースから与えられて はないコ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、おオオテーブルを削除するコは、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テーブル | 行を削除」、「テーブル | 列を削除」、おオガテーブル | テーブルを削除」を選択します。

ツール ーテーブルの編集 アイコン

デフォルトでソールレーの2段目に配置されているテーブル編集アイエンは、テーブルメニューコマンドに対するショートカナトどおっています。これらのコマンドを使用することで、静的テーブルの構造を編集し、フォーマナプロンティを割り当てることができます。これらのアイコンは動的 SPS テーブルに対しても使用することができます。

- 三 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ナ

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。

- 1. メニューコマンドからから「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(以下のスクレーンショナを参照)。ツール・イニあるテーブルの挿入アイエをクトックした 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か表示されます。

StyleVision	
2 Bz	れた数の行/カラムの静的テーブル、または入力データをベースにした動的テーブルを作成しますか?
	静的テーブル 動的テーブル キャンセル

動的テーブルボタンをクリックします。

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。

SPath セレ	クター	(×
XPath コンテキスト:	\$XML/n1:OrgChart/n1:Office/r	n1:Departme	ent
◎ スキーマ パス スキーマの属性:	または要素を選択:		
	() n l:Name () n l:Office ⊡ () n l:Name ⊡ () n l:Desc () n l:Location		
	 In 1:Address In 1:Address_EU In 1:Phone n 1:Fax n 1:EMail n 1:Department 		III
	() n1:Name ⊕ () n1:Person	l	- -
4		•	
■ 絶対 XPal ○ ユーザー定義 X	th Path		
	XPa	thの編集(E).	
	ок (キャンセル	

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

4. 「OK」をクリックします。動的テーフ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)。

動的テーブルの作成	×
n1:Person	の各アイテムに対して
テーブル拡張方向	ヘッダー/フッター ▼ ヘッダーを作成(<u>H</u>) 「 フッターを作成(E) ▼ 数値フィールドの要約(<u>6</u>)
セルの表示方法 ③ コンテンツ ③ コンポポックス ④ 編集フィールド ④ 複数行入力フィールド ④ チェックポックス ④ ラジオポタン	行/カラムを達訳 <pre></pre> <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
	ОК 4 +>>dzu

- 5. デザインウィドウボ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リ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(上の別では、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」をクリックします。カラムは子要素や属性に対して行け作成され、それより下の階層に対しては作成されないことに注意して くたさい。
- 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つけて拡張される場合、テーブルは以下のように表示されます。

name	street	city	state	zip
O ipo:name	O ipo:street	0 ipo:city	0 ipo:state	O ipo:zip
(コンテンツ)			(コンテンツ)	
C ipo:name	V ipo:street	V ipo:city	V ipo:state	C ipo:zip

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

name	<mark>Oiponame</mark> (コンテンツ) Oiponame
street	<mark>Oipo:street</mark> (コンテンツ) Oipo:street
city	<mark>Oipotity</mark> (コンテンツ) <mark>Oipotity</mark>
state	<mark> O ipo:state</mark> (コンテンツ) O ipo:state
zip	0 ipo:zip (コンテンツ) 0 ipo:zip

~ダーとファター

カラムと行はオカラムならいれて行要素の名前となるへ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。ヘッダーを含めるはは、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、テーブルボディー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左から右になっている場合、テーブルボ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)に交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っている場合、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よ、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こないます。更に、ヘッダーとフッターコ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コ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

ある動的テーブルに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、入れ子になった)動的テーブルを追加することができます。 入れ子になった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。
- 3. 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は、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シテーブルの作成」を選択してください。

テキストコンテンン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、テーブルとして作成された要素内に子要素ノードや属性ノードか含まれていなければならず、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という要素からテーブルを作成する場合、Textという名前の子要素を新行ご作成して、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を動的テーブルとして作成することができます。各行のTextにはデキストノードか含まれ、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しています。

テーブルボディーセルのコンテンソ

動的テーブルを作成すると、ノードコンテンンを任意の StyleVision コンポーネン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 ディーのセルはコンテンソ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ィイ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には、テーブルを作成した後に個々のセルを編集する必要があります。セルを右クトックして、コンテキストメニューの変更」から、目的のセルニンテンツ型を選択してください。
- セルを要素コンテンソとして作成し、その要素が派生型である場合、セルの文字列コンテンソと、その派生要素全てが連結されたわた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、テーブル内にカーノルを配置し、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」、おコゴテーブル | 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・イニ用意されているテーブル編集アイエイオテーブルメニューコマンドへのショートカトとなります。これらのコマンドを使用することで、動 的テーブルの構造やフォーマ・トプロ・ティを挿入、削除、そして編集することができます。これらのアイコノオ静的テーブルに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」ないます。

グロー・ジレテンプレート内の動的テーブルの作成

動的テーブルオグロー・シレ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・シレテンプレートでは、グロー・シレテンプレートング、「ドロー・シレテンプレート」 きず、グロー・シレテンプレート要素そのものにつけして作成することはできません。 例えば、グロー・シレ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ことは、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 要素のグロー・シレテンプレート内部にテーブルを作成することはできません。

- □ 関連項目
- 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、お当よカラムや行のヘッダー(されして条件付きの処理をセオすることで、条件の評価 結果に従うカナちでカラムや行、お当よヘッダーを表示/隠すことができます。条件の結果がtrue どなる場合、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、ヘッダーコネルて条件付き処理を追加するコよ目的のデザインコンポーネトを右クリックして、条件処理の編集を選択します。(下のスクレーンショナでは、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トラインされています。行のコ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。)

	—		—
٢	First	Last	Email
Þ	OFirst (コンテンツ) OFirst	OLast (コンテンツ) OLast	OEmail (コンテンツ) OEmail

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が表示され、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コよそのデザインコンポーネントを右クリンクして、条件処理の編集を選択してくたさい。XPath式の 編集ダイアログ
か表示され、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は、目的のデザインコンポーネントを右クリックして、条件処理のクリアを選択してくたさい。

- 三 関連項目
- <u>静的テーブルの作成</u>
- <u>動的テーブルの作成</u>
- <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

静的ならいに動的 SPS テーブルのエレポーネントを、テーブルマーク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

N	•	•
130	Header-C1	Header-C2
	<mark>0 First</mark>)(コンテンツ) <mark>0 First</mark>	<mark>0Last</mark>)(コンテンツ) <mark>0Last</mark>
D	Footer-C1	Footer-C2

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きまれているとう単純なものです。

- 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。

(C) 2015-2021 Altova GmbH

- 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)。

	Ω element	
	•	\sim
Header-R1	OFirst (コンテンツ) OFirst	Footer-R1
Header-R2	0Last (コンテンツ) 0Last	Footer-R2

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。

ドラッグアンド ドロップ 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

行ませまカラムのサインを右クリックして、コンテキストメニューから、タグで囲む | テンプレート」ませまずタグで囲む | 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、ませまユ 一ザー定義テンプレート 」本地て XPath 条件式を入力することで、行やカラムの周リにテンプレートが作成されます。



行ませまカラ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に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 キーを押下してくたさい。

行やカラムのエンテンツ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たことを、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、行ませるわから、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。

- 三 関連項目
- <u>テーブルメニュー</u>
- <u>静的テーブルの作成</u>
- <u>動的テーブルの作成</u>
- 静的と動的テーブルのフォーマナ
- <u>ノードテンプレートオペレーション</u>

<u>ユーザー定義テンプレート</u>

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ナ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

プロッティサイドバー

プロ・ティサイドバーでは、HTML テーブルフォーマ・トプロ・ティを利用することができます(以下のスクノーシンヨ・トを参照)。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・ティーイは、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・オプロ・ティを利用することができます(以下のスクノーシンヨットを参照)。テーブルコンポーネントやサブコンポーネン(ボディー、行、カラム、セルリコ対して CSS プロ・ティを使用することができます。

 スタイル: 1 コンテンツ 1 テーブル 1 trow 1 tcolumn 1 tcell WSL-FO Typ
スタイル: 画性 値 • 1 コンテンツ 日 effects 1 テーブル 田 1 tbody 田 1 trow 田 1 tcolumn 田 1 tcell 田 ※SL-FO 田 テキスト 田 テーブル 田 オント 田 ボックス 田 王
height max-height max-height max-width min-height min-height min-height min-width min
width 200px 💌

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、おけ抜時計回り、30度回転させることができ、上から下お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。デザイノニてテーブルセル内のコンテンンを選択し、プロ・ティサイドバーニて tcell を選択します(以下のスクレーンションを参照)。テーブルセルプロ、ティグループにある方向プロ、ティの値を選択してくたさい。



以下の点に注意してくたさい

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
- 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、ティを適用することができます。

プロノティインらびしてタイルレンとのテーブルのフォーマト

フォーマ・ケプロ・ ティは、 プロ・ ティサイド・・・とスタイルサイド・、・の両方で利用することができます。 両方のサイド・・・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に示します。

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 (スタイルサイドバーこて 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
包レ	align、valign	height、width、vertical-align、color、 font、text styles、box styles

高さ幅

テーブルや行、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)にてセルすることができます。マウスを使ってデザインに表示されているテーブルやカラム、おけょ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、変更されたしも前的に入力されます。余白が指定されたセルに対してheight やwidth スタイルはサポートされません(row-spanned おけま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 ある align プロ・ティの値を center にセルします。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・ティへアクセスすることができます。 お当よ プロ・ティサ イドバーイ ある HTML グループからも align プロ・ティへ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はまご使用のFOP プロセッサービ従う。設定を行う必要があります。FO の仕様に従えば fo:table-and-caption 要素をとなう fo:table によけテーブルを囲み、fo:table-and-caption 要素のtext-align 属性の値に cneter を指 定すること」なります。デザイノニテーブルが挿入された引きでは、fo:table-and-caption 属性は作成されませんが、ユーザー定義要素 と、 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すっ:table-and-caption 要素がサポートされていないこともあります。この場合、幅が固定されたテーブルを作成し、(「テーブル | テーブルのプロパティ」から アクセスすることができる)テーブルプロペティのHTML グループにて width プロペティの値を 4in まちは 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ルすることができます。

- 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、<u>XPath 式 の編集ダイアログ</u>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、さらご複雑なフォーマ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 条件式か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() 関数によし返されます。

PDF 出力におけるテーブルへッダーとフッター

テーブルがページをおよて表示される場合、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か表示されます。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て表示されるヘッダーやフッターの表示を回避するコよ table 要素のtable-omit-header-at-break やtableomit-footer-at-break プロ ディ(属性)を使用してください。これらのプロ ディヘよ 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 おヘッ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 にセルします。 デフォルトの値は false 「なっており、値の指定を行わないと、 改ページが行われるたびにヘッダーとフッターか挿入されます。

<u>テーブルセルのコンテンソンメノフンを追加する</u> テーブルセル内のコンテンソンメフンを追加するコよ、対応するブロックにていイフンを追加してください。

三 関連項目

- 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に対して用意された行ならびにカ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 おけは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。

空の行ならびにカラムをデフォルトで隠す

以下の操作により、HTML 出力にて空の行ませよカラム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
- アリアインビューにて、ア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)。

プロパティ			X
la 📲 📲	PRTH ×		_
プロパティ:	属性	値	A
テーブル	中 テーブル		
	カラムを隠す	決してしない	
	行を隠す	決してしない	
	🕀 Authentic	決してしない	A
	-⊞ HTML	空であれば	
	⊕ イベント	ホティーが呈いめれ	ua 🔟
	田 インタラクティブ		
	田 共通		
			~

- 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タンを含めるように指定することができます(以下のスクレーンショナを参照)。閉じる ボタ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か隠されると、最初のカラムにプラスシンボルが表示されます(以下のスクレーンショナを参照)。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	Č.	€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 (上のスクノーシンョナを参照)。HTML 出力にてこれらのシンボルをクリ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されてインデン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、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ナを参照)。

プロパティ				x
la 📲 📲	PATH			
プロパティ:	属性	値		-
テーブル	田 テーブル			
	🕀 Authentic			
	·⊞ HTML			
	田 イベント			
	中 インタラクティブ			
	カラムを閉じるボタンを追加	no		
	行の展開/縮退ボタンを追加	no	•	
	田 共通			
				-

プロ・ティー・セントすることができる値は、機能を有効にする)yes おけば、機能を無効にする)no どなります。

- 🖃 関連項目
- <u>静的テーブルの作成</u>
- <u>動的テーブルの作成</u>
- 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 お台は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) そのテーブルのエンテンルを含むことができます。XML の構造は、CALS お台は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. <u>XML 構造を CALS /HTML テーブル構造とて定義する</u>
- 2. <u>テーブルのためのスタイルのフォーマトを指定する</u>
- 3. <u>SPS デザイン内にCALS/HTML テーブルを挿入する</u>

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コンテンン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 テーブルモデルに対応しており、HTML テーブルモデルと同一の要素名が含まれています。

<td/tr>
<t/tbody>

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。

<semester>
<subject>
<class>
<student/>
</class>
</subject>
</subject>
</semester>

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れ、このXMLデータをCALS おうまHTMLテーブルとして定義する必要が砂ます。XMLデータ構造がCALS や (デフォルトの)HTMLテーブルとして定義されてしたよ場合、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 メトトニテーブルマークアップが使用されることは砂ません。

以下の方法によりCALS/HTML テーブルマーク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。

 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 を参照ください。HTML テーブル構造を含むテ ブル要素の例については、(プロジェクトウィドウンある) 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。

テーブルの型	メイン要素	
CALS	table	
CALS	informaltable	
HTML	table	
HTML	informaltable	

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。

			\times
说明	要素名	スタイル	
テーブル要素	table		
テーブルキャプション	caption		
テーブルヘッダー	thead		
テーブルフッター	tfoot		
テーブルボディー	tbody		
行要素	tr		
カラム要素	td		
ヘッダー要素	th		

- FOK」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。
 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、ペインの上部にある削除がな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)。

****	西 夫 夕		Χ
況明 ニーゴル 亜 圭		X9174	
テーブル要素	table		
テーノルキャノンヨン	caption		
テーフルヘッター	thead	background-color:#d2d2d2;	
テーブルフッター	tfoot		
テーブルボディー	tbody		
行要素	tr		
カラム要素	td		
ヘッダー要素	th		

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 スタイルサイド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に挿入する

CALS/HTML テーブル構造をデザインへ挿入するココン種類の方法があります:

- 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 テーブルして定義</u>されていない場合、CALS/HTML テーブルの挿入ダイアログが表示され、CALS おけはHTML テーブルとて要素を定義することができます(以下のスクノージョントを参照)。

CALS/HTML テーブルの挿入
メイン要素 'Office'に対して定義をれた CALS/HTMLテーブルがありません! 新規 CALS/HTMLテーブルモデル定義を作成しますか?
○ CALS テーブルを作成
○ HTMLテーブルを作成
OK(<u>O</u>) (キャンセル(<u>C</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)。

Oroot Oelement Otable HTML
サンブル HTML テーブル
サンブル HTML エントリ サンブル HTML エントリ
サンブル HTML エントリ サンブル HTML エントリ
Otable Oelement Oroot

テーブル要素のグロー・ゾレテンプレート

以下のテーブル要素のグロー・シーブルート 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。

- 三 関連項目
- <u>静的テーブルの作成</u>
- <u>動的テーブルの作成</u>

5.7 リスト

SPS にて作成することができるリストには以下の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ドキュメトの構造が動的に得られること はありません。
- <u>動的なリスト</u>: XML ドキュメイから構造とエンテンンを動的に取得するリストです。

このセグョンのサブセクションでは、これら2種類のリストを作成する方法について説明します。



<u>テーブルの作業</u>

5.7.1 静的なリスト

静的なリストとは、リストアイテムのエレデンンを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:

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・</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)。

StyleVis	sion			×
?	入力データをベース	に静 <mark>的または動</mark> B	りテーブルを作成し	ますか?
	静的リスト	動的リスト	キャンセル	

2. 静的リストをクルクします。箇条書き、お台は番号付きのノストを選択するダイアログが表示されます。(アのスクリーンションイ).
| リスト | | | | × |
|--------|----------|---|-----|-------|
| 箇条書き 者 | 野号付き箇条書き | | | |
| | | • | | |
| | | • | | |
| | | | K (| キャンセル |

- 8. 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挿入 箇条書きを挿入 を選択します。そ の後上記の操作を再度行なってくたさい。

灹 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・「ある」スト(記号)まけました(数字)アイ コンをクリックすることでも、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、静的テキストをリ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。テキスト内に改行か含まれる場合、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しませ、必要し応じて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

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エレテンンをリストアイテムとして表示することができます。 インスタレ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。
- 要素がリストとして作成されると、その要素のインスタンスがリ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 からなるリストを作成する場合、 person 要素をリストとして作成する 必要があります。
- 要素に対してリストを作成した後に、テキスト、自動計算、動的コンテンソなどの静的おける動的コンテンンを挿入することで、リストやリストアイテムの外観おけよコンテンン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:

 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 挿入 箇条書きを挿入 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)。

StyleVis	sion			×
?	入力データをベース	スに静的または動B	タテーブルを作成しま	すか?
	静的リスト	動的リスト	キャンセル	

- 2. 動的リストをクリックします。XPath セレクターダイアログか表示されます(以下のスクレーンショナを参照)。
- 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。.

▲ XPath セレクター	x
XPathコンテキスト: \$XML/n1:OrgChart/n1:Office/	'n1:Department
③ スキーマ パス スキーマの属性または要素を選択:	
On DName On 1:Office On 1:Name On 1:Name On 1:Desc On 1:Location	_
 In 1:Address In 1:Address_EU In 1:Phone In 1:Fax In 1:EMail In 1:Department In 1:Name 	=
	· ·
	,
□ #Exy AFath	
XPa	ath の編集(E)
ОК	++>2010 //

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

6. 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。

箇条書き 番号付き	商条書き	
	•	
	•	
	•	
o		
。	•	
。	-	

- ∃ 関連項目
- <u>静的なリスト</u>
- <u>IJZŀ</u>

5.8 イメージ

デザインドキュメント ヘ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、 <u>イメージ URI</u> のセクションを参照ください。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、 <u>イメージ型 と出力</u>セクション」こて記述されます。

イメージのプロレディ

イメージはプロ・ティウィンドウでセットすることができます。 デザイン内でイメージを選択し、プロ・ティサイドバートこて(i) プロ・ティのカラムこてイ メージを選択し、(ii) 目的のプロ・ティグループを選択し、(iii) 選択されたプロ・ティグループにある目的のプロ・ティを選択します。 例えば、イ メージの高さと幅をセットするコよ HTML グループにある height とwidth プロ・ティイゴ値をセットします。

- 三 関連項目
- <u>イメージの挿入</u>
- 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に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せま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(以下のスクノーンショナを参照)。

イメージの挿入	×
静的 動的 静的ならびに動的 インライン データ	
青 争6勺	
アドレス(E):	
□ 絶対パス 参照	単純な静的アドレスの入力に、 上のフィールドを使用してくださ
	<u>OK</u> キャンセル

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、イメージファイルカ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てアクセスされ、ドキュメン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(上のスク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
 (以下のセグニンを参照くだれ)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) パースされたエンティティ名 を含むメードを選択する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 のセクションを参照ください。
- 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 封は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)することはく、ソース XML ファイル内にあるイメージへアクセスすることができるようによります。 XML ソース内部 にてエンコードされたイメージを挿入するはは、イメージの挿入ダイアログにあるインラインデータダブを使用します(以下のスクリーンショットを参照)。

イメージの挿入
インラインデータ XPath : //images/marketinglogo イメージフォーマット : png マ エンコーディング : base64 マ XPath を編集 イメージファイル設定
OK キャンセル

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ドキュメート内のノードを指定します。イメージフォーマトコンボボックスから、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ボボックスでは、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ボタンか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に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 Style Vision 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ださい

- 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 <u>プロン ディダイアログ</u> にて、イメージン使用される相対パンを SPS おけは XML ファイル に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。

イメージのプロノティの編集

イメージを変更するコよ、デザインビューコン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フェッルます。 プロ・ティサイド・バーニ表示され 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atプロ・ティセ合まれており、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 なえ エンティティ URI

5.8.2 イメージ型 と出力

StyleVision にてサポートされているイメージと、StyleVision にてサポートされている出力形式を以下に示します。StyleVisionのエディ ションイニメリサポートされる出力フォーマルのセルが異なる点に注意してください。Enterprise エディンヨンではHTML、Authentic、 RTF、PDF、Word 2007+ がサポートされるのにつけして、Professional エディンヨンではHTML とAuthentic、RTFが、Basic エ ディンヨンではHTML かサポートされます。

イメージ型	Authentic	HTML	RTF	PDF	Word 2007+
JPEG	こと	こと	して	ほ	こと
GIF	こと	こと	こと	ほ	こと
PNG	こと	はい	は、	ほ	こと
BMP	は	はい	は、	ほ	こと
TIFF	は*	こそ	は、	ほ	こと
SVG	は*	記ょ	いえ	ほ	いえ
JPEG XR	はい	はい	いえ	いえ	いえ

* 下の点に留意してくたさい

以下の点に注意してくたさい

- 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えい場合、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 サイトを参照けさい。

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:

C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples/Tutorial/Images

HTMLにおるSVG

マウスイベトーマオオ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 のレンダレグが行われ、インタラクティブ性が失われます。この制約は、オブジェクトとして外部 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 コードフラグメートを追加することに回避することができます。.

SVG をケエブページに含めるコよ以下に示される3種類の方法があります。

- 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 タグを生成します。インタラクティブ性は失われます。
- ユーザー定義要素機能によりオジェケトとして挿入された外部 SVG(以下のスクレーンショナを参照): type 属性を正しく挿入するようしてくたさい(例: type="image/svg+xml")。このようご挿入することで、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います。

ユーザー定義要素の編集
要素の名前と、二重引用符(ダブルウォート)で囲まれた属性の繰り返しを入力してください。
例えば、HTMLにて50%の幅を持った「水平線」要素を入力するには:
hr width="50%"
動的属性の値に対しては、括弧で囲まれた XPath 条件式を入力します:
hr width="{concat(width-in-pct, '%')}"
object data="circles.svg" height="buupx" type="image/svg+xml" width="100%"
<object data="circles.svg" height="500px" type="image/svg+xml" width="100%"></object>
OK ++>ンセル

3. <u>ユーザー定義 XML ブロック</u>により挿入されたインライン SVG: SVG コードブラグメトの例を以下のスクレーショナに示します。 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 おは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。



三 関連項目

• <u>イメージURI</u>

5.8.3 サンプル:イメージのテンプレート

StyleVision パッケージコは StyleVision は搭載されているイメージ機能のデモするけなの SPS ファイルは(マ <u>1ドキュメトフォルダー</u>: C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\Images\Images.sps です。 Images ドキュメナ (Images.xml とImages.sps)は 3つの ーンから構成されます:

- 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+ 出力フォーマナは、StyleVision のEnterprise Edition とProfessional Edition (RTF) て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。
- 三 関連項目
- <u>イメージURI</u>
- <u>イメージ型 出力</u>

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、(入力フィールドやコンボボックスとッオー・テータ入力デッドスとして作成することができます。 HTML 出力では、デザインビュートこおける表示に類似したカヤェちで、データ入力デッドイスがオブジェイトとしてレンダレングされます。 HTML 出力では動作してい 点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ノストを、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
入力フィールド(テキストボックス)	ユーザーにより入力されたテキスト
複数行 入力フィールド	ユーザーにより入力されたテキスト
コンポポックス	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ピング
ラジオボタン	ユーザーの選択を値へマピング
おと	ユーザーの選択を値へマノピング

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ィンスでは、 Authentic View ユーザーの選択が値へマンピングされます。StyleVision 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 プロントン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」ております。

- 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
- コンボボックス
- チェックボックス
- ラジオボタン
- おン

一般的な使用方法

以下の操作により、データ入力デッドノ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カらデザインビューイン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、ノードに対して作成するデータ入力デバイスを選択します。
- 3.データ入力デッドイスによってはダイアログか表示されます。ダイアログか表示される場合、必要な情報をダイアログに入力し、「OK」を クトックします。

データ入力デバイスのプロ・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)そのデータ入力デバイスをクトックして、プロノ・ティサイドバーに表示されているプロ・ティを編集します。

<u>×</u>E

- データは、HTML 出力のデータ入力デバイスコは入力することができません。データ入力デバイスは、コンテンンを表示する代替の 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トをデータ入力デバイスへ変更することでも、データ入力デバイスを作成することができます。ノードを右 クリックして、変更を選択してくたさい。

• HTML 出力では、ユーザーにより選択されたエトリーカ表示されます。HTMLドキュメントに表示されたデータ入力デバイ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。

- 入力フィールド、複数行入力フィールド
- <u>ユンボボックス</u>
- <u>チェックボックス</u>
- · デジオボタン、ボタン

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ドウからデザインビュー・ソ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 てきます。そのソ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ノティを編集する

入力フィールドを選択し、プロ・ティサイドバーのHTML プロ・ティを修正することで、入力フィールドのHTML プロ・ティを修正することができます。

例えば、入力フィーノドが選択された状態で、編集フィールドを選択し、プロイティのHTML グループ以下にある maxlength プロイティを選択します。maxlength の植フィーノドをダブルクトックして、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。
- 三 関連項目
- <u>ユノボボックス</u>
- <u>チェックボックス</u>
- <u>ランオドタン、ドタン</u>

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バ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バ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 かりポートされない ため、XML ファイルに値を与えることはできません)。

[□] 関連項目

🛐 チェックボックスの編集	×
以下の要素に対して : Data	ОК
見目 X	キャンセル
true	
チェック解除の値: false	
	1.

上のスクノーンショナでは Name とう名前の要素がチェックボックスとて作成されまた。Authentic View ユーザーかチェックボックスに チェックを入れた場合、true とう値がName 要素の値とて入力されます。チェックボックスのチェックが外された場合、(ダイアログにて定義 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コはチェックボックスの編集ダイアログか表示されます。以下の操作に より、チェックボックスの編集ダイアログへ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)。

Properties		X
占 🗄 🐇 🚠		
Properties For:	Attribute	Value 📥
checkbox	🔁 checkbox	
	checked values (true / false)	true;1 / false 🛛 💀
	⊕ common	
	🕀 event	
		*

- 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、 メモ 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。

三 関連項目

入力フィールド、複数行入力フィールド

- <u>エノボボックス</u>
- ラジオドダン、ドダン

5.9.3 コンボボックス

コンボボックスにより、ドロップダウンリ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。リストは生成されオポキ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、コンボボックスを使った。 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。
- エンドドックスダイアログニて定義されたリストから。表示されるエントリーと、それに対応する XML の値を入力します。 XML の値は Enterprise ならびに 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 へのマメビングが行われます。 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 すっか)。
- 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カドロップダウンリストのエトリーとして表示され、

5.9.4 ラジオボタン、ボタン

ポタノコよ通常のたシンテンオオタンの2種類がおます。ランオオタンとオタノよフォームに入力する場合、または、HTML 出力内でイベントをトリガーする場合役にさきます。

- 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。
- 旦 関連項目
- 入力フィールド、複数行入力フィールド
- <u>エノボボックス</u>
- <u>チェックボックス</u>

5.10 リンク

リンク(お台ンイパーンク)を、ドキュメント内のブックマークやウェブページといす。外部リソースコネリして作成することができます。リンクは動的 に生成されたアンカーコネリしても作成することができます。StyleVision では、ハイパーレンクニ使用される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ランでは、静的ならび、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。

- 三 関連項目
- <u>ブッケマークとハイパーレク</u>

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、出力ドキュメント内ロンー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ンを参照)。デザインドキュメント内のゾーコードを挿入する場所に<u>パーコードデザイン要素を挿入し</u>プロンティを指定します。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)モードでのメッサポートされるということに注意してくたさい。



重要: バーコードを動作させるためはは、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 パッケージのビットバージョン (32-ビット おけお4-ビット) と同一のものではければないません。

バーコードを挿入する

以下の方法により、デザイン内部にいーコードを挿入することができます

 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)。

-) barcode	O-d-b-	
	Codabar	_
lext	ItemID	
setModuleWidth	Ipx	
setWideFactor		
setBarHeight		
setHeight		
doQuietZone		_
setQuietZone		
setMsgPosition		_
setPattern		
setFontName		
setFontSize		
setChecksumMode		.
orientation		-
pixelDensity		T
generatedImageSettings		
eneratedImageSettings		

- 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 守ィの値と 目的口応じオプションのプロ ディをセルしてくたさい。XML ファイル内にある値をプロ 守ィの値とて使用する場合、XPath エを入力 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 条件式を使用するゴよ 目的のプロ ディを選択し、プロ ディダイアログのソール ドームある XPath ポタンを有効にして、 XPath 三人中式 クログログロ CPA 条件式を入力します。XPath 条件式に対した ノード内で評価におます。
- 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、トーコードのプロ・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 ションとなります。 バーコードの種類により、利用することのできるプロ・ティが異なるということ コ留意してくたさい。

• 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。

1%-ンの構文

・ ゆーンにより、バーコードに含まれる入力メッセージの文字列をより読みやすい めたちで出力することができます。 ・ ゆーンでは、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。 ・ ゆ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。 、 ドックスラッシュ ´¥´ はエスケープシンボリとなります。 例えば ´¥?´ により ´?´ と ら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、オリ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、 ゆー ンの構文例を参照くたさい。

入力メッセージテキスト	ヽ゚゚゚゚゚ーン	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、プロハティダイアログのパタダブにて指定された場所に保存されます。プロハティダイアログへは、メニューオプションカジファイル | プロパティ」を選択することでアクセスすることができます(以下のスクリーンショナを参照)。

 3 プロパティ
出力 XSD/XSLT/XBRL イメージ パス 解像度 Authentic
以下のディレクトリにプレビューファイルを作成
◎ SPS ファイル(S)
◎ メインス キーマ ソースの作業用 XML ファイル(X)
追加で生成されたファイルの場所
 デフォルト (メイン出力ファイルのフォルダー)
● Windows 一時ノオルター ● カフタム パフ・
ファイル パスを SPS ファイルに保存
SPS ファイルと同一のまたはそのサブフォルダーにあるすべてのファ ③ イルを SPS ファイルからの相対パスとし、それ以外は絶対パスと する
○ 可能な場所すべで SPS からの相対パスを使用する (他のディ スクまたは外部にあるファイルは絶対パスで表現されます)
スキーマ、作業用ならびにテンプレート XML ファイル、SPS モジュー ル、CSS ファイルなどに適用されます。
OK キャンセル

プレニーで使用される、トコードのイメージファイルは、SPSファイルませば作業用XMLファイルが収められているデルクトリニ作成されます。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た際には任意の場所にファードイメージファイルを作成することができます。パタダブにある「追加で生成されたファイルの場所」ペイムにて場所を指定することができます(上のスクレージショントを参照)。

三 関連項目

• SPS ファイル エレテンソ

5.12 レイアウト モジュール

レイアオモジュール出まレイアオを含むオブシェオのことです。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 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オボックスをレイアオービシュ ール内に配置することもできます。レイアオー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、レイアオ を作成することができます。

レイアナモジュールを作成するためのステップを以下に示します。

- 1. レイアナンテナーを挿入するレイアナ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、(サイドバトーこある)スタイルを使用することでレイアナ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つ以上のレイアナボックスを挿入するレイアナボックス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、(スタイルサイドバーにある)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ト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に対して相対的に位置を変更し、前面おけば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、おこしイアナボックスやその他のラインといすこしイアナオブジェク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、フレー・ンドのデザインか、フォームを基にしたデザインカ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、レイアウトコンテナーにと、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。

三 関連項目

- <u>デザインの作成</u>
- SPS ファイル エレテンソ

5.12.1 レイアウトコンテナー

レイアナコンテナーは以下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しるコンテナー内に 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、レイアナコンテナーの サイズ セケシュン参照ください。
- レイアナクリッドとズーム機能により、レイアナコンテナー内にあるオブシェク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 <u>スタイルプロパティ</u>を、全てのエンテナーコズルて指定することができます。
- レイアウト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に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オメ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。

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。

レイアウトコンテナーを挿入する

レイアナンテナーを挿入するコま、デザイン要素の挿入 ツールゲーにあるレイアウトコンテナーの挿入アインをクトック、た後に、 レイアナンテナーを挿入する位置をクトックします。レイアナ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、そのドキュメトセンションのページレイアナプロ・ティーで定義されたページの寸法と同じ大きさがレイアウ トコンテナーイン対して与えられます。いいえをクトックした場合、デフォールサイズの3.5in x 5.0inのレイアナコンテナーが作成されます。

レイアナコンテナーは 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

レイアナコンテナーの大きさは、以下の2つのプロ・ティには決定されます:

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。

プロパティ			×
占 🗄 🔚 🚠 🥻	×		
プロパティ:	属性	値	A
レイアウトコンテナー	日 レイアウトコンテナー		
	用紙サイズに自動調整	no	
	🕀 Authentic		
	⊕ イベント		
	田 共通		
	□⊡ 設計図		
			$ \nabla $

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ゲ

レイアナトコンテナー内にグル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、グル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・トーニある)ズームコンボボックスを使用するか、Ctrl キーを押下しなか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。

レイアナコンテナースタイルプロパティ

レイアナコンテナー」ご適用することができるスタイルリコよ、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レティは、スタイルサイドバーはあるレイアナコンテナーにてセナされます(上のスクレーンショナを参照)。

レイアウトコンテナーのコンテンツ

レイアナンテナー内に含めることができるデザインアイテムよレイアナボックスとラインに限られます。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ンファ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、レイアナンテナーの大きさころ、そうアイルが使用されます。青写真イメージルイアナンテナーよりた小さい場合、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(以下のスクレーンショナにある青写真グループプロ、テを参照くたさい)。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もできます。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います。青写真はデザインビューだまっされ、出力ビューで表示されることはありません。

(プロ、ディサイド、、ーの)レイアナオンテナープロ、ディーある青写真のプロ、ディグループから、青写真のプロ、ディヘアクセスすることができます (以下のスクノーンショナを参照)。

フロパティー		×
님 내 내	PATH X	
プロパティ:	属性	[値] [4]
レイアウトコン…	田 レイアウトコンテナー	
	🕀 Authentic	
	田 イベント	
	田 共通	
	日 春写真イメージ	
	URL	file:///C:/Altova/StyleVision 🚥
	イメージを表示	yes 💌
	オフセット(上)	10mm
	オフセット(左)	10mm
	レイアウトコンテナーに収まるように調整	100%
	不透明性	50%
		v

レイアナンテナーに表示される青写真の透明度を指定することで、デザインの邪魔にならないよう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イティーに、必要に応じて表示/非表示を切り替えることができます。

- 🖻 関連項目
- レイアナホックス
- ライン

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、レイアナボックス内に配置する 必要があります。デザイン要素を含んたシイアナボックスは、必要に応じてレイアナニンテナートで配置されます。デザイン要素をレイアナニン テナートニ直接配置することはできず、レイアナ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、レイアナホ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
- レイアナボックスの選択と移動
- 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
- レイアウトボックスレニンテンンを挿入
- レイアナボックスの表示順序

レイアナボックスの挿入

レイアナボックスはレイアナニンテナーの中に対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デザイン要素の挿入ッツールバーにあるレイアナボックスの追加アイエをクリックして、レイアナボックスを挿入するレイアナニンテナーの場所をクリックします。クリックされた

場所が左上端となるようなかすたちでレイアナトボックスか挿入されます。レイアナトボ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す」まと側にある境界線までカーンルを移動させ、カーンルめ移動状態になった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。マウスパダンを押下しながら、レイアナボックスをレイアナニ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バーに表示されます。

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う名前のプロ、ティか含まれています(以下のスクレーンショナを参照)。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)コンテンソこ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、コンテンン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かま せん。

右および下側の境界線をドラッグすることで、レイアナト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 レイアナトボックスを選択 し、 Shift キーを押下しながら、 矢印キーを押下することで、 レイアナトボックスを目的の大きさこ変更することができます。

プロパティ			×
la 🕂 🖑			
プロパティ:	属性	値	A
レイアウトボッ	・ロ レイアウトボックス		
	サイズの自動調整	yes	
	方向	水平	•
	追加幅		
	追加高さ		
	🕀 Authentic		
	田 イベント		
	田 共通		
			$\overline{\mathbf{v}}$

追加高さど追加幅プロ・ティニより、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。追加幅ど追加高さよ、レイアナトボックスの大きさか手動で変更された際に、新活に計算されます。反対に、これらの値を変更することでも、レイアナト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ノティ

レイアナボックスのスタイルプロ・ティは、スタイルサイドバーにあるレイアナボックスしてセナすることができます(以下のスクレーンショナを参照)。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。

スタイル			×	
le 🗄 🐫 🚦				
スタイル: ユレイアウト… エリテキスト 	田 フォント 田 ボックス 田 共通 田 効果 日 印刷媒体 日 色 E background attachment			
	color	transparent	I 🕤	
	image		• •••	
	position		▼	
	repeat		•	
	color		- 😳	
	⊕ 詳細			
	田 順序付きリスト		-	

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ンを挿入

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。 レイアナボックスコ挿入できないてとい注意してくたさい。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。

プロパティ			×
님 내 내	PRTH		
プロパティ:	属性	値	A
レイアウトボッ	日 レイアウトボックス		
	サイズの自動調整	yes	
	方向	垂直(下から上)	
	追加幅	20%	
	追加高さ	1.18in	
	🕀 Authentic		
	⊕ イベント		
	田 共通		
			-

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。
- このプロレディ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。

レイアウト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したい場合、覆われたレイアナボックスが別のレイアナボックスには隠れてしまいます。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ーコンズを使用することで、選択されたイアナボックスを前面おけま背面へ移動 することができます。これらのコマズを使用することで、レイアナボックスを(i)最も近いボックスに対して移動(前面へ移動ならびに背面 へ移動コマズ)、おけま(ii)全てのボックスに対して移動(最前面へ移動ならびに最背面へ移動コマズ)することができます。上の スクレージョナでは、以下の順字によばバックスの配置されています。

- *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(レイアナボックスにデフォルトで使用される背景)をともなうレイアナボックスは、特に複数のボックスか透明となっており、更に境界線を持たない際に、他のアイテムに対して順序が変更されていないように見えることもあります。以下のスクノーンショナでは、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。



メモ
レイアサラインを、レイアサ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

• レイアオコンテナー



5.12.3 ライン

ライノは、レイアナボックスではよく、レイアナニンテナー内に挿入した後に、レイアナニンテナー内で選択、移動や大きさの変更を行い、 プロ・ティの割川当てや、レイアナボックスならびにラインからなるし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、ラインをレイアナコンテナートに追加することができます:

- 1. <u>デザイン要素の挿入</u>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。両端に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

メインウィンドウでは、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、起点おけは終点を選択し、大きさや方向を変更するように移動してくたさい。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、終点か移動されます)。

ラインプロ・ディ

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(以下のスクレーシンヨオを参照)。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。

プロパティ			×
님 🕂 📲	$\frac{\mathbf{x}}{\mathbf{p}\mathbf{R}\mathbf{T}\mathbf{H}} \mid \mathbf{X}$		
プロパティ:	属性	値	<u> </u>
ライン	豆 ライン		
	color	blue	💌 😳
	style	dashed	
	····· width	2px	
	終点♡>	0.79in	
	終点(Y)	0.84in	
	━━━ 起点♡>	0.27in	
	—— 起点(Y)	0.84in	
	🕀 Authentic		
	⊕ イベント		
	田 共通		-

プロ・ティサイドバーでは、以下のラインプロ・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。
- ★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」な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、そのラインのコンテキストメニューに表示される 順序メニューコマン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。これらのコマンドを使用することで、ラインを(i)最も近いアイ テムニマオして移動(前面へ移動ならびに背面へ移動コマンド)、おさは(ii)全てのアイテムニマオして移動(最前面へ移動ならびに 最背面へ移動コマンド)することができます。



上のスクレージョナでは、以下の順序で重ね合わせか表示されます:緑色のボックス、赤色のライン、黒色のライン、青色のボックス

- □ 関連項目
- レイアサコンテナー
- レイアナボックス

5.13 変更機能

テンプレートやテンプレートのエレテンン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、ませま(ii)そのノ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

ノードやそのエレテンルを変更することができます。以下に示されるスクノーシンコナのうち、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、ノード内部にあるコンテンンが選択されています。

0n1:Name (コンテンツ) 0n1:Name

<mark>0n1:Name</mark>)(コンテンツ)<mark>(0n1:Name</mark>

上にあるスクノーシンコナでは、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)プレースオルダにより 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: 入力フィーリドやコンボドックス)。その他の種類のコンテン ッ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リックして、コンテキストメニューから変更コマドへアクセスすることができます(以下のスクノーンションを参照)。

	グローバル テンプレート(z設定(M)
	グローバル テンプレートを使用(<u>U</u>)
	グローバル テンプレートをローカルにコピー 🕐
	テンプレートマッチを編集(工)
[11]	XPath フィルターの編集の
	XPath フィルターのクリア(L)
G	グループ(上(<u>G</u>)
A4	並べ替え(<u>S</u>)
(\$)	変数を定義(⊻)
ц <u>т</u>	レベルとして機能するテンプレート(<u>A</u>)
₽	デザイン フラグメントを作成(<u>C</u>)
	タグで囲む(110) ・
	変更(<u>H</u>) ,
	タグだけ前1除(<u>R</u>)
	Authentic プロパティの編集(<u>A</u>)

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、スキーマーにで要素の名前か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。ノードをクリックすることで、そのノードに対してテンプレートが適用されます。

	クロニハルフノフノレニドに設定し切り			
	グローバル テンプレートを使用(U)		コンテンツ(Q)	
	グローバル テンプレートをローカルにコピー (Y)		段落(P)	
			テンプレート(E)	n1:CompanyLogo
[//]	 XPath フィルターの編集型		変数テンプレート(1)*	✓ n1:Name
	XPath フィルターのクリア(L)		テーブル(I)	n1:Office
G	 グループ化(<u>G</u>)		CALS/HTML テーブル	
A4	並べ替え(<u>S</u>)		イメージΦ	
(\$)			入力フィールド(E)	
4=	 レベルとして機能するテンプレート(<u>A</u>)		複数行フィールド(<u>M</u>)	
₽	 デザイン フラグメントを作成(<u>C</u>)		リッチエティット(R) コンボ ボックス(Q)	
		•	チェック ボックス(<u>K</u>)	
	タグで囲む(W)	•	ラジオ ボタン(<u>R</u>)	
	変更(<u>H</u>)	•	ボタン(<u>B</u>)	
	タグだけ前11除(<u>R</u>)			
	Authentic プロパティの編集(<u>A</u>)			

テンプレートの記述とマッチしない ロンテンン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。

テノプレートマッチを変更することで、(ノードではなく)変数テノ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レテンンが選択されている場合、ノードが作成されたエレテンンの型を変更することができます。エレテキストメニューの変更コ マレドレ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レテンンの型がサブメニュートこて表示されます(以下のスクノーシンョナを参照)。

コンテンツ(<u>C</u>)	
テーブル(①	
イメージΦ	
入力フィールド(E)	
複数行フィールド(<u>M</u>)	
リッチエディット(<u>R</u>)	
コンボ ボックス(Q)	
チェック ボックス(<u>K</u>)	
ラジオ ボタン(<u>R</u>)	
ボタン(<u>B</u>)	

上のスクレージョナは、コンボボ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

- <u>SPS ファイルエレテンソ</u>
- テノプレートとデザインフラグメント

6 SPS 構造

SPSドキュメトの構造は、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。

- <u>_____</u>
- <u>モジュールSPS</u>
- <u>テンプレートとデザインプラグメナ</u>

入力ベースの構造: スキーマとモジュール SPS ファイル

入力ベースとうのよ SPS のノーススキーオことリ入力ドキュメナの構造が指定されることで、この構造がSPS ドキュメナの構造となります。例えば ソーススキーオこて 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ノースデータドキュメナト内にある各 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 といた子要素のセル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(以下のスクレーンショルを参 照)。テーブルのカラムは、Personの詳細(Person 要素の子要素)とないます。

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になります。テンプレートではPerson 要素に対してデザインされた行が「つ定義されており、この行によりノースドキュメント内にある各 Person 要素が処理され、Person 要素の数だは新たな行か作成され、子要素がテ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にて記述されます。

更に、StyleVision では作成したSPS をモジュールとて別の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こなります。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。

出力ベースの構造、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ド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デン プレートが含まれ、グロー・シ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、<u>テンプレートとデザインフラグメント</u>のセク ションコンに記述されます。

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アブメトと呼ばれる 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ト フラグメトに対して割り当てることができます。 グロー・ シレテンプレートと違い、 デザインアラグメトは(i) 複数のテンプレートから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ンテンンを別々のデザインアラグメ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になります。 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になります。 デザインフラグメートを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クションにて記述されます。

- □ 関連項目
- <u>使用方法の概要</u>

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リレト要素によげザインの構造 は影響を受けることさいます。

スキーマの選択

以下の操作により、新たな SPS ファイルマオオ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。デザク概要サイバーにて新しいソースの追加コマンドを選択しま 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。



スキーマノースは、ファイル、おとは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、 おーグロー・ シレ テンプレートとして使用する要素がスキーマにてグローマル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います。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マが要素となることに注意してくたさい、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・シリ要素として定義されているか、そしてスキーマーン対して変更が必要な場合にはよお更いのXML 環境にてそれがで きるかどうか調べるようしてくたさい。

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 メモ 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・シル要素が含まれている場合、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 とう名前のスキーでをベースしたUniversalSPS.sps にて Element-A とElement-B とう2つのレト要素が与 えられている場合、Element-A なギュメント要素とするインスタンスギキュメートと、Element-B なギュメント要素とする別のインスタンスギ キュメートを、単一のSPS から扱うことが可能しておます。各 XML インスタンスマネル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 カ無しためです。例えばドキュント要素の/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 するものココマッチしません。複数のグローノ い要素がスキーマーて定義されている場合、それらの方のどれかをデキュメント要素としている XMLドキュン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マーマレマ当でないければみません)。

複数のルート要素(ドキュメト要素)を使用するようにSPSをセルアップするコよスキーマのルート要素エトリーの右側にある…… ボ タ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。

ルート要素の選択				
スキーマで利用可能な利用可能な全てのルート要素が以下のリストに表示されます。 要素の隣にあるチェックマークは現在スキーマソースツリー内で可視化されている示しています。 グレーになっている要素はデザインで使われており、チェックマークを削除することはできません。				
スキーマソースシリー内で可視化したい要素をチェックしてください。 ■ n1:Desc n1:EMail n1:Name ■ n1:OrgChart = n1:Person n1:Person = n1:para				
OK キャンセル				

ダイアログココスキーマ内部にある全てのグロー・シル要素が表示されます。ルト要素(ドキュメト要素)とて使用することができるグローバ ル要素を選択し、OKをクックしてください。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 にないます。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にないます。これらのテンプレートは、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、代替ルート要素テンプレートのうちの1つだけー XMLドキュメトのレート(ドキュメト)要素-が使用されます。

Element-A 谷ギ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 にマッチするレトテンプレートが使用され、SPS 内にあるその他全てのレト要素テンプレートは無視されること さいます。Element-B だキュメト 要素とするドキュメト か処理された場合、Element-B にマッチするレトテンプレートが使用され、SPS 内にあるその他全てのレ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、異なるレート(ドキュメト)要素を伴う複数の XML ドキュメントに対して同じ SPS を使用することができるよう になります。

- □ 関連項目
- <u>スキーマソレーサイドバー</u>

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おか、。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

コマイを選択することことり、XML スキーマやDTD を参照するさめのダイアログが表示されます。スキーマか妥当な場合、そのスキーマかス 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こあるスキーマノースツ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) おさむ DTD ファイルが XML ファイル 理連付けられてし る場合、 XML スキーマや DTD ファイルがノーススキー 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。XML ファイルは関連付けられているスキーマが存在しない場合、XML ファイルの構造とコンテンンから 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ノ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・ジレリノースを使用してファイルを選択ませょ保存する

「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 おけおグロー・ シリハノースを使用して必要なファイルを選択おけまファ 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けっかに「URL に切り替える」 おけよ「グロ ー・ シリノノース」をクトックしてください。

Open			—
Look in:	Examples	"	
(Pa	Name	Date modified	Size 🔺
-	📾 cond-address.xsd	01/07/2014 12:10 AM	4 KB
Recent Places	🔄 Conditional.sps	01/07/2014 12:10 AM	41 KB
	📶 Conditional.xml	01/07/2014 12:10 AM	4 KB
	📾 Conditional.xsd	01/07/2014 12:10 AM	5 KB
Desktop	🕞 Conditional-Final.sps	01/07/2014 12:10 AM	47 KB
F	🕞 DBSample.sps	01/07/2014 12:10 AM	25 KB
	🜒 DebuggerClient.htm	01/07/2014 12:10 AM	9 KB
Libraries	🛃 EU.bmp	01/07/2014 12:10 AM	1 KB
	🔁 Examples.spp	01/07/2014 12:10 AM	9 KB
	ExpReport.sps	01/07/2014 12:10 AM	166 KB
Computer	ExpReport.xml	01/07/2014 12:10 AM	2 KB
	ExpReport.xsd	01/07/2014 12:10 AM	7 KB
	ExpReport.xslt	01/07/2014 12:10 AM	168 KB
Network	🔝 exterior.gif	01/07/2014 12:10 AM	18 KB 👻
	File name: ExpReport xml		• Open
	Files of type: All Files (*.*)		Cancel
	Switch to URL	Switch to Global Resources	h.

URL を使用してファイルを切り替える

URL から開くおけは保存するファイルを選択するコは以下を行ってくたさい

1. 「URL に切り替える」コマンドをクリックします。これにより、開くまは土保存ダイアログの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)。

Open			—		
File URL:			•		
Open as: O Auto	© DTD	File load	Reload		
Identification User: MyDocs	Password:	Remember between	er password application starts		
Available files Server URL: http://vietsp	ostest/	•	Browse		
✓ This is a Microsoft® SharePoint® Server					
		New Folder	Delete		
Switch to File Dialog	Switch to Global Reso	ources Open	Cancel		

- アクセスする URL をサーメー URL フィールドに入力します(上のスクレーシショナ)。サードーが 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。この種類のサーメート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. サードーが、アワードには保護されている場合、ユーザーとノアワードフィールド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. フォルダーツ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。
| Open | — |
|--|---|
| File URL: http://gd.tuwien.ac.at/vietspstest/_ca | talogs/lt/Forms/DispForm.aspx 🔹 |
| Open as:
O Auto | File load
O Use cache/proxy O Reload |
| Identification
User: TestUser Password: •••••• | Remember password
between application starts |
| Available files
Server URL: http://gd.tuwien.ac.at/ | ■ Browse |
| gd.tuwien.ac.at wietspstest | |
| | New Folder Delete |
| Switch to File Dialog Switch to Global Resources | Open Cancel // |

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」ますは「保存」ボターを使用できるようこないます。

6. 「開く」をクリックしてファイルをロードするか、「保存」をクリックして保存します。

<u>以下の点に注意してください</u>

- WebDAVをサポトするサードと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ュまゴよファイルが以前し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は、便に処理スピードを向上することができます)。また、電子出版まゴ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、ファイルアイコンココンアイルのチェッククレノチェックアナの状

⁰

態を表示するンンホルのめばす(トのスクリーンンヨット)。	を表示するシンボルがみます(下のスクリーンジ	<i>⊴⊁</i>)。
------------------------------	-------------------------	--------------

Open	
File URL: http://vietspstest/Docs/Documents/flc.	/AutoCalc.sps 🗸 🗸
Open as:	File load © Use cache/proxy © Reload
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/	Browse
His is a Miclosofte Shaler Ginte Server Documents Or Chr AutoCalc.sps Check Out Bank_Java.ur Check In	2
E Indo Check Out I Indo Check Out	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

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:

4 8 859	チェック ん状態。
	チェックアオ可能。他のユーザーによチェックアオ状態。
Š	チェックアナ不可能。ローカルに交してチェックアナ。

- ファイルをチェックアナした後、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」ダイアログのエレ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) おさよ 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はポップアップするコレテキストメニューを 使用して(アのスクリーンションケ)。

🕤 AutoCalc.sps	Persons.xml	ExpReport.xml
	0	

- ファイルが他のユーザーによりチェックアナされている場合、チェックアナすることができません。
- 他のユーザーによりアイルカ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。これによりアイルを変更す

ることないサーバーに戻すことができます。

- 変更することなしに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「チェックアナを元に戻す」です。
- ▼ グロー・ ジルノノースを使用してファイルを選択ませば保存する

グロー・ ジリノノースを使用してファイルを開くおけは保存する場合、「グロー・ ジリノノース」をクトックします。 グロー・ ジレノノースを選択 するすっかのダイアログが表示されます。 これらのダイアログは以下のセクションで説明されています… グロー・ ジリノノースの説明に関しては このドキュメトのセクションを参照してください。

XML スキーマンおける 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、おけどのデータ型も割り当てられていない場合、そのスキーマーある全ての要素が使用可能な子要素とてスキーマソノーに表示されます。例えば、email と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 といナー全てのグローン以要素が、使用可能な子要素としてスキーマソノーに表示されます。例えば、xs:string といナニデータ型を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。

- 三 関連項目
- <u>スキーマソレーサイドバー</u>

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ドキュメトルマ対して 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

- 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。 デザイン概要サイドバー 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)...
 ユーザー定義スキーマを追加(U)

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<u>ルート要素(ドキュペト要素)</u>が表示されます。
- 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>ドキュメナ 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 ツレーサイドバーのソールドーは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)。



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ニある削除アイコンをクトックするか、コンテキストメニューから削除を選択します。

- 🖃 関連項目
- <u>スキーマソノーサイドバー</u>
- 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 ませは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分さい)。

サンプル

<u>(マイ)ドキュメトフォルダー</u>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、異なるソースXML ファイルのマージの方法が説明されています。SPS は、架空の注文 (下にリストされるMergeOrder.xml) からデーダを選択します。

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がら製品を注 文するために使用されます(アのリストを参照してくたさい)。

```
<?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</ProductName>
      <Price>99.95</Price>
   </Article>
   <Article PartNum="748-OT">
      <ProductName>Diamond heart</ProductName>
      <Price>248.90</Price>
   </Article>
   <Article PartNum="783-KL">
      <ProductName>Uncut diamond</ProductName>
      <Price>79.90</Price>
   </Article>
   <Article PartNum="238-KK">
      <ProductName>Amber ring</ProductName>
      <Price>89.90</Price>
```

```
</Article>
<Article PartNum="229-OB">
<ProductName>Pearl necklace</ProductName>
<Price>4879.00</Price>
</Article>
<Article PartNum="128-UL">
<ProductName>Jade earring</ProductName>
<Price>179.90</Price>
</Article>
...
</Article>
...
</Article>
```

/Order/Item テンプレート内にユーザー定義テンプレート 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(以下のスクレージョナを参照)。(/Order/Item コンテキスト内に ある) XPath 条件式は \$ Articles//Article[@PartNum=current()/@partNum] どひます。

sp rou have ordered: sp
Conder O Item Sdiv = quantity (コンテンツ) = quantity Cave (コンテンツ) = quantity Cave (コンテンツ) O ProductName (コンテンツ) O ProductName Sdiv Sdiv O Item O Order Cave Cave

このテンプレートにより、以下のスクレーンション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か得られているのに対して、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に注目してくたさい。ProductNameノードが/Articles/Article テンプレートのエンテキスト内にで選択されている点にも注目してくたさい。

<u>自動計算</u>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(以下のスクレーンショットを参照)。スキーマソレーウィドウから 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、以下のスクレーンショット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。



188 SPS 構造

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

おは

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 スキーでを参照しているのに対して、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
- <u>自動計算</u>

6.3 モジュール SPS

SPS のグロー・シリ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 デザインこで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ント(ませまオブジェクト)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

<u>利用か可能なモジュールオブシュクト</u>のセクションゴよ SPS の様々なコンポーネトを 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 ロンポーネントが記されています。 追加されたモジュールそのものにモジュールが含まれて いる場合、これらモジュールは再帰的に SPS へ追加されます。 これにより、モジュールを広範囲なデザイン構造や複数のレベルへ展開するこ とが可能しています。

モジュールSPS を作成する

モジュール化された SPS を作成するコよ メイン SPS ヘ<u>目的の SPS をモジュールとて追加</u> します。追加されたモジュール内部にある 全ての Java Script 関数、グロー・ シレテンプレート、デザインフラグメント、そして XPath 関数が、参照元の SPS にて利用可能とひょ す。これらオブジェントはデザインソレーにて表示され、チェックボックスヘチェックを入れる/外すことで、個々のオブジェントを有効/無効にすること ができます(以下のスクレーンショントを参照)。

デザインツリー ×
BusinessAddressBook.sps
田 🔁 スクリプト
田 🗰 メインテンプレート
🗇 🖏 グローバルテンプレート 🛨
- 🕀 🗹 🚺 addr: Address-EU [Address.sps]
🕀 🗹 🚺 addr:Address-Other [Address.sps]
🕀 🗹 🚺 addr:Address-US [Address.sps]
🕀 🗹 🚺 addr:ContactPoints [ContactPoints.sps]
🕀 🗹 📢 addr:Email [ContactPoints.sps]
🕀 🗹 🌔 biz:Companies
🕀 🗹 🌔 biz:Company
■ 一 電デザインフラグメント 💌
────────────────────────────────
🛄 XPath 関数群 🚽

これらのオブジェクトはそれぞれのメカニズムコ従うかたちで、参照元のSPS にて再利用することができます。グロー・シル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。デザインフラグメートは、デザインソレーから目的の場所へドラッグする必要かありま す。JavaScript 関数は、選択されたデザインコンポーネントに対してイベント・ンドラーとしてプロ・ティウイ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 の作成のセクションを参照くたさい。

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、それそれ追加された 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、そのオブジェクトか参照元のSPS モジュール・ ールー追加されます。これらのオブジェク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・シリテンパート、デザ インプラグメート、JavaScript 関数、ページンイアナアイテム。

- □ 関連項目
- デザク概要
- · 2+-7/-2
- 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

このセクタンでは、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シェクトを紹介します。 SPS モジュールこれで、各オブジェクトをどのように利用することができるのか、おこ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、次セクションの<u>モジュールSPS の作成</u>を参照くたさい。セクションの終われては参照元のSPS モジュールこで利用 することのできないオブジェクトか示されており、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。

- <u>名前空間の宣言</u>
- <u>グロー・ジレテンプレート</u>
- デザインフラグメト
- <u>追加されたモジュール</u>
- スクレプト
- <u>CSS 7911</u>
- <u>~-፡፡//ምት</u>
- 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 のスキーマノースに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、エラーメッセージが表示されます。

スキーマツリー	x	
- 🗉 🗙 🐖 🔮		
📲 V-A		
🕞 🛛 名前空間		
xpath=default=ns :	(not set)	
addr :	http://www.altova.com/stylevision/tutorials/addresses	
biz :	http://www.altova.com/stylevision/tutorials/companies	
clitype :	clitype	
iso4217 :	http://www.xbrl.org/2003/iso4217	
ix :	http://www.xbrl.org/2008/inlineXBRL	
java :	java	
l link :	http://www.xbrl.org/2003/linkbase	
n1 :	http://www.altova.com/stylevision/tutorials/myaddresses	
n2 :	http://www.w3.org/1999/xhtml	
per :	http://www.altova.com/stylevision/tutorials/personal	
sps :	http://www.altova.com/StyleVision/user-xpath-functions	
xbrldi :	http://xbrl.org/2006/xbrldi	
xbrli :	http://www.xbrl.org/2003/instance	
xlink :	http://www.w3.org/1999/xlink	
xsi :	http://www.w3.org/2001/XMLSchema-instance	
🗇 🚰 \$XML(メイン)		
│ □ / ルート要素 …		
⊕ 🕀 🌔 n1:AddressBool	k j	
🛛 🖽 🖏 グローバルテンプレート	• •	
田 T 全てのグローバル要素		
田⁄⁄⁄ 全てのグローバル型		
■ 電デザインフラグメント 💌		
🛄 XPath 関数群 🔒		

上のスクレーンショットでは、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ノ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・ジレテンプレート

追加されたSPS モジュールのクロー・シレテンプレートは参照元のSPS モジュールにて使用することができ、デザインソノーサイドバーに表示されます(以下のスクノーシンヨナを参照)。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・シレ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

バルテンプレートをオー・ドーライドする場合、グロー・シルテンプレートエトリーの隣にある アイエンをクリックしてください。新しいグロー・シルテンプレートを作成する要素おけば属性を選択してください。ホ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もできます。「OK」をクリックすることで、新たなグロー・シルテンプレートと追加されたモジュ ール内部のグロー・シルテンプレートのどちらを有効化するのかを選択するようご捉されます。選択結果にこと、新たコプレートと追加されたモジュ レート、おけば追加されたモジュール内にあるグロー・シルテンプレートが有効化されます。2つあるグロー・シル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ンオルトの処理(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<u>デザインアグメント</u>は 参照元のSPS にて利用することができ、SPS は、<u>デザインソノーサイドバー</u>に表示 されます(上のスクノーンショナを参照)。 デザイン内部にデザインフラグメナを挿入する際には、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は参照元の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にないます。勿論これらのモジュールによい構成されるコンテンンモデルは、参照元 SPS モジュールのノーススキーマイン対して妥当なものでなければないません。モジュールの確認と管理は、デザイン概要サイドバーで行うことができます。

スクレプト

追加された全てのSPS モジュールにあるスクレプトは参照元のSPS にて利用することができ、これらのスクレプトは、デザインソノーサイドバー に表示されます。モジュールに追加されたスクレプ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 きるようこないます。

CSS スタイル

SPS モジュールに追加されたプロー・シルスタイルは、グロー・シルスタイルとして参照元のSPS へ渡され、 スタイルレポントリサイド・バー内にレ ールならひえ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・バートにも表示されます。 同様に、 SPS モジュールへ追加さ れたか部 CSS ファイルも、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クトを参照元の 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、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 コンテンノモデルは、参照元 SPS のコンテンノ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 使用できないという点に注意してくたさい。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。 新たなスキーマイコは、追加されたグロー・シリテンプレートのコンテンノ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: これらのファイルに対する参照か無視されます。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にて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ありません。
- 三 関連項目
- デザイン概要
- モジュールSPS の作成
- テンプレートとデザインフラグメト
- スケノプトの使用

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. <u>追加する SPS モジュール</u>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<u>モンユールを追加</u>。
- 3. 必要に応じて追加されたオブジェクトを有効/無効にする。
- 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あります:

- 1. <u>参照元のSPS モンュール</u>にて使用することのできるテンプレートは、<u>グロー・シレテンプレート</u>だけです。つまり<u>追加される SPS モンュール</u>にて再利用した テンプレートは、グロー・シ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、参照元のSPS のノーススキーマ. にお定義されるエンテンソモデル内部で妥当でなければむません。追加されたテンプレートがSPS のメインスキーマにて定義されたエレテンソ モデルに含まれていたい場合でも、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 以下のどれかどなります:

- 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こてコンテンソモデルを定義する場合、使用される<u>名前空間</u>によりノードの展開名が決定されるため、慎重に操作を 行う必要があります。

<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まデザクと概要 にて新たなモジュールを追加コマイ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。

▼ Portable XML Form 押か込まれたつっくルの構成	
押め込まれたつって心の構成	
2±07/2364.022711700/140200	
埋め込まれたファイルの更新/展開	
マリース	
🚰 \$XML (メイン) 🤇 🚺	3
スキーマ OrgChart.xsd (3
作業用 XML OrgChart.xml 《	3
テンプレートXML	3
XML 署名 Authentic (こて有効 (3
<u>新しいソースを追加</u>	
▽ モジュール	
<u>新しいモジュールを追加</u>	
▽ CSSファイル	
🐻 Samples.css 🔹 🕻	3
<u>新しいCSSファイルを追加</u>	
▽ パラメーター	
💲 \$Year 2012	3
新しいパラメーターを追加	
▽ XSLT ファイル	
📾 NameAttrUpper.xslt 🤇	3
<u>新規XSLTファイルを追加</u>	

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クト かデザインソ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も表示されます。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も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

ポートされたモジュールが、どのモジュールによりインポートされたものなのかは、その、間接的にインポートされたシ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や間接的にんポートされたモジュールを Style Vision で開くには そのモジュールを右クルクして、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たさい。

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)のモジュールスされ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 ゴよ、そのモジュールをデザイン概要にて右クルクし、コンテキストメニューから上 へ おゴゴ下 へを選択します。

スクリプトやグロー・シリテンプレート、デザインフラグメント、そしてページン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 モジュールは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ンナトは、(直接的、おけば間接的に追加されたかにかかわらず)参照元のSPS へ追加され、デザインソー内の対応する箇所(スクレプト、グロー・シレテンプレート、デザインフラグメナト、XSLT テンプレート、XPath 関数) 以下に表示されます。これらオブジェケ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、そのオブジェケトを 有効/無効にすることができます。オブジェケトを無効にすることで、そのオブジェケトを SPS から実質的に消除することができます。



上のスクノーンショナでは、AddressBook.sps モンユールにて使用されている全てのグロー・シレテンプレートかグロー・シレテンプレート以下に表示されて、ます。他のモンユールを介して追加されたモンユールは、それか道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ブジェクトは黒色で表示されます。スクノーンショナでは、addr.Email というプロー・シレテンプレートだ しが、AddressBook.sps 内に作成され、その他のグロー・シレテンプレートは全て別のモンユールを介して追加され、オブジェクト名の隣に定義さ れたファイルの名前か表示されます。 addr:Email とう名前のグロー シリテンプレートかつあることご注目してください。1つは参照元のSPS (AddressBook.sps) にて作成され たきので、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 シレテンプレートが、同じ名前の(名 前空間)展開名を持つ場合、そのうちの1つだけを同時に有効にすることができます。チェックボックスにより、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 シレ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 モジュールにて作成したグロー シレテンプレートのオー・デークドや、(ii) ある要素に対して適用されるグロー・シレテンプレートが 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運用な機能とないます。。

現在のSPS にて定義されたグロー・シレテンプレートは、選択した後に削除ボタンをクリ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にて定義されたグロー・シレ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・シレテンプレートを削除する必要があります。

個々のスクリプ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クトは、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ルオブジェクトを <u>有効にする</u>必要があります。

グロー・バレテンプレート

- 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。

<u>デザインフラグメント</u>

デザインプラグメトを使用するコま デザインソーからメインテンプレートやグロー・シリ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 ンプラグメトをドラッグします。 デザインプラグメントが、 適切なコンテキストノード ヘドロップされるようししてくたさい。 詳細についてしま 次を参照 してくたさい デザインプラグメント。

<u> スクリプト</u>

デザインフラグメトを使用するコよ、デザインソレーからメインテンプレートやグロー・シ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 ンフラグメートをドラッグします。デザインフラグメートが、適切なコンテキストノード・ヘドロップされるようにしてくたさい。詳細については<u>プロ・ティサ</u> <u>イドバーから特定のイベートにさしてセートすることができます</u>。

□ 関連項目

- デザン概要
- <u>利用が可能なモジュールオブジェクト</u>
- サノプル: Address Book
- · 2+-7/-2
- テンプレートとデザインフラグメント
- <u>スクレプトの使用</u>

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 Vision2021 Style VisionExamples Tutorial ModularSPS はモジュール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とされているアドレス帳の プロジェクトファイルは構成されます。サンプルにより、モジュールとかどのように実装されるか確認することができるとともに、モジュールとを行う利 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:

- · <u>スキーマファイル</u>
- <u>XML データノース</u>
- <u>SPS ファイル</u>

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 が要素として作成しなければならない点に留意してくたさい。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) をインポートすることで構成されています。BusinessAddressBook.xsd スキーマによと会社用のコンテンソモデ ルが与えられ、PersonalAddressBook.xsd スキーマによ個人用のコンテンソモデルが与えられます(以下のスクリーンショナを参照)。



両方のスキーマーて連絡先詳細のコンテンソ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 スキーマイゴはEU、US、そしてその他(Other)の地域に対応したアドレス帳のエンテンソモデルが定義された Address.xsd スキーマが含まれています(以下のスクリーンショナを参照)。



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、スキーマイゴ「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、ます。このファイルでは、AddressBook 要素内に companies とpersons 要素が 子要素とて含まれて、ます。これら2つの要素のエンテンソモデルは、それぞれ BusinessAddressBook.xsd と PersonalAddressBook.xsd スキーマファイルにて定義されて、ます。

BusinessAddressBook.xsdとPersonalAddressBook.xsdスキーオこそれそれ対応する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 構造

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.xsd をスキーマとして使用しています。

<u>Address.sps</u>

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。

Oper:Persons Oper:Person Oaddr:ContactPoints Oaddr:Address SXML On1:AddressBook (コンテンツ) (Oaddr: Address O addr:ContactPoints Oper:Person Oper:Persons On1:AddressBook < 🖼 \$XML

Address 要素だけが処理されるため、出カプレビューにて 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

このSPS ではつのモジュールがんポートされます。グロー・シレテンプレートが他のグロー・シレテンプレートやメインテンプレート内部で使用されている点に注目してくたさい。

- 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 と 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・シリ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
- <u>モジュールSPS の作成</u>
- <u>x+-v/-x</u>
- テンプレートとデザインフラグメント
- <u>スクレプトの使用</u>

6.4 テンプレートと デザインフラグメント

デザイズキュ火トはテンプレートには構成されるため、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つのメインのテンプレートとオプタンとなっている1つ以上グローバルテンプレートには構成されます。グローバル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内で再利用ことができるテンプレートです。

このセクションでは、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ン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ノティではなく、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、とデザインプブメトと、複数のテンプレートを、SPS に含めることができます。 <u>ツールドーアイエン</u>と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ができます。

三 関連項目

- <u>デザインの作成</u>
- <u>_____</u>

6.4.1 メインのテンプレート

<u>メインのテンプレート</u>により、出力の構造が決定されます。つまり、デザイノー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。これシノードが処理されるテンプレート は<u>ローカルテンプレート</u> と呼 「おます。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コ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 <u>グロー・ いテンプレート</u> への参照が行われることもおります。この場合、そのノードにつけしてグロー・ いテンプレート か実行 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、次のコ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ントリーポイントはスキーマの<u>ドキュメントノード</u> どむます。StyleVision ではオプションとして複数のルート要素 (<u>ド <u>キュメント要素</u>) を選択することもでき、メインテンプレート内部で、アクティブな各ドキュメント要素に対して <u>ローカルテンプレート</u>をセントすること 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ントのドキュメント要素に対して定義されたテンプ レート どむます。</u>

- 三 関連項目
- <u>デザインの作成</u>
- <u>テキストとてXML コンテンを挿入</u>
- <u>グローノリテンプレート</u>
- <u>デザインアデグメト</u>
- <u>テンプレートフィルター</u>

6.4.2 グローバルテンプレート

グロー・バテンプレートは、任意のノードやスキーマ内の型、おけはXPath条件式により指定されたノードに対して定義することができます。

グロー・シレテンプレートは選択されナノードや型の処理命令について適用され、<u>メインのテンプレート、デザインフラグメート</u>、おけまその他のグ ロー・シレ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し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。このようにして、ある定義を複数回再利用することが可能によいます。グロー・シレ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:

- <u>メインのテンプレート</u>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リックし、グロー・シレ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) おは(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 テンパや型には、 グロー・ ルテンプレート のノードや型が含まれている。

様々な要素や複数の場所でノード(おけま型)が出現する、おけお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、おけまblockquote要素内にあるかどうかけかからず、 para要素の書式が常に同じものけなるようにすることを考えてみましよう。para要素に対してグロー・シルテンプレートを定義して、メインのデン プレート にて para要素に対するグロー・シル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がヴロー・シル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がヴロー・シルテンプレートを参照するようは、、//chapter/titleを追加した後に(コンテング) おけま(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コンテンンが、利用可能なグロー・シルテンプレートやデンオルトテンプレートにより処 理されるようけないます。更に、グロー・シル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(複合型おけま単純型)により定義された出現のうち、グロー・シルテンプレートを呼び出す全ての出現にお いて、グロー・シルテンプレートのようにます。.

グロー・シレ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)。

スキーマツリー ×
-E - 🙆 🗙 🐖 🚰
🖷 ソース
田田 名前空間
🕀 🚰 \$XML (メイン)
□
🗇 🕄 グローバルテンプレート 🛨
-⊕♥() ts:bold [TextState.sps]
- 🕀 🗹 🜔 ts:italic [TextState.sps]
🖽 🗹 🜔 ts:underline [TextState.sps]
🛛 田 T 全てのグローバル要素
田⁄⁄/ 全てのグローバル型
■ ■ デザインフラグメント 🛨
└──── <u>f</u> XPath 関数群 ┣

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、メインのテ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ンタン作成されたノード名を含むシーを操作して表示されます。: (単一) おけは(複合型)。グロー・シレテンプレートは、下のスクレーシンタット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。 おっ、 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・シレ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 ー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 <u>XPath フィルターの編集</u>を選択します。 <u>XPath 式 の編集ダイ</u> プログ が表示され、目的の条件式を入力することができるようプンリます。

再帰的なグロー・ゾレテンプレート

グロー・シレテンプレートは再帰的に作成することも可能で、グロー・シレ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・ティーゴ値をセル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・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・ティーで値をセルす 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が行われた場合、エラーが返されます。

グロー・ジレテンプレーをローカルレニピーする

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るという 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・ジレ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・ジレ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・ジレテンプレートと複合型のグロー・ジレテンプレート

グロー・シレテンプレートはよ単純型と複合型の2種類がおます。複合型グロー・シレテンプレートは下位互換性を保証するさめのたので、バージョン2006 より前のStyleVision にて作成された SPS のグロー・シレ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・デジョンでは、通常複合型として、そのグロー・シレテンプレートが開かれます。 複合型のグロー・シリテンプレートと単純型のグロー・シリテンプレートとの違いは、グロー・シリテンプレートに対して作成されたノードの処理方法 にあります。ノードの最初のインスタンスカドキュメントにて見つかけた場合、そのノードの直後にある全てのインスタンスか複合型のグロー・シリテ ンプレートにより処理されます。単純型のグロー・シリテンプレートの場合、個々のノードインスタンスか発見されるごとこ処理が行われます。

グロー・シ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ナ ガリ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ナ ガリテンプレートは、自動的に複合型のグロー・シレテンプレートへ変換されるとら点に留意してくたさい。これにより、複合型のグロー・シレテ ンプレートと同様の処理が行われます。単純型のグロー・シレテンプレートへ戻したい場合は、(定義済みのフォーマナの外側・ヘノードをドラッ グし、定義済みフォーマナを削除することで)定義済みのフォーマナ を削除するか、デザインソノ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あります。単純型のグロー・シレテンプレートから複合型のグロー・シレテンプレートへの自動変換を回避するこ は、単純型のグロー・シ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ナ や改行を追加するようにしてくたさ い。

モジュール SPS におけるグロー・ シレテンプレート

<u>SPS モンュール別のSPS モンュールシ追加された</u>場合、追加されたモンュール内にあるグロー・シレテンプレートを、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</u>のセクションを参照ください。

□ 関連項目

- <u>テキストとてXML コンテンを挿入</u>
- その他のエレテノツ
- メインのテンプレート
- デザインフラグメント
- デノプレートフィルター
- モンコール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に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まし、以下のスクレーシショナでは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 トアイエンが示されます。ユーザー定義テンプレートが特つ柔軟性により、テンプレートを作成することができます。

<u>▲ /n1.Name</u> (コンテンツ) ▲
👼 for \$i in n1:Office[Loc=NY] return \$i[1]
<u>♣123</u> →=(自動計算) <mark>♣</mark>

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、全てのコ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こなります。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エム、表示されます。
- XPath 2.0 とXPath 3.0 のfor コンストラナトにお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

内でエンテンソプレースオルダを使用することはできません。これは、(エンテンソプレースオルダーにおり生成される)xsl:apply-template 処理命 令が、(原子型の値ではなく)ノードアイテムにしか適用できないためです。そのカヤル、箇条書きといた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、以下のスクノージョント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 定義テンプレート)には選択することで、XPathの スセレクターメカニズムを使オ・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」なり、ノードの選択に対してフィルタレグや条件をセルすることが可能になります。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」なります。例え ば、//Office/Department[@Location="NY"]とうXPath条件式により、NYとう値を持ずこLocation属性が含まれるDepartment ノード 扫が選択されます。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管まれる場合、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ブループ化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た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 select="Dept"> </xsl:for-each> </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 並び替えしよ 各 Office 内部で 行われる。
有り	<xsl:for-each select="/Org/Office/Dept"> </xsl:for-each 	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

- デザイン要素の挿入ツール・トーンあるユーザー定義テンプレートを挿入アイエンをクリックし、デザイン内のテンプレートを挿入する場所を クリックします。まけは、テンプレートを挿入する場所を右クリックし、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 <u>XPath 式を編集</u>ダイアログにて XPath 条件式を入力し、**OK**をクレクします。 XPath 条件式のエンテキストノードは、クレックされたノードになる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、ターゲル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たは、場合(つま)複数の階層レベルにて分割されている場合)、ループによど格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、(結合まずはよう)割により、返された両者のノードセナトに違いはおりません。 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、: ノードテンプレートオペレーションを参照してくたさい。

テンプレートマッチの編集

任意の、ユーザー定義ませば通常の)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デ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リックレ、デ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」をクリックします。

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コキンーケンスコネルしてユーザー定義テンプレートが作成された場合、 スキーマソノ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/2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/24により参照されるのか決定 するよう促されます(以下のスクノーシショナを参照)。

デザイン要素の作成
現在のロケーションに対するコンテキストノードを決定することができません。 以下から操作を選んでください:
● 単一のテンプレートを作成
○ 絶対スキーマバスに対してテンプレートを作成
□ このオプションを常に使用
OK ++>ンセル

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なっています。「<u>ツール」オテンョンダイアログ」</u>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。

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、カーソル位置に、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オ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(以下のスクレージョナ 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)「動的な)コンテンンを含めることができるほか、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。

以下の操作によ変数テンプレートを挿入することができます:

- 1. 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カジ挿入 | 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 リーンショントを参照)。

変数テンプレートを挿入		
新規テンプレートに対する変数を追	選択してください:	
\$name (読み取り専用)		
I		
	OK	キャンセル

- 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> か表示されます。変数テンプレートに対して追加する変数を選択してくたさい。
- 4. OK をクリ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(以下のスクレージョナを参照)。以下のスクレージョナでは、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。次も参照してください、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、そのノードテンプレートの開始ませま終了タグ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確認することができます。

以下ではコ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:

- <u>/u-virvlv-h</u>
- <u>テンプレートの一致</u>
- XPath 7/11/5-
- 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
- 編集、タグで囲む、変更

以下ではこれらのメニューコマンドについて説明します。ノードテンプレートによっては利用できないロマンドもある点に注意してください。利用できないロマンドはコンテキストメニュートこてグレーアサンされます。

グロー・シリテンプレート:設定、使用、ローカルにコピー

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、以下のコマンドによりグロー・シ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。

は関する詳しい説明は、次のセグョンを参照してくたさい、グロー・ショテンプレート.

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、このコマンドには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、<u>XPath 式 の編 集ダイアログ</u> 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 <u>ユーザー定義テンプレート</u>へ変更することもできます。

XPath フィルターの編集/クア

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レクを行うことができます。XPath フィルターは<u>グロー・ シレデンプ</u> レートに対して適用することができます。

デフォルトでは、ノードテンプレートが作成されナノードに対応するノード(要素おけば属性)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 とう式を持ずこ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] どなります。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」なります。

フィルターは、デザイン内にある任意のノードテンプレートならびに複数のノードテ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'] といすこXPath 条件式により、社内のUS オフィスユ 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 のノードテンプレートにて、それそ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しつさして XPath 条件式を適用することで、ターゲナトとなるノードセナ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ックして、XPath フィルターの編集を選択します。引用句や角かって、デトターなしで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、そのノードテンプレートの開始タグにフィルターシンボルは表示されます。以下のスクレージショナでは、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。

<u>
Osynopsis</u>
(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、<u>グループ化のセグタ</u>メーズ記述されます。
- 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、並べ替えのセグタ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て記述されます。
- 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、目次(TOC)と参照.

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 し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ひト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) タグガゴ 消除

このコマトドにより、選択されたテンプレートやフォーマトタグだきが削除され、子孫ノードやそのフォーマトタグが削除されることはありません。 あるタグに含まれる子孫ノードなどを削除することなしにフォーマトタグや親要素のタグを削除するために使用することができます(タグが選択 された状態で削除コマノドが実行されると、これら全てか削除されます)。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不正とな る可能性があることに注意してくたさい。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

これらコマド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: <u>段落、特殊な段</u> <u>落、リスト、ハイパーレク、条件、目次ブクマークとレベル</u>
- ・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、変更機能のセグシンを参照くたい。

🖃 関連項目

- <% SPS% ファイルエレテンソ
- テンプレートとデザインフラグメント
- <u>XPath ダイアログ</u>

6.4.6 デザインフラグメント

デザインフラグメントは、ドキュメント内の様々な箇所で再利用することができる。シンを作成するために使用することができます。以下に使用するための手順を示します:

- 1. デザインフラグメトのエンテンンを入力する
- 2. デザインフラグメントのエレテンンを入力する
- 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トを挿入する

デザインフラグメントのコンテンンを入力する

以下の操作によげザインフラグメトを作成することができます

 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 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<u>デザインフルター</u>ツールバーにある<u>表示、隠す</u>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。

 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。

日電デザインフラグメント 🛨

- 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ア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、デザインノーやスキーマソーから挿入。

デザインフラグメントのコンテンンを作成する

デザインフラグメトテンプレートのエレテンソは、その他のテンプレートと同様の方法で作成されます。静的エレテンンを挿入するコよカーソルをデザインフラグメートのテンプレート内部へ移動させ目的の静的エレテンンを挿入します。動的コレテンンを挿入するコよ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へドラッグします。

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)。

'EmailDocNode' バラメーターを定義
SXML On1:OrgChart On1:Office
<u>On1:Name</u> (コンテンツ) On1:Name
$\begin{array}{c} \hline \textbf{O}_{n1:Name} & (\neg \cancel{2}, \neg \cancel$
=(自動計算)
On1:Person
Onl:Department ? 1
On1:Office On1:OrgChart SXML
'EmailPerson' バラメーターを定義
On1:Person M On1:First (コンテンツ) On1:First On1:Last (コンテンツ) On1:Last : =(自動計算)
Whiteerson

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 カが生成されます。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 <u>メトフォルダー</u> 以下にあるC:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial\DesignFragments 内のEmail.sps ファイルに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インフラグメントに対する、ラメーターを参照ください。

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<u>デザインフラグメントの</u> <u>コンテング作成する</u>を参照ください。

その地にも、デザインフラグメント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

デザインフラグメトはそれ自体を呼び出すような、再帰的なかけっちで作成することもできます。Authentic View におお (無限ループを回避 するけっか、呼び出しの階層を制限するプロ・ティをセル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・ティ」)にある Authentic タブ内のテンプレート呼出し階層上限プロ・ティニで値をセルすることができ、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・ティ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コン選択をして、デザインソノーの削除ツールレーアイコン 🔀 をクトックします。

モジュールSPS におけるデザインフラグメト

SPS モンュールが別のSPS モンュールへ追加された場合、追加されたモンュール内にあるデザインプグメントも、参照元のSPS にて利用することができるようてないます。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たさい、モンュールSPS。

サンプルファイル

サンプルSPS は <u>(マイ)ドキュメントフォルダー</u>にある、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。

三 関連項目

- <u>テキストとてXML コンテンンを挿入</u>
- その他のエレテンソ
- メインのテンプレート
- グロー・ジレデンプレート
- デノルートフィルター
- EJJ-JUSPS

6.5 XSLT テンプレート

XSLT ファイルを SPS ヘインポーすることで、ファイル内部にある XSLT テンプレートをグロー シレテンプレートとして使用することができるよう さいます。 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 つがインポートされた XSLT テンプレート 内のノード ヘマッチした 場合、 インポートされた XSLT テンプレート かそのノード へ適用されます。 インポートされた XSLT ファイルに名前付きテンプレート かきまれる 場合、 デザイン内部にそれらを配置することができます。

<u>×</u>E

• 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 ファイルをイパ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)。

テサ	心昼夏 x
\square	ソース
\square	モジュール
\square	CSSファイル
\square	パラメーター
∇	XSLT ファイル
	ExpReport.xslt 📀
	<u>新規XSLTファイルを追加</u>

 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)。

デザインツリー ×
🚇 • 🛛 🗶 🐖 🚰
B.sps
田 🔁 スクリプト
田 🗰 メインテンプレート
🖽 🕰 グローバルテンプレート 🛨
田 🛱 デザインフラグメント 🛨
🕀 🖏 XSLT テンプレート
match - /
match – italic
match - strong
name – BytesToKBytes
<u>援</u> XPath 関数群 🕌
スキーマツリー デザインツリー
イポートされたXSLT テンプレートには2種類あります: (i) マッチテンプレート(match により表示)と(ii) 名前付きテンプレート(name により表示)。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 属性の値ともに表示されます。

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・ジレテンプレートが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に優先順位か高くなるため、 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)の優先順位か最も低くなり、2番目にインポートされたファイルの優先順位かを番目に低いものとないます。

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(以下のスクノーンショナを参照)。

<xsl:call-template name="BoldItalic" /> -----

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ートとその他のドキュメートとうかたちで、複数の出力ドキュメートを生成するようにデザインすることもできます。この機能を使用することで、出力のモジューリルを行うことが可能になります。新規ドキュメートを挿入することで、出力ドキュメートをデザイン内に作成することができます(以下のスクノーンショートを参照)。各出力ドキュメートのコンテンソは、新規ドキュメートテンプレート内部 「工配置されます。

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 プロパティを編集 ヘッダーワッターを追加
Sdiv Table of Contents: Chapters and Their Sections
<mark> ≸div ^ペ [●]'MyTOC' (num-lvl</mark>): (テキスト参照)()(ページ参照) ペ [●]'MyTOC' [●] [●] div
<mark></mark>
<mark>[§div 〉 [☆]Ψ'MyTOC'</mark> (num-lvl): (テキスト参照)()(ページ参照) ◇Ψ'MyTOC' く§div
く <u>新規ドキュメント Document1</u>

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リル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つの文章にまとが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テ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ントアップする方法にこしては、新規ドキュメントテンプレートとデザイン構造のセクションにて

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

- <u>新規ドキュメントテンプレートの挿入</u>
- 新規ドキュメト テンプレートとデザイン構造
- <u>新規ドキュメント 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
- ドキュメント プロ・ティとスタイル
- 三 関連項目
- <u>出力内のHTML プラグメント</u>

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、新規ドキュメントテンプレートをSPS デザイン内部に配置することができます:

- ・新規出カ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。

コンテンルを含む新規ドキュメトテンプレートを以下のスクレーンショントに示します。

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 プロパティを編集 ヘッダーワッターを追加
Solution Sections
^{~↓-} [§] div ^{~↓'MyTOC'} (num-lvl): (テキスト参照)()(ページ参照) ^{~↓'MyTOC'} (§div)
 <u> ↓'MyTOC'</u> (num-lvl): (テキスト参照)()(ページ参照) <u> ↓'MyTOC'</u> ↓'MyTOC' ↓
[●]

上のスクレーンショナでは、以下の点に注目してくたさい

- 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</u> ドキュメト テンプレートのURLのセクタンにて記述されます。
- 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。

三 関連項目

- 複数のドキュメトの出力
- 新規ドキュメト テノ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>
- 出力ドキュメントファイルのプレビュー
- ドキュメト プロ・ディとスタイル

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、ソースドキュメント内にある適切なノードにて個別の新規ドキュメン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ントをデザインする際には、<u>出力の構造</u>と入力(ソースXMLドキュメント)の構造</u>の両方を考慮す る必要があります。

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、この新規ドキュメントテンプレートの外にある全てのデザインコンテンンが別のドキュメントへ自動的に割り当てられます。このX新規ドキュメントテンプレートとお別のドキュメントがメーンの出力ドキュメントどなり、 StyleVisionの出力プレビューではメイン出力ドキュメン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」コマンドを使用することで)生成された出カドキュメントでは、メイン出 カドキュメントの名前が、生成されたファイルの保存コマンド、お出力ドキュメントを生成する際に割り当てられる名前どなります。メイ ンドキュメント以外の出力ドキュメントファイルコ使用される名前は、それぞれの新規ドキュメントテンプレートの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よいます。以下にある2つのケースが考えられます:

- 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。。
- 三 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
- <u>新規ドキュメト テノプレートのURL</u>
- 出力ドキュメントファイルのプレビュー
- · ドキュメント プロレティとスタイル

6.6.3 新規ドキュメント テンプレートの URL

このセクションでは、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ノ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1度12 地理される場合(<u>前のセクション</u>を参照)、テンプレートのURL プロ、ティを静的な URL 1指定す ることができます。以下のスクレーン・ヨナーでは、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12 行われます。URL 13は Table Of Contents と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なります。ファイ ル名の、ペンプレフィックスは与えられていない、サムダ、メイドキュメートと同じディレストリにファイルが注成されます(詳細については<u>複数のド キュメートの出力とプレビュー</u>を参照くたさい)。その他にも、URL 12 次が含まれる場合、出力ドキュメートが、ペにより指定された場所へ保 存されます。

新規ドキュメント 'TableOfContents'
イニシャルドキュメントセ <u>プロパティを編 ヘッダー/フッターを</u> クション <u>集</u> <u>追加</u>
ETOC Sections Sections Secti
<mark>◎Ψ=</mark> [§] div <mark>◎Ψ'toc'</mark> (num-lvl): (テキスト参照)()(ページ参照) <mark>◎Ψ'toc'</mark> [§] div
<mark> </mark>
<mark> </mark>
新規ドキュメント 'TableOfContents'

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か生成される場合(<u>前のセグシュン</u>を参照ください)、テンプレートのURL プロケテをXPath 条件式を使って動的な URL とする必要があります。以下のスクレーショナトでは、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います。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少ます。 topic 要素に対する各イタレーションにて、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 要素にて新して新たなドキュメートが作成されることではます。各ドキュメートには、トピックヘッダーのテキストである) body/header/para 要素から得られた名前が個別に与えられます。.

? 1 2 3 Otopic 4- 新規ドキュメント 'Document1'
イニシャルドキュメントセ <u>プロパティを編(ヘッダー/ワッターを</u> クション ^集 … <u>追加…</u>
Obody Oheader ↓ ™'toc' ↓ ™'toc' Opara
^{§div} (num-lvl): (コンテンツ) ^{§div}
Opara Oheader
(その他のコンテンツ) (Obody)
新規ドキュメント 'Document1'

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とり名前(Xは整数)の静的なテキスト文字列をデフォルトとする URLとともに作成されます。URLを編集するは、新規ドキュメトテンプレートを右クリックして、URLの編集コマンドを選択します。プロ ノティダイアログが表示され、ファイル拡張なしのURLプロノティのフィールドに値を入力することができます(以下のスクレーンショントを 参照)。

プ	อパティ		
	년 - 년 📲 🗮 🗙		
	属性		<u> </u>
	日 新規ドキュメント		
	ファイル拡張無しの URL	C:¥TableOfContents	
			$\overline{\mathbf{v}}$
	OK	キャンセル	

静的なURLを入力するコよ、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、値フィールドをクルクル たあ出こ、プロ・ディダイアログのソール・イーあるXPath ポタンをクルクして、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 条件式のエンテキストノードは新規ドキュメイトテンプレートが挿入されたノードしています、(ii) XPath 条件式にプレ フィックスを与えるコよ (例えば concat('C:¥MyOutput¥', body¥header¥para) といすっかすったのXPath の concat() 関数を使用してく たさい。この列にある条件式により、C:¥MyOutput¥filename とらURL 文字列が注成されます。出力フォーマトにつだい、適切なファイル 拡張子が自動的に生成されます。

ドキュメントへのレンク

<u>ブックマークとハイパーレク</u>を使用することで、複数の出力ドキュメント間にレン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ペッダーを含む新規ドキュメン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、他のドキュメントから作成した ブックマークへのハイパーレクを作成することができるようになります。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かある場合、 ブックマークの名前か動的に生成されるようにする必要かあります。そうでは、場合(静的なブックマーク名か与えられた場合)、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でいます。

<u>目次(TOC)</u>もドキュメントへのアン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アンプを作成し、それぞれの出力ドキュメントからTOC へ戻るアンプを作成することができます。

- 三 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
- 新規ドキュメント テンプレートとデザイン構造
- 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

6.6.4 プレビューファイルと出力ドキュメントファイル

デザイボキュメトの出力プレビューでは、デザイノニて個別のドキュメトとして指定された各出力ドキュメトを確認することができます。(以下のスクノーシンヨットを参照)。



上のスクノーンショナでは、複数の出力ドキュメン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か示されています。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タンをクトックすることで、おけは(ii)コンボボックスのドロップダウノ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トを参照)。ドロップダウノノストのアイテムゴは、URL 全体(パスとファイル 名)か表示される点に注目してくたさい。

プレビューファイルの場所

プレニーファイルは、デフォルトで SPS ファイルが作成されるディレクトリニて作成されます。デフォルトの設定は、(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 タタブニて変更することができます(以下のスクレーショッ 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 マカ管まれてい る場合、この 文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ます。指定された場所が見つからな、場合、エラ ーか返されます。出力ドキュメント間のレノを作成する場合、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。

 3 プロパティ
出力 XSD/XSLT/XBRL イメージ パス 解像度 Authentic
以下のディレクトリにプレビューファイルを作成
◎ SPS ファイル(S)
◎ メインス キーマ ソースの作業用 XML ファイル(X)
追加で生成されたファイルの場所
◎ デフォルト (メイン出力ファイルのフォルダー)
◎ Windows 一時フォルダー
◎
ファイル パスを SPS ファイルに保存
SPS ファイルと同一のまたはそのサブフォルダーにあるすべてのファ ◎ イルを SPS ファイルからの相対パスとし、それ以外は絶対パスと する
○ 可能な場所すべで SPS からの相対パスを使用する (他のディ スクまたは外部にあるファイルは絶対パスで表現されます)
スキーマ、作業用ならびにテンプレート XML ファイル、SPS モジュー ル、CSS ファイルなどに適用されます。
OK キャンセル

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、ソナプレビューを行うかりに一時的に使用さ れるファイルの配置場所を指定することもできます。新規ドキュメトテンプレートのURL に、なか含まれる場合、そこで指定された、なか使 用される点に注意してくたさい。

出力(パン)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えその他生成されるファイルの場所が、ポップアップウィンドウェて表 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、メイドキュメトファイルが保存されます。新規ドキュメ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 につびか管まれていないドキュメントも、メイド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 につびかり入力された場合、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。

三 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
- <u>新規ドキュメナ テンプレートのURL</u>
- · <u>ドキュメト プロノティとスタイル</u>

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、ティならびパギ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、各ドキュメントのイニシャルドキュメントセク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オプロ、ティならびパギキュメントスタイルタブニで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には、編集を行うドキュメントのイニシャルドキュメントセクションのみイトル、トーさあるプロ、ティの編集リングをクリックしてください。ドキュメントプロ、ティビキュメント スタイルは、その出力ドキュメント全体について、近日されます。

プロパティ	
プロパティ ドキュメントプロパティ ドキュメントスタ	17JU
占 📲 📲 🚠 🗙	
属性	値 🔼
🕀 Authentic	
⊕ HTML	
田 イベント	
日 ドキュメントプロパティ	
キーワード	StyleVision, ドキュメンテーション
9-1 hu	製品ドキュメンテーション
著者	Altova
田共通	
	7
OK	キャンセル

ドキュメトプロ・ティタブでは、プロ・ティのドキュメントプロ・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・トーン対応したプロ・ティース保存されます。例えば、HTML 出力フォーマ・トの場合、HEAD 要素の META タグトこてプロ・ティか保存されます。 ドキュメトスタイルに関する詳細については CSS プロンティ値の設定のセクションを参照くたさい。

三 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
- <u>新規ドキュメナ テンプレートのURL</u>
- 出力ドキュメントファイルのプレビュー

7 高度な機能

SPS デザインの基本的なコンテンパ構造の作成方法については、SPS ファイルコンテンツとSPS ファイル構造 にて記述されます。しか し、ソースデータ「あるコンテンパ構造を特定の形式に修正しなけれ」なよびは ことも多々 みます。 例えば、 (個人情報が含まれる) ノード のグループを、 苗字といた特定の条件により並べ替えるケーズ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ごグルー 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。 Style Vision にはよい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:

- 自動計算: 自動計算とはXPath をベースリュナンパフフルなメカニズムで、(i) 修正されたデータを出力にて表示するおうたなく、
 (ii) 自動計算の結果により、XMLドキュメント内にあるノードを更新することができます。
- <u>条件</u>:データ構造や、XMLの値、おさはXPath条件式の結果からなる条件により、デンプレートとテンプレートのエンテンツ処理 を行うことかできます。
- <u>グループ</u>: XPath 条件式により選択された要素のグループに対して処理を定義することができます。
- 並べ替え: XML 要素のセルを複数のノートキーによし述べ替えることができます。
- <u>ノラメーターと変数</u>: グロー)
 がは SPS レベルでデフォルト値を持った。ラ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、コマン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トにて、かプキュメト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とい作成されます。参照を使用するその他の機能には、(i) <u>テキスト参照</u>(ドキュメト内にあるテキストをマークして、ドキュメ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レク</u>(ブックマークのキーにより、ハイパーレンの参照先を作成する)かあります。パイパーレク にお外部のリ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いても、このセクション以下で説明されます。
- 三 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) オプションとして、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。.
- デザイン内部の任意の場所でノードのエレテンンを表示する

- 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ヘ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)。

'ath 2 条件式: run t(newsitem)				
様条 評価 スキーマ属性や要素の選択 選択: \$XML/presswatch/newsitems 選択: \$XML/presswatch/newsitems 回 ○ A前空間 日 ○ \$XML (3-1>) 日 ○ ♪XML (3-1>) 日 ○ ♪ XML (3-1>) 日 ○ ♪ x	演算子/式の選択 田軸 田比較 田条件 団Quantified/for 田論理 田ノード 田数値 田シーケンス 田 Type		関数の選択 田 Accessors 田 Boolean 田 コンストラクター 田 コンテキスト 田 変換 田 日付と時間 田 日 田 この他 田 ノード 田 数値 田 QName 田 シーケンス 田 文字列 田 URI	
■ 絶対 XPath (A)	4	F.	4	Þ.

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 式を編集セクションを参照くたさい。

「OK」ボタンをクリックすることで、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が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リックしてくたさい(以下のスクレーンショントを参照)。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(上のスクレーンショント) が表示されます(上のスクレーンショントを参照)。



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、通常のテ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・ティウィドウの自動計算グループにある入力フォーマトプロ・ティでは、数値や日時と、ソナデータ型を自動計算の入力フォーマト 型を自動計算の入力フォーマト とて指定することができます。

XPath 条件式には改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、新たな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 リペコピー、貼付けや移動を行うこともできます。新たな場所のコン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。

重要なポイント

以下の点に注意してくたさい

- 自動計算はデザイイ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

- <u>挿入 | 自動計算</u>
- <u>XPath ダイアログ</u>
- <u>XSLT エンジの情報</u>
- サンプル注文書

7.1.2 サンプル:注文書

<u>(マイ)ドキュダトフォルダー</u>以下のC:\Documents and Settings\<username>\My

Documents\Altova\StyleVision2021\StyleVisionExamples\Tutorial**Auto-Calculations**\ フォルダーにある Invoice.sps では 以下にあるような自動計算の使用方法を確認することができます:

- 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(以下のスクリージェョナを参照)。

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序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(以下のスクレーンショナを参照)。

Product <mark>1</mark> :	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() 関数によば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 とう XPath 条件式が使用されま す。後者の方法は、position() 関数を使用するのか難ししょうな(おオオネマ可能な) 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、サンプル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、ユーザーがコンボドックスの値を選択することで、製品のカテゴリー(Book、CD、DVD、お当は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なりま 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ドキュメントに ルッグ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ン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」ことはます。製品のCategory コンボボックスにて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てよります:

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 ノードか更新されます。従って、VAT ノードのコンテンツ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:

for \$i in Category return PriceNet + (PriceNet *(/Invoice/Categories/Category[. =
 \$i]/@rate) div 100)

XPath 条件式は、
<u>プロ・ティウィンドウから確認並びに編集</u>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
か動作していることを確認することができます。
製品の、VATを含む)販売価格か変わることに注目してくたさい。

Product <mark>6</mark> :	A Short History of the American Century
Net price:	€ 20.00
Category:	DVD 💌
VAT:	<mark>15</mark> %
Price including VAT:	€ <mark>23</mark>

- 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・シリテンプレートの両方で、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。条件とはコーン以上の分岐 から構成される SPS コンポーネトのことで、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、コンテキストノード上にある Location 属性の値が "US" であるかか チェックされます。 2番目の条件分岐にある XPath 条件式では、Location 属性の値が "EU" であるかか チェックされます。各分岐には、テンプレート(条件テ ンプレート)か 含まれています。条件を伴うノード か 処理され、結果か 真となる最初の条件(条件テンプレート)か 処理されます。条件内にある 処理か 終了すると、その他の分岐にある条件に ま甲価されません。この機能を使用することで、ノードの値によい異なるテンプレートを使用するこ とか可能に ゴンドす。ここで紹介した例では、US とEU という値によい異なるテンプレートを使用することができます。

このセクションコよ以下のトピックか含まれています

- 条件のセルアップ 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

以下の操作にお条件をセオ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(以下のスクノーシンョナを参照)。

関数の選択	≣ .
 田 Accessors 田 Boolean 田 コンストラクター 田 コンテキスト 田 変換 田 日付と時間 田 ID 田 その他 田 ノード 田 数値 田 QName 田 シーケンス 田 文ーケンス 田 文字列 田 UFI 	
	田 UNAame 田 シーケンス 田 文字列 田 URI

- 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)。

XPath の編集⊗…	
新規条件分岐の追加(A)	
条件分岐のコピー(0)	
条件分岐の削除(D)	
全ての条件分岐を表示(Y)	
条件分岐を上へ移動(山)	
条件分岐を下へ移動(₩)	
	×
タグで囲む(W)	

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。 <u>XPath 式 の編集ダイアログ</u>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、<u>XPath 式の編集を行つづ</u>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(以下のスクノーンショットを参照)。

2条件式:				<u>र</u> स
構築 評価 →マ属性や要素の選択	演算子/式の選択		関数の選択	Ē
************************************	n/me 田 田 田 田 泉 件 田 Quantified/for 田 油理 田 ノード 田 数値 田 シーケンス 田 Type	*	 Accessors Boolean □ンストラクター □ンテキスト □変換 □日付と時間 □D □その他 □ノード □数値 □QName □シーケンス ①文字列 □URI 	
絶対 XPath (A)	4	•	4	- F

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やめ、条件内にある分岐の順序が重要な意味を持つ ことったります。分岐の位置を変更するコよ移動する分岐を選択し、右クルクにと、展示されるコンテキストメニューが条件分岐を上 へ移動おは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、右クハクで表示されるエンテキストメニューから条件分岐の削除を選択します。

- □ 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

条件の編集 7.2.2

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:

- (条件ではなく)条件分岐を選択します。
 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)。

プロパティ			×
la •13 •5	PRTH		_
プロパティ:	属性	[値	_
段落	🔄 when		
条件分岐	XPath	not(n1:Address or _n1:Address_EU)	
			~

3. グループの XPath プロパティにある編集ボタン 🛄 をクリックします。 <u>XPath 式 の編集ダイアログ</u>が表示され、選択 した条件分岐のXPath 条件式を編集することができるよう なります。

-	関連項目
_	因是沒口

- 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、しくつか留意するへきことかあります。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

- 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。
- 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
- <u>自動計算</u>

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こなります。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 属性かある場合、プ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、各ブランド内にあるモデルや年式ごとこグループ化を行うことができます。

グループ化の条件

アイテムは(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、ます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よ、ノー ドますす現在のグループ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(以下のスクレーンショントを参照)。

出力グループを定義
HTML、RTF、そして PDF XSLT2 出力のグループ化を定義します。グ ループ化は Authentic には適用されません。
入力されたマッチは単一の要素を評価しなければなりません。複数の要素 を評価するマッチは処理エラーを発生させる原因になります。
▶ グループ化を有効
グループ化の種類
● group-by
C group-adjacent C group-ending-with
Øgrade
XPathを編集(<u>E</u>)
OK キャンセル

ダイアロゲでは、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ポシンから選択します。 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 ならい)こgroup-adjacent の場合)することができるまか、マッチ を行うための、ダーンを入力(group-starting-with ならし)こgroup-ending-with の場合)することができます。「OK」をクルックすると グループセルをアルファベナ順(昇順)で並べ替えるか選択するよう促されます。グループセルの並べ替えは後から変更することができるま か、並べ替えを削除することもできます。以下のスクリーンショントでは、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。

§div 0	employees	- ノードのグループ化
§div >=(自動計算)〈 [§] div	
	Grade	Name
हा हा ।		
8 .08	<mark>≡grade</mark>)(コンテンツ) <mark>≡grade</mark>	= first →(コンテンツ) < = first = last →(コンテンツ) < = last
C opersor	■grade (コンテンツ) ■grade 1 (Oemployees) [§] div 現在のク	<pre>= first >(コンテンツ) <= first = last >(コンテンツ) <= last //レープテンプレートの</pre>

上のスクレーシンコナトでは、person ノードカヴループ化され、作成されるグループはアルファベナト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べ替えを行うか尋ねられます。グループの並べ替えしは、つても行うことができるまか、ここで作成した並べ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、封け捎除を行うコよ目的のグループ化テンプレートを右クソクして並べ替えを選択します。この操作により<u>出力の並べ替えの順序を定義ダイアログ</u>が表示されます。このダイアログの使用方法については、<u>並べ替え</u>のセク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コよ、並べ替えキー「こcurrent-grouping-key() XPath 関数を使用しなければよりません。詳細 については、このセクション以下にあるサブセク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、おけは編集するには、テンプレートを右クルクして、グループ化ませば、べ替えをそれそれ選択します。該当するダイアログが表示されるため、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

<u>ユーザー定義テンプレート</u>により、指定したXPath条件式により選択されるアイテムコネル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。

三 関連項目

- 出力の構造
- <u>並べ替え</u>
- ユーザー定義テンプレート
- <u>サンプル: Group-By (Persons.sps)</u>
- <u>サノプル: Group-By (Scores.sps)</u>

7.3.1 サンプル: group-by(Persons.sps)

Persons.sps サンプルはPersons.xsd スキーでをベースご作成されており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\「本納されてい 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な 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お表示され、部門はアルファベナ順にて並べ 替えられます。各部門テーブル内部では、person にお表される従業員が与えられた権限(grade)に応じてグループ化が行われ、各 grade では従業員(person)が苗字(last)によ述べ替えられて表示されます。

グループの作成の手順

グループを作成するよめの手順は以下のようよります。まずはgrouping-keyをdepartment属性に対して使用することでperson要素に 対してグループを作成します。この操作によりdepartment属性の値によるperson要素のグループ化が行われます。並べ替えが指定された 場合、departmentによるグループをアルファベナ順に並べ替えることができます(例えばAdministration か最初に表示され、次に Board of Directors か表示されます)。部門ごとは独立したテーブルを作成するため、(department grouping-key をベースした) 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 属性にお更にグループ化され るらに指定します。

SPS の作成

以下の操作によげ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。

Operson © Pt & current-group() (コンテンツ) 🌲 Operson

- 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)。
- 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() テンプレートが新たに作成されます(以下の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は、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グループ化ま だは並べ替えを選択します。この操作により応するダイアログが表示され、設定を確認ませま修正することができます。

関連項目

- <u>出力の構造</u>
- <u>並べ替え</u>
- <u>サノプル: Group-By (Scores.sps)</u>

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人 Scores.xml を作業用 XML ファイルとて使用します。ファイルは (マイ)ドキュメトフルダー 以下にあるC: \Documents and Settings \<username>\My Documents \Altova \Style Vision 2021 \Style Vision Examples \Tutorial \Grouping \Scores \に格納されてま す。ドキュメト 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 要素には、下に示されるような構造を持ちます。

デザイノは3つの、ーンから構成されます(以下のスクレーンショナを参照)。(i) //match/@date によりグループ化された日付ごとの試合結果、(ii) //match/@group によりグループ化されたリーググ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
ltaly - Holland	2 - 2

2007-10-13

Argentina - France	2-0
England - Spain	0 - 0

Match Results: By Group

Group A

2 - 1
2-2
1 - 2
2 - 2
1-0
1 - 1

Group Tables

Group A

Team	Ρ	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、日付とトーナメトグループにお試合の結果をグループ化します。データの配置を行うため、日付とトーナメトグループにお作成される各グループのメンバーイズさして、境界線が表示されないテーブルを作成します。試合の結果は日付とトーナメトグループごとご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

以下の操作によげ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()タグを右クリックして、「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。
 - 三 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、同じ修飾名を持すこ兄弟要素ノードのセオに対して、1つ以上のノートキーによけうことができます。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 により並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(要素おけば属性) どよります。ここで紹介した例では、LastName がノートキーとなります。

並べ替えの対象となる2つの要素に含まれているソートキーの値が等しい場合、更にノートキーを追加することで、お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なるLastNmae に加え、2番目のノートキーとなる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。

並べ替えのセナーに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

<u>ユーザー定義テノプレート</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に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テノプレートに対して適用されます。

このセクション

- <u>並へ替えのメカニズム</u>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
- <u>XPath ダイアログ</u>
- 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

要素のノードレマオレて並べ替えを行うしては、2つのステップを踏む必要があります:

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。

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)。

出力ソートの順序を定義
HTML, RTF, PDF そして Word 2007+出力の並べ替えの順序を定 義します。並べ替えの順序は Authentic には適用されません。
入力された各マッチは、単一の要素を評価しなければなりません。複数の要 素を評価するマッチは処理エラーを発生させる原因になります。
ソートノード: newsitem
LastName テキスト 💌 昇順 💌
OK ++>24214

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。

並べ替えか使用される例にていては、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べ替え、

ユーザー定義テンプレート

<u>ユーザー定義テノプレート</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に対してテンプレートを適用することができます。 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。

ソートキー、関するとモ

入力されたXPath 条件式により選択されるソートキーは単一のノードでなければならず、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、文字列ませば数値の値に対して解決可能である必要があります。

XSLT 2.0 または 3.0 のSPS にてノードのシーケンスがノ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り。

XSLT 1.0 では、ソートキーのセンターによりノー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おく、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ない場合もみます。例えば、上に学げた…/Person/LastName XPath 条件式によりエラーが返されることはかませんが、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みません。この場合、location/@* と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になりますが、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

- サノプル 複数のノートキーによる並べ替え
- ユーザー定義テンプレート
- 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、チームメンバーが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\StyleVision2021\StyleVisionExamples\Tutorial\Sorting\SortingOnTwoTextKey s.sps にて確認することができます)。テーブルの行け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出力のスクレーシンコントを参照)。

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

デザインビューーこて member 要素(以下のスクノーシンコット)こで青色で イライトされているノード)を右クトックし、コンテキスト メニューカら並べ替え コマイ を選択します。



この操作により、出カソートの順序を定義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)。並べ替えに選択された要素の member がノートノードエナトリーに表示されます。このノードがノートキーを選択する XPath 条件式のコンテキストノードにもなります。ペイン ツールバーの左側にある行の追加ボタンをクリックして、最初の並べ替え処理を追加します。追加された行にて XPath 条件式をマッチカラム

に入力し、last ノードを選択します。構築ポタン・・・・・ をクリックすることでも、XPath 条件式を構築することができます。データ型カラムでは、 ソートキーのエンテンンを評価する方法を、テキストと数値から選択することができます。順序カラムでは、並べ替えが行われる方向を、昇順と 降順から選択することができます。テキストと昇順を選択し、「OK」をクリックしてくたさい。

出力ソートの順序を定義
HTML, RTF, PDF そして Word 2007+出力の並べ替えの順序を定 義します。並べ替えの順序は Authentic には適用されません。
入力された各マッチは、単一の要素を評価しなければなりません。複数の要素を評価するマッチは処理エラーを発生させる原因になります。
ソートノード: member
見目 × データ型 順序
last <u>…</u> テキスト 🔽 昇順 🔽
OK キャンセル

デザインビューでは、並べ替えのフィルターを表すアイエンがmember タグに表示されます member 回う。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 目してくたさい (Nadia が John の前に表示されており、これは XML ドキュメント における要素の順序に従ったものです)。名前による並べ 替えを行うすっかに、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

デザインビューインある member タグを右クリンクして、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ス、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ントを参照)。「OK」をクリングして完了します。

出力ソートの順序を定義
HTML、RTF、PDF そして Word 2007+出力の並べ替えの)頂序を定 義します。並べ替えの)頂序は Authentic には適用されません。
入力された各マッチは、単一の要素を評価しなければなりません。複数の要 素を評価するマッチは処理エラーを発生させる原因になります。
Ункунк; member
1ast
first
OK ++>ンセル

HTML 出力では、最初に苗字が、次に名前によし述へ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

- <u>動的テーブルの作成</u>
- XPath ダイアログ
- <u>ユーザー定義アンプレート</u>

7.5 パラメーターと変数

ハラメーターと変数はSPS内部にて宣言ならびに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、ハラメーターは、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。

- <u>ユーザー宣言ハラメーター</u>: ユーザー定義のパラメーターをSPS にて使用する方法について説明します。
- デザインアラグメトレマオする、ラメーターデザインフラグメトとともし、ウメーターを使用する方法について記述します。
- ソースに対する SPS / ラメーター: 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 か自動的に 定義する特殊な種類の、ラメーターについて説明します。これら、ラメーターの名前と値はユーザーにとって既知のたのであるため、 SPS 内部で/ ラ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。
- <u>変数</u>:(i)特定のスコープに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。
- 三 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ト

7.5.1 ユーザー宣言パラメーター

SPS では、ユーザーには宣言されたッテメーターがデフォルの文字列値とともにグロー・ジルに使用されます。一旦、テメーターが宣言されると、SPS 内の任意の場所から XPath 条件式には使用することができます。パテメーターのデフォルト値は、<u>Style Vision Server</u> から XSLT スタイパシートに対して新たなグロー・ジル値を渡すことで、個々の XSLT 変換に対してオー・メ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パテメーターは、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て使用する この場合、目的の値をパラメーター値とて保存して、 必要とされている場所や計算にてパ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度す: 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度すことができます。

メカニズムの使用方法

ユーザーによ「宣言された」デメーターを SPS にて使用するコム以下にある2つのステップを踏む必要がみます。

- 1. <u>必要ないテメーターを宣言する</u>。
- 2. <u>宣言されたい デメーターを参照する</u>。

パラメータの宣言

ユーザーによる/ ラメーターの宣言と編集は/ ラメーターの編集ダイアログにて行うことができます(以下のスクノーンショナを参照)。/ ラ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/ ラメーター 」コマンドからアクセスすることができます。

パラメーターの編集	
パラメーター名は先頭の '\$'なし	で入力してください。
見冒	×
パラメーター名	デフォルト値
state1	CA
state2	NY
	OK ++>>tz//

・プメーターの名前とデフォールの値により、プメーターの宣言を行します。デフォールの値を与えない場合、デフォール値を空してます。

以下の操作により、ラ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/ ラメーターの追加や、既存/ ラメーターの編集は、SPSの編集時にいつでも行うことができます。

×ŧ

- 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イン概要サイドバーでも宣言することができます。

宣言されたパラメーターの参照

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'))。

宣言されていない シテ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 シテメーターを宣言してもエラーとはおはません。

デザインフラグメントに対するパラメーター 7.5.2

デザインフラグメートの、ウメーターにより、作成したデザインフラグメントに対して、ウメーターを定義し、この、ウメーター(ネルてデフォルト値を 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、ラメーター値を入力することも可能で、各デザインフラグ 火トの出力を個別に修正することもできます。

例えばEMailAddresses とら名前のデザインフラグメートをaltova.com とらデンオルト値を持つDomain とら名前の デメーターとともに 作成する例を考えてみましょう。この、デ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
のアドレスマオレてはEmailAddresses デザインフラグメントにある Domain ノラメーターの値に altova.eu を使用します。 同様の方法で、日本のアドレスマオレては altova.jp を Domain ノ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マオしては Domain の値を変更すること無く、デフォリト値の altova.com を使用します。

デザインフラグメントに対してい デメーター値を使用するには、以下にある2つのステップを踏むことにないます。

- 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パラメーターを定義します。
- 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て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 ラ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・トーエある ラメーターを定義リ ンクをクリックすることで、ラメーターを定義できます(以下のスクリーンショントを参照)。

'EmailPerson'	<u>パラメーターを定義</u>
<mark>On1:Person</mark> On1:First (コンテンツ) On1:First =(自動計算) On1:Person	<mark>On1:Last</mark> (コンテンツ) On1:Last

この操作により、デザインフラグメトの、ウメーターを定義ダイアログが表示されます(以下のスクレーシショントを参照)。左上にある追加なら びに挿入アイエンをクリックすることで、パウメーターを入力する行が追加されます。パウメーターの名前、データ型、出現の回数、そしてデフォル ト値を入力ませく選択します。パウメーターの出現属性により、パウ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。出現属性におプションとなっており、デフォルトで「無しませまけつ」にセントされています。パウメ ーターの数に制限はありません。

デ	ザインフラグメン	ノトのパラメー	ターを定義		
	見冒				\times
	名前	型	出現	デフォルト値 (XPath)	
	Domain	xsistring	💌 無しまたは1つ	💌 'altova.com'	··· 🔽
				OK 4	和心也儿

ダイアログロシス種類の削除アイコンの数はす。各ハ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、ハラメーターのデン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ハ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

デザインフラグメートを作成すると、デザイン内にある複数の箇所へ、デザインソレーやスキーマソレーからドラッグすることにより挿入することができます。デザインフラグメートのEmailPersonがn1:Name要素の後に挿入された状態を以下のスクレーシンヨントに示します。

§div 0n1:Office
<u> 0n1:Name</u> (コンテンツ) <u>0n1:Name</u>
On1:Department
0n1:Name (コンテンツ) 0n1:Name
💼 'EmailPerson'
On1:Department ? 1

デザインプラグメトにてい ラメーターが定義すると、デザインプ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。 デザインプラグメントの ・ラメーターを編集ダイアログが 表示されます(以下のスクノーシンコットを参照)。

デザインフラグ:	メントのパラメーターを	編集	
名前 Domain	型 xs:string	値 (XPath) altova.eu	
. I		OK	++>2U

このダイアログでは、ラメーターの値を編集することができます。「OK」をクリックすることで操作を完了します。入力された、ラメーター値がデザ インフラグメントのインスタンスにて使用されます。パラメーターの値が編集されていない場合、オリジナルの、デフォルトのパラメーター値が使用 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

• <u>ユーザ 宣言/ラメーター</u>: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、ませは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 キーマイニオすると ラメーターが Style Vision により自動的に宣言され、そのスキーマイニオして割り当てられた作業用 XML ファイルの URI が、パ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、Style Vision 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 ラメーターにセナ されます。

ソースマ対するパラメーターを参照する

各 SPS パラメーターは、そのスキーマーズ対応する 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、各スキーマーズ対応 した XML ファイルは、作業用 XML ファイルませは DB から生成された XML ファイルどなります。 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:

- 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。
- 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。
- 三 関連項目
- <u>ユーザ 宣言/ ゚ラメーター</u>

7.5.4 変数

変数は (i) 変数の宣言 と(ii) 変数の使用によ構成されます。

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ゆるテンプレートにて宣言することができます。変数はコお名前、データ型、そして値か与えられます。更 に、Enterprise エディンコンのAuthentic View にて編集可能かを指定することができます。変数はエ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。ドキュメント全体をスコープとする変数を宣言すること、ル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ようの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。

変数の定義						
ここで定義された変数は現在のテン	グレートのスコープ内で	使われている任	達の XPath 条件式がら参照する	ことができます。		
変数が編集可能とマーク(<auto> そうするには、コンテンツ要素または:</auto>	変数が編集可能とマーク(<auto> の代わりに型を選択する必要があります)されている場合、その値は Authentic にて変更することができます。 そうするには、コンテンツ要素または書式コントロールを持つ変数テンプレートにて変数を参照します。</auto>					
ここで入力された値は変数の初期(首になります;実際の値	は変更されるこ	ともあります。			
・"計算" が "一度" にセットされて	いる変数の実際の値は	t, Authentic (て手動で編集された時のみ変更な	5れます。		
- '計算''が '自動''にセットされて 数)の変更後)。	いる変数の実際の値は	北要に応じて自	目動的に更新されます (例: 変数が	依存するアイテム (XML	ノードが他の変	
- これら Authentic ピューでの値の	変更は出力のピューに	影響を与えず、	Authenticピューが閉じられる際に	破棄されます。		
□ 変数						
					~	
		开山	VD-4-名化士			
SelectGroup ▼)≌ vs:string			- 면 ▼	
Groups		<auto></auto>	💌 distinct-values(results/	/group/@id)	自動 💌	
				ок	キャンセル	

- 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 エディ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。 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。変数を修正するこは、変数が作成された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コントロールされます。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/Name とら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。

1	
2	
評価	
	2 1 22 評価

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(上のスクレ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まレイパトーレンカら、これらのアンカーに対して参照を行うというかちで作成されます。

まずはアンカー(ませはブックマーク)のメカニズムについて簡単に説明し、目次メカニズム全般について説明します。ブック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こなります。

ブックマークのメカニズム

ブックマークトコよ、単純ブックマークと複合ブックマークという2種類のメカニズムが存在します。目次の作成にコよ複合ブックマークが使用されます。

- 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。ハイパーレクオボキ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ご留意してくたさい。
- 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。

 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

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、参照元のエレ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 デジョンコンズSLT 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) -正 確には日次のテンプレート - 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。

Stylevision User Manual (excerpt from v200
Note: This is an excerpt from an outdated version of the StyleVis. Manual. It is used here solely to demonstrate the use of various StyleVis features. The XML document is structured into sections and sub-sectior down three levels. It contains headlines, paragraphs, lists, tables, images, does not include all the content, presentation, and usability features of the user manual and should not be used as a substitute for the latter.
Sdiv Table of Contents: Chapters and Their Sections
^{~4年} [§] div [~] ↓'toc' (num-M): (テキスト参照)()(ページ参照) [~] ↓'toc' [§] div
[§] div [~] Ψ'toc' (num-M): (テキスト参照)()(ページ参照) [~] Ψ'toc' [§] div [~] Ψ'toc' (num-M): (テキスト参照)()(ページ参照) [~] Ψ'toc' [§] div
Ohelpproject Otopic 4- Ohelpproject Otopic 4- Ohelpproject Umbedder Umbedder
Sdiv (コンテンツ) Sdiv Obserder
(その他のコンテンツ) Obody Otopic Otopics Ohelpproject

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) 目次は階層構造おはステル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カムらびに章レベルの目次に加え、図表のノストをフラルな構造で作成することができるようこないます。

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ク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。
- <u>目次レベル内部のドキュメントの構造</u>
 日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 一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い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ル</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。詳細については番号の自動割り当てのセクションを参照くたさい。

用語解説

イク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ン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て示されるように、 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あります。

- 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て定義され た目次レベルの構造に対応します。対応するレベルにある目 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:名前を与えることこより、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クマークへの参照を行います。

三 関連項目

- 目次に含まれるブックマークアイテム
- 目次テンプレートの作成
- · ブッケマークとハイパートンク

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はは、2つのステップを踏む必要がみます(ステップの順序に意味はありません)。

1. <u>デザイズキュメ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>: デザイインおける目次レベルは テンプレート上に おけ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。以下のスクレージンコントでは topic テンプレート ①topic いー にて目次レベルが作成されています。

Ohelpproject 4-目次レベル Otopics Otopic 4- Obody Oheader Opara
^{§div} ∕(コンテンツ) ^{<§div}
Opara Oheader (その他のコンテンツ)

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:

「日次レベル」
「日次レベル」
「「日次レベル」により日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、レベル内部のデザイン構造 セクションにて説明されます。日次テンプレーの作成された場合、デザインにて作成されたレベルに対応するかたちで目次テンプレートを作成する必要があります。フラットな構造(レベル1つ)の目次においても、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. <u>目次ブックマークの作成</u>:名前なられニ目次アイテムデキストを伴うデザインニて目次ブックマークを作成します。目次ブックマークは デザインコンポーネントを囲むかすちでデザインすることもでき、コ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 下のスクレーンショントでは、デザインコンポーネントを囲んでしない、目次ブックマークが示されています。

Ohelpproject Otopics Otopic - Obody Oheader VEN'toc' Opara
^{§div} (コンテンツ) ^{(§div}
Opara Oheader
(その他のコンテンツ) <mark> Obody Otopic Otopics Ohelpproject</mark>

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、<u>目次ブックマークの作成</u>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コ作成されオンベルは違う)レベル参照とう名前のネストされオンベル内にTOC テンプレート されます。各レベル参照 にある目次参照からよ(目次ブッケマークの名前による)目次ブッケマークへの参照が行われます。XMLドキュメト内部の対応するレベル にある指定された名前を持った目次ブッケマークが、目次内のアイテムとて作成されます。例えば したたいたいで、 れる目次参照により、XMLドキュメト内の対応するレベルにある chapters とう名前の全目次ブッケマークが参照されること」これます(目 次参照のスコープが 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デキスト属性が目次アイテムのデキストとして 使用されます。



上のスクレーシショナに示される目次テンプレートでは、3つのベル参照がなみされており、各レベル内にある目次参照にコまの、ベルの目 次アイテムご対するテンプレートが含まれています。例えば、最初のレベル参照にコまмутос 予いまたのに、ためにより、最初のレベル内にあるMytoc とう名前を持って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なります。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ります。ドキュメート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。

このセクタン以下にあるサブセクションでは(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クションで説明されています。

- 三 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

7.6.1.1 目次レベル内部のデザイン構造

目次をデザインするための階層構造は、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 構造に関連しますが、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となります。フラナな構造(レベリが1つだけの)目次の場合、デザインドキュメトにつかなくとも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つ以上のレベリの存在する場合、これらレベリのどれか、まけよ(1つのレベリとして複数のレベリを統 合することにより、複数のレベリ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・シリテンプレート、またはこれらか合わさったり、ロインいめ作成されます。レベンはどこに作成されるかに関わらず、これらのレベンリ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おけまこパーネントの周りに作成されます。以下にあるスクレーシ ショナでは、topic テンプレート上に(① topic いーン により示される) テンプレートが作成され、topics 要素の周り、注別のレベルが作成され て、ます(いー目次レベル)。これら2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 いー目次レベル いーマ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。

Ohelpproject 4-目次レベル Otopics Otopic 4- Obody Oheader Opara
[§] div (コンテンツ) ^{(§} div
 Opara Oheader (その他のコンテンツ) Obody Otopics 毎日次レベル Ohelpproject

以下の操作によりべいを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」」または「シを挿入」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・シレテンプレート内部のレベル

グローッシリテンプレート内部にもレベリをセオすることができます。この場合、グローッシリテンプレート内部に作成されたレベリとメインテンプレー ト内に作成されセーベルの両方により、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つのレベリが、そしてグロー・グリテンプレートのtopic 内部に別のレベリは定義されています。

Ohelpproject Otopics Otopic 4= Oheader Opara
^{§div} ∕(num-lvl): (コンテンツ) ^{(§div}
<mark>Opara</mark> Oheader (その他のコンテンツ) Obody Otopic Otopics Ohelpproject
メインテンプレートの終端 〈🍱\$XML
topic (複数)
? 1 2 3 Otopic 4- Obody Oheader WyTOC' Opara
^{§div} ∕(num-lvl): (コンテンツ) ^{§div}
<mark>● para</mark> ↓ ^{IIII} 'MyTOC' ● header (その他のコンテンツ) ● Dbody ? 1

上のスクノーシショナに示されるコンテンソモデルでは、topic か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。(SXML タグによ終了地点が示される)メインテンプレートでは、最初のレベルのtopic にてレベルがセナされています(● topic い)。メインテンプレートにあるその他のコンテンソニより、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・シレテンプレートも処理されること」なります。

topic 「マオするグロー・シーテンプレートでは、topic テンプレート「マオレて(「Ctopic G-ルは2番目の、ベルのtopic 要素にて出現し、目次では、最初の、ベル内部にネストされる2番目の、ベルトマオ応します。このグロー・シーテン プレートにはその他のコンテンパと含まれており、topic 「マオするグロー・シーテンプレートも再帰的な topic 要素につけて適用されること」なり、 目次階層内にネストされたシーベル(3番目の、ベル、4番目の、ベル、など)が新たに作成されること」なります。

デザインを行う際には、デザイン内に作成されたレ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、フォーマトすることになります。

フラナ構造の目次におけるレベル

フラオ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:ドキュメイ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。

- デザイボキュメナトに含める目次レベルを1つ付けにする: 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つ付けなります。
- デザイズキュメト内の目次レベル物複数存在する場合: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メベル参照を配置することしています。ターゲナム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つ配置することしています。
- デザイボキュメトに複数の目次レベルが存在する場合:目次テンプレート内に配置された目次参照には、ターゲオとなるドキュ 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ります。

ドキュメト内にある全てのイメージのノストを、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オ少なくとも 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ク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)ターゲナン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作成します。目次参照には、ターゲナンとなる目次レベルにおける目次ブックマークの名前か含まれます。

以下のスクレージンコナにある目次テンプレートでは images とう名前の目次ブックマークを参照する目次参照かきまれているレベル参照 か示されます。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(つまドキュメント全体)にある image とう名前を持って全目次ブックマークか参照されること」ことはます。



デザイン内に複数のレベルは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番目のレベルはさけ目次参照が存在する2つのレベル参照から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レティにより、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。

- 三 関連項目
- <u>目次(TOC)</u>
- 目次に含まれるブックマーケアイテム
- <u>
 日次ブックマークの作成
 </u>
- <u>日次テンプレートの作成</u>

<u>目次参照のスコーププロ・ディ</u>

7.6.1.2 目次ブックマークの作成

目次ブックマークオジキュメトデザインの目次レベル内部に作成されます。目次ブックマークオンインテンプレートならびにグロー・シレテ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つの目的のナックコンガンを用されます:

- 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:

- <u>目次ブックマークの作成 ウィザード</u>を使用する: 目次ブックマークの名前、そのテキストエントリー、自動的に番号か割り当てられる か、表示されるレベリを指定します。
- 空の<u>目次ブックマークを挿入</u>する:ブックマークのプロパティを、ブックマークの作成後個別に指定すること」ないます。

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。
- 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 | 目次を挿入 | 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)。

目次ブックマークの作成	×
○ 目次ブックマーク要素の子を使用して目次ブックマークテキストを決定① ◎ 以下の XPath 条件式を使用して目次ブックマークテキストを決定♡	
para/@title	
、 XPath を編集(<u>E</u>)	
目次ブックマーク名:(相互参照はこの名前を使ってこの目次ブックマークを参照することができます)	
chapters 🔽	
▼ この目)次ブックマークを自動番号割り当て(A)	
< 戻る(日) 次へ(11)> キャンセル	

- 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。

目次ブックマークのレベルを定義	×
目次ブックマークがドキュメントレベルで定義されている場合、相互参照はそのレベル内の目次ブックマークのマッチ のみ使用されます。以下には全ての親テンプレートとドキュメントレベルが表示されます。新規ドキュメントレベルとし てマークされるテンプレートを選択してください!	,
☐ XML	
□ 目次ブックマークの定義を後で行う(□)	
く戻る(B) 完了 キャンセル	

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「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 編集コマイを使用してください。

コンポーネントしつ対して階層的ませば連続下番号を挿入する

目次ブックマークには出カドキュメントの(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。

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リックして、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。プロノティウムドウに表示されているプロノティを直接編集するか、目次ブックマークを右クリックして、編集するプロノティを選択 し、これらのプロ、ティを編集することができます。

プロパティ			×
님 내 내	PATH ×		_
プロパティ:	属性	値	
目次ブックマ	⊖ 目次ブックマーク		
	id		
	グループ	toc	
	元となるテキスト	XPath	···
	参照されていなければ削除	no	•

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>
- 目次に含まれるブックマーケアイテム
- 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さ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 デザイノ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ためのステップを以下に記します。

- 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。
 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。

268 高度な機能

目次ページの作成	
目次ページプロパティー	
生成された目次ページの名前(N)	: toc
生成する目次レベルの階層(上):	3 ÷
▼ ページ参照を含める(円)	
🔽 ハイパーリンク /ブックマークを移	軍効にする(<u>日</u>)
	ОК + +>ン12л

- 3. ダイアログニで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ントリーは、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、必要な時にはしいでも編集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ショ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、おけば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・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、イパーリンクとして作成されるかを指定することができます。
- <u>日次参照</u>を目次テンプレートのレベル参照へ追加、ませ、胡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>Style Vision メカニズム</u>によりCSS プロ・テを介したフォーマトを行うこと 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。

三 関連項目

- 目次に含まれるブックマークアイテム
- 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ナ

7.6.2.1 目次テンプレートにおけるレベル参照

<u>目次テンプレート</u>はレベル参照にて構築されます(以下のスクレーシンヨルを参照)。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、<u>目次ページの作成ダイアログ</u>にて指定された数のレベル参照が作成されます。

Sdiv 目次 - 章と節 Sdiv
²⁴⁻) _{\$div} <u>へいm-lyl)</u> (テキスト参昭)()(ページ参昭) <u>へいm-lyl</u>)(_{\$div})
<mark>■ Interactional Interaction Interactio Interaction Interaction Interaction Interaction</mark>

レベル参照がネトされているとうごと注目してくたさい。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1対1で 対応することはよります。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番目のレベル参照は、SPS デザイン内にある2番目のレベルに対応することはなります。目次テンプレートのレベル参照 により、SPS 内で<u>指定されたスコープ</u>のにある<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に対応したドキュメントレベル の目次ブックマーグを参照のターゲートとすることができるまか、全ドキュメントレベルはある目次ブックマーグをターゲットにし、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グをターゲットとすることができます。

レベル参照の挿入省県除

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。

コンテンソの周リ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ツを選択し、コンテキストメニューおけメ ニューオプションからタグで囲む | 目次レベル参照」を選択します。メニューオプションおけココンテキストメニューから選択することのでき る 挿入 | 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ゴ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、ベル参照だけで、内部にあるコンテンソ、排削除されないということに注意してください。

三 関連項目

- <u>目次テンプレートの作成</u>
- 目次に含まれるブックマーケアイテム

• <u>日次参照:名前、スコープ、ハイパーレク</u>

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、守か与えられています(以下のスクレージョントを参照):

- 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カーノルを配置して、メニューオプションコある「挿入」 おうコ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ックして編集するプロ、ティ(ハイパーリンクの 作成、ID を編集、グループを編集、おコよスコープの編集)を選択します。この操作によりプロ、ティウィンドウが表示され、選択 されたプロ、ティの値を指定することができます(以下のスクノーンショナを参照)。

∑לם לבי	<u>د</u>
日日次参照	
id グループ	toc
	現在のレベル ・
ハイバーリンクを作成	現在のレベル
	現在と下位のレベル
	Y
	UK 47020

目次参照タグが選択されている状態で、プロレディウインドウニある目次参照プロレディグループから直接プロレデ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クアイテム
- <u>日次テンプレートの作成</u>

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種類の標準コンポーネトに加え、オプションのユーザー指定コンテンンを加えることができます。以下に4種類のコンポ ーネントを示します(以下のスクレーシンョントを参照):

- (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-seq)により示される階層的または連続した番号

Image: Second State (Second State St

目次テンプレーか作成されると、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ションが選択された場合、リーダーとページ参照コンポーネトも追加されます。目次アイテムからコンポーネトの挿入や削除を行うこともできます。コンポーネトを 挿入するは、目次アイテム内にある目的の場所にカーンルを配置して、右クリックにお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」おけば目次を挿入 | 階層的な番号付け / 連続的な番 号付け」を選択します。ネストされたノベルによりデザインが構成されている場合には階層的な番号を使用し、階層構造が存在しない場合 (つまり1つのプラナな目次レベルはけが存在する場合)には連続的な番号を使用します。プラナな目次に関する下記の記述を参照くたさい。 い、コンポーネトを削除するはよ、そのコンポーネントが選択された状態でDelete キーを押下してくたさい。

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

フラオ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:ドキュメイ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。

- デザイボキュメナ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つだけにないます。
- デザイボキュメト内の目次レベル
 物複数存在する場合:目次テンプレート内に、参照される目次レベルの
 位置に対応したレベル
 参照を配置すること
 ないます。
 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
 ないま
 す。
- デザイボキュメトに複数の目次レベルが存在する場合:目次テンプレート内に配置された目次参照には、ターゲオとなるボキュ 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ります。

ドキュメト内にある全てのイメージのノストを、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オ少なくとも 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ク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)ターゲオン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作成します。目次参照には、ターゲオンとなる目次レベルにおける目次ブックマークの名前が含まれます。

以下のスクレージショナにある目次テンプレートでは images とう名前の目次ブックマークを参照する目次参照か合まれているレベル参照 か示されます。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(つまドキュメナト全体)にある image とう名前を持すて全目次ブックマークが参照されることでないます。



デザイン内に複数のレベルは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ットな目次を作成する必要かある場合、 2番目のレベルコジェ目次参照が存在する2つのレベル参照から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 に目次参照を配置しません)。また、目次参照のスコーププロレティにより、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 スタイルプロンティー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。

三 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さめのサンプルSPS ファイルが<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 \Style Vision 2021 \Style Vision Examples \Tutoria; \TOC 「れめられています。この SPS は 章ベースのドキュメトを記述するけるのエレテンソモデルには構成されるスキーでをベースしています。 Chapters Simple.sps を Style Vision にて開くと 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(以下のスクルーンショナを参照)。次にあるセ グションの、サンプル、階層ならびに連続的な目次では、同じスキーでをベースしたより複雑な目次サンプルについて記述されます。



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要素におキュメトの章と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(最初のノベルにある topic)が表示されます。目次の作成を行うコお以下に記されるステップを踏むことになります:

- 1. <u>目次レベル内部のデザイ構造</u>: 1つ以上のレベルをデザイボ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なります。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。
- <u>目次ブックマークの作成</u>: ステップ1 で作成された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>目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ナメのデンプレートです。デザイド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に注目してくたさい。メインデンプレートにある(その他のエレテンツ)により、グロー・シレ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・シレテンプレートとないます。

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 レネオマグロー・シリテンプレート内に配置されています(以下のスク リーンショナを参照)。このグロー・シリテ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、くつあるかに従って 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」なります。

topic (複数)
? 1 2 3 Otopic 4- Obody Oheader ↓ 'toc' ↓ 'toc' Opara
^{§div} →(num-lvl): (コンテンツ) ^{(§div}
Opara Oheader (その他のコンテンツ) Obody ?1

上のスクレージマットには、最初の条件分岐(つまし最初のレベルで章に対するtopic 要素)のエンテンカ示されています。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たさい。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つの目次レベルはしかドキュメトリまり当てられず、それが、章に対する)最初のレベルのtopic 要素となります。

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(以下のスクレ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ルの各 topic 要素に対するアンカーとして使用されます。

Oheader

目次ブックマークのプロ・ティオプロ・ティサイドバーコで編集することができます(以下のスクノーンショナを参照)。

プロパティ			×
🔓 🕂 🏭 🚠	×		
プロパティ:	属性	値	A
目次ブックマーク	日 目次ブックマーク		
	id		
	グループ	MyTOC	•
	使用されるテキスト	XPath	
	参照されていなければ削除	no	
	🕀 Authentic		
	田 共通		
			7

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・ティース対して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」ないます。IDプロ・ティーことし、作成されたブックマークインスタンスコンドしてユニークな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ケイにより、参照されていないブックマークが削除されるようになります。使用されるテキスト プロケティにより、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 ザインにおけるブックマークの開始ならびに終了タク間にあるエレテンツ、ませは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) か挿入されています(以下のスクレーンショナを参照)。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に対応すること」さいます(上記のSPSの構造とレベルを参照くたさい)。



このレベル参照内部は目次参照 (TOCref) ^{企业MyTOC} か挿入されます。この目次参照により、(i) MyTOC とう名前のブックマー クグループ内にあるブックマークの選択(上記の目 次ブックマークの作成を参照)、そして(ii) スコープ内にあるブックマークの選択が行 われます。これらの設定は目次参照が選択されている状態でプロ・ティサイドバーにて行うことができるほか、デザイン内の目次参照を右クノッ 久して、コンテキストメニューから関連する編集コマンドを選択することで行うことができます。

目次アイテムの外観は、目次の目次参照タグーンで指定することができます。目次参照タグ内部を右クトックして、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、番号のフォーマナ、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 ントをデザインニン選択し、プロ・ティサイドバーにあるプロ・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クマーケアイテム
- <u>目次テンプレートの作成</u>

7.6.4 サンプル: 階層的ならびに連続的な目次

Chapters.sps とら名前の目次機能のデモを行うすっかのサンプル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 「スレンシれてます。このSPS は 章ベースのドキュメトを記述するためのエンテンツ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、スキーマソノーウイ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ルを参照)。



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ます。

SPS には3つの目次が含まれており、以下の順序でドキュメ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. <u>チャプターとそのセクション</u>: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。
- <u>イメージのパスト</u>: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 含まれていることに注目してください。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・ドレテンプレートならびにデフォルトテンプレート か適用されます。

目次定義はtopic ならいにimage に対するグロー・シリテンプレートにて行われます。グロー・シリテンプレート内では、目次レベルが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(以下のスクレーンショントを参照)。

topic (複数)
? 1 2 3 Otopic 4- Obody Oheader ↓ 'toc' ↓ 'toc' Opara
^{§div} →(num-lvl): (コンテンツ) ^{(§div}
(その他のコンテンツ) ●body ?1

topic 要素は再帰的にデザインされているため、目次レベルならいに目次ブックマークも再帰的にないます。この再帰構造により、最初の再帰 にて下位の目次レベルならいに目次ブックマークが新たった成されます。この処理は各子孫 topic 「ズオして行われ、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)フォーマナトは目次レベルにより異なるため、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) がしべい内部に挿入されていることに注目してくたさい。目的の場所を右クレックして、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」を選択します。これにより、各 topic タイトルに対して(4.2.3 という形式の) 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 に対するグロー・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 を使っオーテキストの構築が行われている点に注目してください。使用されるテキストプロ・ティの値フィーリドにあるボタンをクリックすると、定義さ れたXPath 条件式がpara どな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ンが目次アイテムのテキストとして使用されます。

プロパティ			x
	×		
プロパティ:	属性	値	A
目次ブックマーク	申 目次ブックマーク		
	id		
	グループ	MyTOC	•
	・・・・・ 使用されるテキスト	XPath	
	参照されていなければ削除	no	
	🕀 Authentic		
	田 共通		
			7

以下のスクレーシンコナーにて示される目次テンプレートには、レベル参照(levelref) ごう かりつきまれており、そのレベル参照内部には MyTOC とう名前の目次ブックマークを選択する目次参照 ごじ MyTOC かきまれています。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)だけだなっており、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ため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クションセクション・以下のスクレーシショナに示される目次では、3つのレベルはネストされたわたちで定義されており、各レベルロ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 ブロックか含まれており、インデントなどのフォーマルを各ブロックスマオしてセル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ュメトにて、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 要素の子どなります。topic 要素にてレベリを作成した場合、ドキュメナト内にある最初、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 なります。image 要素に対して新たなレベリを作成する必要におりません。

image コオオるグロー・シリテンプレートでは、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」なります。

image (単一)
Oimage ?1 * ↓ 'images' IMG (num-seq): ↓ 'images' [= src)(コンテンツ) <= src image intentionally omitted]

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か配置されていて、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 しないとうことに注目してください。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」 により挿入され、(最初のイメージは最初の条件分岐の中にあるけっか)2番目のイメージから番号付けが行われます。番号はスォーマナするこ ともでき、(num-seq)を右クリックして、フォーマットの編集を選択するとフォーマナを指定するけっのダイアログボックスか表示されます。 このサンプルでは101とう値が定義されており、1桁の数値の前に0か挿入されることづいます。

今のスクレー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 とう名前のブックマークを識別するオタの目次参照がっ配置されており、この目次参照がレベル参照内部に配置されています。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(この値は目次参照の選択時にプロンティウイ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ルとなり、そのレベル以下にあるレベルはを番目のレベルとなります。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て選択されます。

Altova StyleVision 2021 Basic Edition



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てなります。

三 関連項目

- 目次に含まれるブックマークアイテム
- 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ドキュメントの構造をベースにした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(フラオ)まける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しています。階層的な番号はギキュメント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か与えられます。

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、個々の番号トークカ個別にフォーマトされることではます。例えば、3つのトークカら成る番号に対して A.1.i と うフォーマトを与えることに以、それぞれのトークカ理なる番号フォーマトで表示されるようになります。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にて異なる方法で割り当てられます。

連続的な番号付け(num-seq)

連続的な(まとしてうみな)番号付けは、ドキュメトデザイン内部にある。

目次ブックマークに挿入することができます(以下のスクレーンショットを参照)。

以下の操作によい連続的な番号を作成することができます:

- 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、章ヘッダー(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。



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。

上のスクレーシンコ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、以下のスクレーシンコ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 ドキュメントの階層における topic 要素シーケンスにおける位置をベースに topic 要素の番号が決定され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には num-seg プレース市レダを右クトックして、フォーマットの編集コマナドを選択します。連続的な自動番号割り当てフォーマットダイプログが表示されます(以下のスクレージョナを参照)。

連続的な自動番号割り当てフォーマット		
	n	
フォーマット文字列目勤番号:	<u> </u>	
利用可能な番号スタイル:	123 💌	
	123	
コンホホックスから利用可能なスタイル たトークンの番号スタイルを変更してく	ABC	
フォーマット文字列は xsl:number]	i ii iii I II III	
OK	one two three ONE TWO THREE One Two Three	

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、目的のフォーマットを選択し、OKをクリックすることで選択されたフォーマットを適用します(上のスクノーンショントを参照)。

構造的な番号付け(num-lvl)

階層的な番号付けは<u>デザイノこおける目次レベル</u>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に作成する必要がレベル内部のデザイン構造セクションにおける記述を参照くたさい。以下の点に留意してくたさい

• 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)。

O helpproject
O topics
<u> Sdiv</u> (num-lvl): 0header (コンテンツ) 0header
O topic 4-
Sdiv (num=lvD: Obody Oheader (コノナノツ) Oheader Obody Sdiv
O topic O body Sdiv O topic O topics O helpproject C \$XML

上のスクレーショナには、2つのレベルが示されて、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ックして、レベルとして機能するテンプレートを選択します)。上のスクレーショナで、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 レースオルダの(num-lvl) が挿入されています(レベル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)。

上のスクノーンショナ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

最初のノベリは太字により、2番目のノベリは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)プレース市レダを右クトグして、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(以下のスクノーンショナを参照)。

階層的自動番号割り当て書式…		
<u>⊅</u> ≉ ⊢⊽৩ł	→文字列自動番号:	
1. <mark>8</mark> .1		
	トークンの数: 3 <u>-</u>	
1	abc	
3	レベルを選択し、コンボボックスから 利用可能なスタイルを選択して、 各トークンに対する番号のスタイル を選択してください!	
最初の	0 - レベルを省略	
	OK キャンセル	

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ル定義は、個々のレベルに対して個別に行われるとう点に注意してくたさい。従って、各レベルにある(num-lvl) プレースヤルダに対して個別にフォーマル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。

- 三 関連項目
- <u>目次(TOC)</u>
- レベル内部のデザイン構造
- <u>日次ブックマークの作成</u>
- <u>目次テンプレートの作成</u>

7.6.6 相互参照

相互参照とおキュメン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ナと そのターゲナトに対するレクという2つのペーソニと相互参照が構築されます。ターゲナトをセナするはよ、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ナトへのレンクとなます。テキスト参照によりレクとな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つのステップにと、構築されることしています。

ステップ1:レベル

レベル内部のデザク構造のセクションに記されているようこ、ドキュメン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クだけが多ーゲットとよります。以下のスクレーンショントでは、n1:Office要素にてレベルが作成されています。

ステップ 2: 目次ブックマークの作成

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エンテキストメニューから目次を挿入|目次ブックマーク」を選択することで、目次ブックマークがレベル内部に作成されます。目次ブックマークオコは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、通常ドキュメント内にあるノードが識別され、そのノードのエンテンンが目的のテキストとして使用されます。

以下にあるスクレーンショナでは、n1:Name 要素内にある目次ブックマーク 小 10003 く 小 10003 とう名前と カントノードを選択する XPath 条件式が与えられています。これにより、n1:Name ノードのエンテンが出力テキストとして使用されます。

SXML On1:OrgChart On1:Office 4-
<mark>≸div</mark> <mark>0n1:Name</mark> Office Name <mark>↓'toc3'</mark> (コンテンツ) <mark>0n1:Name</mark> ≸div
<mark>§div</mark> ∕Reference to the office name: <mark>~↓'toc3'</mark> (テキスト参照) ~↓'toc3' ⟨§div
On1:Office On1:OrgChart SXML

XMLドキュメントか処理されると、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ケを作成するために、(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。

<u>~↓'toc3'</u>(テキスト参照) <u>~↓'toc3'</u>

上のスクレーシショナでは、toc3 と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同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づかます。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 とう名前の目次ブックマークだけが、参照のターゲナビないます。

上のスクノージョナでは、n1:Office テンプレートが表示されています。n1:Office ノードが処理されると、n1:Name ノードのエンテンを出力 テキストとするアンカーが作成されます。これは、(目次ブックマークの使用されるテキストプロ・デ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、このノードのエンテンンが出力テキストと指定されているすっかです。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りレンクのテキストが生成されます(以下のスクノーンショナトにて紫色で示されて いるテキスト)。出力は以下のようごなります:

Office name: <u>Nanonull, Inc.</u> Reference to the office name: <u>Nanonull, Inc.</u> Office name: <u>Nanonull Europe, AG</u> Reference to the office name: <u>Nanonull Europe, AG</u>

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プションは、この他にも(i)グロー・バル スコープと(ii)現在と下位のレベンはあり、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ンルこある目次ブックマーケをターゲナ とすることもできます。

三 関連項目

(C) 2015-2021 Altova GmbH

- <u>目次(TOC)</u>
- レベル内部のデザイン構造
- 目次テンプレートの作成

7.6.7 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、ハイパートンクの参照先として使用することができます。ハイパートンクはブックマークだけではよく、ウェブページのようよ外部トリースへの参照を行うこともできます。StyleVision では柔軟なかたちで、イパートンクに使用されるターゲート 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、ターゲオページへのレノを作成する方法。
- 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

ブクマーク(ハイパーレク)は、カーノルの配置位置、おは SPS コンポーネトの周ルと、SPS におお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(以下のスクノーンショナを参照)。このダイアログでは静的おけよ動的なブックマークの名前を定義することができます。動的な名前とよ(i)XMLドキュメントのコンテンから得られた値、おけよ(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

以下の操作によバ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、右クルクにより表示されるエ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. ブックマークの挿入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)。作成されるブックマークか静的なテキストの場合は 静的タブを、名前がXMLドキュメントやXPath条件式から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パーンか 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。以下のスクノーンショットでは、コンテキストノード以下のName 子要素につけてコニークな名前を持って動的なブックマークが作成されます。

ブックマークの	の挿入	
静的動的	静的ならびに動的	
	動的	
アドレス(E):	generate-id(n1:Name)	
	XPath を編集…	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。
	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
		OK キャンセル

4. 「OK」をクリックすることでブックマークか定義されます。

作成されたブックマークは、イパーレンクから参照することができます。

ᄯ 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か作成されます。静的な名前か与えられた場合、これらブックマーク全てに同じ名前か与えられ ます。そのナダ繰り返し出現する要素にすけしては、例えばエンテキストノード(ブックマークが作成されナノード)の子要素にあるコンテ ンン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 のスクレーンショナを参照)。ブックマークの参照を行うコよ同じIDをノイパーレクの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たとい XPath 条件式は concat('#. generate-id(nodeXXX)) どいます。

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へアクセスすることがで きます:

- デザインにあるブックマークを選択します。
 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)。

プロパティ			×
La 🗜 📲	PRTH ×		
プロパティ:	属性	値	A .
ブックマーク	包 ブックマーク		
	ブックマーク名	Тор	
	🕀 Authentic		
	·⊞ HTML		
	田 イベント		
	田 共通		
			$\overline{}$

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。

三 関連項目

- ハイパーレクの定義: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。
- <u>挿入 ブックマーク</u>: 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

ノイパーレンクはデキストやイメージとレオニSPS コンポーネントの周りに作成されます。ノイパーレン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、まけは(ii) ウェブページや電子メールのメッセージとレオュ外部リソースを使用することができます。このセクションではレイパー リンクのコンテンソ(テキストやイメージなど)について、そしてノイパーリンクのターゲットについて説明します。

ノイパーレンクの作成

以下の操作により、イルシン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。

ハイパーレンクのターゲトを定義

<u>タグで囲む / ハイパーリンク</u> おけま <u>挿入 / ハイパーリンク</u>によりアクセスすることができる イパーリンクの挿入ダイアログでは イパーリンクの ターゲットを作成することができます(以下のスクリーンショットを参照)。

ハイパーリンクを挿入	×
責任的 ■ 動的 ■ 責任的ならびに動的 ■	
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	
アドレス(<u>E</u>):	
□ 絶対パス	単純な顔明ワンドレスの人プロこ、 上のフィールドを使用してくださ
	OK キャンセル

リンクのターゲットは以下のどれかしています

- 同一 SPS デザインにある <u>ブッ/ァーク</u>: ターゲオ URI はフラグメオ識別子どむます。
- 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もフラグメト識別子どなます。
- <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 ~ンから構成されるもの まけまい ~ スされていない エンティティのどれかどなります。

以下ではこれらのターゲナを定義する方法について説明します。ハイパーレンクの挿入/編集ダイアログにてURIを定義して、「OK」をクリックして完了します。

ブックマークへのレンク作成

以下の操作によりブックマークへのレクを作成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。

⑤ ドキュメント内のブックマークを選択	×
ドキュメント内の実在するブックマークを選択:	
DocumentTop generate-id(name) (動的)	
ОК. + +>ЭН210	

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)。

ハイパーリンクを動的ブックマークに作成するには、生成されたブックマーク名の1つにに一致する XPath 条件式を入力してください。ブックマーク名は以下に示される(ブックマークデザイン要素によ り定義される)XPath 条件式により生成されます。 動的ブックマーク XPath: generate-id(name) ブックマークの XPath (こマッチする XPath 条件式を入力: XPath を編集	動的ブックマークへのハイパーリンク作成	×
動的ブックマーク XPath: generate-id(name) ブックマークの XPath (こマッチする XPath 条件式を入力 : XPath を編集 のK キャンセル	ハイパーリンクを動的ブックマークに作成するには、生成されたブックマーク名の1つにに一致する XPath 条件式を入力してください。 ブックマーク名は以下に示される(ブックマークデザイン要素に り定義される)XPath 条件式により生成されます。	ደ
generate-id(name) ブックマークの XPath (こマッチする XPath 条件式を入力 : 	動的ブックマーク XPath:	
ブックマークの XPath (こマッチする XPath 条件式を入力 : 	generate-id(name)	
XPath を編集 OK	ブックマークの XPath (こマッチする XPath 条件式を入力:	
XPath を編集 OK		
OK キャンセル	XPath を編集	
	OK キャンセル	

<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。このようズブ・クマークへのレンを作成する場合、アンカーはランタイ ムこて動的に生成されるため、このようズアンカー「ごさして、ハイパートンクのhref 値をどのようこセントするかか問題」ごかます。動的に生成されるアンカーゴごしてはgenerate-id() 関数を、ハイパートンクのhref 値の内部で使用することごかます。変換が一度だけ行われた場合、generate-id() 関数が同一のノード「ごさして評価されると、常に同一のID が生成されます。これにより、ブックマークとハイパーレンクにて 生成されたID が同一のとかります。

以下の2点に留意する必要があります:

- 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(nodeXXX))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。

外部リソースへのレンク作成

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 <u>ていない エンティティ</u>名のどれかを表すテキスト文字列を与えることはよります。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)。

ハイパーリンクを挿入		×
静的 動的 静的ならびに動的		
青銅白	動的	静的
アドレス(E): mailto:		
2つの静的ならびに1つの動的表現によりアドレスを 作成します。	XPath を編集	
	□ 動的なパートを XSLT 変換に対して	こ分析されていないエンティティとして扱う
		OK キャンセル

パースされていないエンティテ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のセクタンを参照ください。

ハイパーレクプロパティの編集

ハイパーレンを編集するココンイパーレンクの開始ませは終了タグを右クリンクして、コンテキストメニューからURIの編集を選択します。ハ イパーレンクの編集ダイアログカ表示されます(上のスクリーシショナを参照)。プロパティウィンドウのアイパーレンクグループ内にあるURLプロ パティからも、ハイパ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。

ノイパーレンの削除

ハイパーレンを削除するコよハイパーレンの開始ませま終了タグをクリックしてハイパーレンを選択し、Deleteキーを押下します。ハイパーレンを選択し、Deleteキーを押下します。

🖃 関連項目

- <u>ブッケマークの挿入</u>
- ハースされていないエンティティ URI
- <u>挿入 ハイパーレク</u>

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コンポーネトに対して定義されます。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グアップへ変換されます。(Enterprise Edition のAuthentic View HTML、RTF、PDF およびW ord 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 エディションの比較ページを参照くたさい。

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まCSS2の構文が使用されます。スタイルは外部のスタイルシートにて、SPS 全体に対してグローイ シレ に、そして特定のコ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、このセクションのサブセクションである <u>CSS</u> 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、ティサイドバーコで直接入力することができるまか、<u>XPath 式</u>を介してセットすることもできます。XPath 条件式を使用することで、(i)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たかたちでプロ、ティの値を割り当てることができるようこなります。

その他にも、SPS デザインでは特定のHTML 要素をSPS コンポーネトのマークアッ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。このようなマークアップは SPS コンポーネントのスタイルとしても使用することができます。CSS ス タイルは定義済みのフォーマナトにつけて適用されると、CSS スタイルは定義済みフォーマナのスタイルよりに高い優先度で処理されます。定 義済みのフォーマナトについては、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。

- 三 関連項目
- <u>使用方法の概要</u>
- <u>デザインサイドバー</u>

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ト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 クアップされます。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 <u>ローカルスタイルを定義</u>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、ツールバーにあるフォーマナドロップダウンリ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。

標準	•
フィールドセット(fieldset) ブロック(div) ブロックはIPA look quote)	•
フロック51月101000Kquote/ 中央揃え(center) 整形済みテキスト (pre wra	
標準 段落(p)	
見出し1(h1) 見出し2(h2) 目山し2(h2)	
元田(J3VI3/ 見出(J4(64)	•

固有のスタイル

StyleVision にて使用される定義済みフォーマナには、以下にあるスタイルコンポーネントの方の1つませま両方が含まれます:

- テキストスタイルコンポーネト
- 余白ゴノポーネント

例えば、定義済みの段落(p)フォーマトには余白コンポーネトだけが含まれており、選択されたコンポーネト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。それとはおりに、見出し1(h1)フォーマナトにはデキストスタイルコンポーネントと余白コンポーネントの両方が含まれます。

定義済みのフォーマナー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エンポーネトは全種類のSPS エ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 げ適用されます。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ナを適用すると余白エンポーネントが適用されるのに対し て、テキストスタイルエンポーネントは適用されません。
- 定義済みフォーマト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マ扎て適用されません。
- 同一のエレポーネトーマオーて適用される定義済みフォーマナはいつだけです。
- 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)等角フォントが使用されます。テキストにより構成される段落のように、改行を持たされ、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トのための追加スタイルを定義する

定義済みのフォーマト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。

□ 関連項目

• <u>CSS スタイルをローカルに定義する</u>

出力のエスケープ 8.2

テキスト文字列内にある文字が文字参照ませまに、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なります。参 照は(文字参照かエンティティ参照かば関わらず)アノッサンド(&)文字で始まれミコロン(:)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#x41; 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 HTML(ならびに XML)エンティティ参照は wit; 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しいます。入力にてエスケープされた文字だけが出力エスケープできるという点に注意し てくたき、く以下のテーブルはある例を参照くたき、)。SPS では以下のものです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 プレース市ば

プロンティのテキストグループである disable-output-escaping 属性にて設定を行うことができます。プロンティのデフォルト値はしいえ」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ンォルトでエスケープされます(以下のテーブルこある例 を参照くたれい。

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。 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になっている場合、選択されたテキストはその出力フォーマナトにてエスケ ープされず、その他の出力フォーマットにてエスケープされます。

出力エスケープが無効/有効していている状態で、テキストか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
&	ાહ	&
&	こと	&
æ	ાપદ	&
ŵ	こと	&

<	ાત્ર	<
<	ほ	<
A	ાપ્દ	A
A	ほい	А
<	ાત્	<
< <	いえ	< <
<pre>&lt; < &amp;lt;</pre>	いえ 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。

出力フォーマトごとコ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」さいます:

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ウ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にないます:

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クアップが、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)エスケープされていかい テキストは、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、出力エスケーン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 プロンティを使用する際には、この違いについて常に留意する必要があります。

- □ 関連項目
- <u>テキストとてXML コンテンンを挿入</u>
- <u>自動計算</u>
- •<u>デザイビューシンボル</u>

8.3 値フォーマット(数値データ型のフォーマット)

値フォーマトを使用することで、数値型のXMLスキーマデータ型ノード(アのノスト)に収められたエンテンンを、元のデータ型を使って表示と は違うフォーマル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 ンコンポーネントが付随した形式になります。値のフォーマルは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ルを使用することもできます(アのノスト)。

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

- 値フォーマナ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
- メモ 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。

値フォーマトか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ト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
- メモ 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せん。
- 三 関連項目
- 日付のフォーマナ

8.3.1 値フォーマットのメカニズム

値フォーマナーは以下のアイテムコマ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 <u>数値型</u> のデータ型ノード
- 数値型のデータ型フォーマ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 Iズルてノードお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。

値フォーマット	×
型 'date'のフォーマット 〇 書式なし (標準書式 = 'YYYY-MM-DD') ④ XML スキーマ値としてフォーマット	
XMLスキーマ値のオプション XMLインスタンス内のデータタイプ: date 出力ドキュメント内のフォーマット: DD Month YYYY	
✓ フィールドの挿入(!) > フィールドオプション(E) エラーへ移動(E) ✓ 同じフォーマットを XSLT 出力へも適用(フォーマットオプション全ては適用できません)	
OK (<u>D</u>) キャンセル(<u>C</u>)	

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。

値	[フォーマット	×
	コンテンジフィールドのフォーマット	
	○ 書式なし	
	● XMLスキーマ値としてフォーマット	
	○ インライン XBRL 値としてフォーマット	
	○XML スキーマ値のオブション	
	XML-1ンスタンス内のデータタープ: integer (or related)	
	出力ドキュメント内のフォーマット: +##0 ##0	
	✓ フィールドの挿入([] > フィールド ⁰ (##0)	
	✓ 同じフォーマットを XSLT 出力へも適用(フォーマットオプション全ては適用できません)	
	OK (<u>0</u>) キャンセル(<u>C</u>)	

- 3. コンポーネトの値がフォーマトされたわたちで表示するのかを指定することができます。出力をフォーマトされないかたちで表示す るコよ、書式なしラジオがなを選択してくたさい。XMLスキーマ値としてフォーマットラジオがなた選択することで、値 のフォーマトを行うことができます。値のフォーマトを行わない場合、選択されたノードや自動計算結果のデータ型に対して標準 的なフォーマト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 ロップダウンリストから)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あります。
- 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、値フォーマナ構文を参照くたさい。

構文のエラー

構文内にエラーかある場合、以下のような振る舞いか行われます:

- 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」ボタンが無効しています。
- 値フォーマトダイアログのエラーへ移動ボタンが有効になります。ボタンをクリックすると、フォーマト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トのミスマッチ

XMLノードに入力されたデータがそのノードのデータ型の形式にマッチしない、ませま自動計算の結果が期待されたデータ型の形式にマッチしない場合、フォーマトは未定義となり、正しいかたちで出力で表示されません。

値フォーマトを出力へ適用する

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。

全てではおよものな、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あります。その他の出力へも値フォーマトを適用す るさよ、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ン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

値フォーマトの構文は以下のようコなります:

([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ト おけよテキスト どより、 {field-option(s)} は更なるフォーマトオプションを定義するオプションの修飾子 どよります。

定義構文の説明

値フォーマトの定義は以下のように構築されます:

- 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)。
- フィールドはお必要に応じて使用できるフィールドオプションを含めることができます。各フィールドオプションは角かっこ内に含め、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ションを使用する場合、コンマ((パ) にて分離する必要があります。例えば、DD Month[uc,ro] YYYY とり定義では、uc とro がMonth フィールドに対するフィールドオプションどなります。

例

xs:decimal データ型に対する値フォーマトの例を以下に示します:

*"\$"(##*0.00)

出力は以下のようづなります:

\$ 25.00 \$ 25.42 \$267.56

xs:date データ型に対する値フォーマトの例を以下に示します:

DD Month{uc,ro} YYYY

ここで、 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プションとなります。

出力は以下のようてなります

24 SEPTEMBER 2003

フィールドタイプ

フィールドタイプは、データのコンポーネトと、そのコンポーネトかどのようご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トは、接頭部ならびは接尾部にある修飾子やフィールドオプションイント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。値フォーマトダイアログのドロップダウンメニュートゴは、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ト定義が含まれます。定義を選択し、接頭部の修飾子、按尾部の修飾子、フィールドオプションを必要に応じて加えること」ごがます。

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 しています
0	指定された場所に対応する数値がむ、場合、0 に なります
	小数点
,	数値グループのセッシーター
Υ	年
у	年(base = 1900)、下記のメモを参照(かざい)
MM	月、2文字分の長さが必要
DD	日、2文字分の長さが必要、
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文字分の長さが必要
НН	時(1から12)、2文字分の長さが必要
mm	分、2文字分の長さが必要
SS	秒、2文字分の長さが必要
AM	AM #dtPM
am	am およpm
AD	AD おさまBC
ad	ad #Jabc
CE	CE #JJBCE
се	ce #Jabce

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て関するメモ

データンポーネトの長さ、関する以下の点に留意してくたさい

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。

接頭部ならびま屋部の修飾子

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リドのポンティブ/ネガテ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に示します。

接頭	接尾	説明
<		左揃えを行い、テキストに対してデフォルトの振る舞しを行います。数値に対しては デフォルトで右揃えとなっています。
>		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ボルを付け加えます。数値に対するデフォルトのオプ ションイズルます。

</th <th></th> <th>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。マイナス記号の後には左 寄せされた数値が続きます。</th>		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。マイナス記号の後には左 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。数値は右寄せにより表示されます。
-	-	負の数値に対して、マイナス記号が隣り合わせて表示され、そうでない場合は空白 か表示されます。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か表示されます。
<-	>-	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どよります。 数値と記号は減り合って表示され、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。数値は右寄せに以表示されます。
+	+	常にプラスませま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。
<+	>+	常にプラスませま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。
<+>		常にプラスおけまマイナス記号を表示します。記号は左寄せて表示され、数値は右 寄せて表示されます。
()	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が数値の隣に表示されます。
<(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左寄せで 表示されます。
<(>		負の数値の場合かこか表示され、それ以外の場合は空白が表示されます。左かっ こか左寄せて表示され、その後に数値、そして数値の隣に右かこか表示されます。
[]	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。
*	*	(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。
٨	٨	オプションで定義された文字の挿入。
	th	英語におお序数(1st、2nd、3rd、4th など)。
	ТН	英語におお序数(1ST、2ND、3RD、4TH など)。

フィールドオプション

フィールドオプタンにより、高度な修正をフィールドレマオして行うことができます。以下のオプタ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(デフォルトで有効)。
dec= <char></char>	小数点の文字を指定(デンォルトではピリトド)。
sep= <char></char>	数値の区切りに使用される文字を指定(デフォルトではエンマ)。
fill= <char></char>	塗りついて使用する文字を指定します。
base= <year></year>	年フィールドのbase 年を指定(以下のメモを参照)。
pos	プラス記号だを表示、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う欧州式の数値フォーマト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つ使用される値フォーマトは以下のようつむます:###,###.##(dec=,,sep=.)。

フィールドは英国式のフォーマトが指定されたままったって、るー方で、dec ならびに sep オプションによ数値の小数点と区切り記号が Windows OS の地域設定に近い指定されている点に注目してくたさい(コントロールパネル| すべてのアイテム| 地域 | フォーマット)。

三 関連項目

- 値フォーマトのメカニズム
- 日付のフォーマナ

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>スタイルルポネリ</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。
- グローバルスタイルして: 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、これらグローバルスタイルが 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パシーサイドバーにて定義 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<u>ローカル</u>で: HTML 出力では これらのレールが個々の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 <u>スタイル</u>サイドバーニて定義されます。

上記のメカニズムによりスタイルを作成する方法の詳細については、このセクション以下にあるサブセクションにて記述されます。

用語解説

CSS スタイルシートは1つ以上のスタイルルールによ、構成されます。ルールは以下のように表示されます:

H1 { color: blue }

おけお

```
H1 { color: blue;
margin-top: 16px; }
```

各ルールコキセノクター(上の別におけるH1)と宣言(color: blue)か含まれます。宣言はプロケティ(例えば color)と値(blue)か含まれます。各プロケティと植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)ならびにスタイルルポネーリサイドバー(グロー・シルスタイル)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おり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グロー・シルスタイルより生低く、グロー・シルスタイルの優先度はロ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たものとして扱われ、後からインポートされた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。
- 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りに高い優先度を持つようてなります。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 スタイルの方、よい下部にあるスタイルが高い優先度を持つようてないます。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くたさい。

CSS ザポート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、ージョンのIE に対してデザインを行うのかが重要にないます。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、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 出力ドキュメトのスタイルを、<u>Internet Explorer の古い新し、解釈</u>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 <u>プロハティ CSS サポート</u>を参照くたさい。

メモ CSS の仕様に関する詳細については、<u>http://www.w3.org/TR/REC-CSS2/</u>を参照くたさい。

- 三 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 坊 十</u>
- <u>モジュールSPS</u>

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:

- <u>外部 CSS スタイルシートを SPS へ追加する</u>
- <u>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</u>
- 優先順位を変更する
- <u>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</u>

外部 CSS スタイルシートは、<u>スタイルレポネリサイドバー</u>とデザイン概要サイドバーと、う2つのサイドバートこて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、あるサイドバートこで行われた変更は、自動的に他のサイドバートも反映されま 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:

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)。
- 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 デザイン概要サイドバーからも外部 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たさい):

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 (スタイルリポジトリウンドウェンで定義することができます)
 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
```

スタイルルボネリウイボウでは、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ンをクリック して、ドロップダウノノストから目的のメディアを選択してくたさい。外部 CSS スタイルシートに定義されたルレールがスタイルルボネリウイボウニ 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-シンボルをクリックすることで、ツノー内部にあるセノクターアイテ 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レポネリソール・トーエあるリセットポタンをクリックしてくたさい。



外部 CSS スタイルシートの優先順位を変更する

スタイルルポネリウイボウニ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〜インポートされます。 HTML ファイルではおよ下のような記述がなされます:

<html> <head>

Altova StyleVision 2021 Basic Edition

</1101112

HTML ファイルにて表示されるファイルの順序は、スタイルルポネリの外部ソレー以下ならびにデザイン概要サイドバーのCSS ファイルソレー 以下に表示されている順序に従います。スタイルルポネリにて CSS ファイルの順序を変更するコよ 優先順位を変更するスタイルシートを 選択し、スタイルルポネリソール・トーンある上へ 🎱 ポタンまナゴよ下へ 💟 ポタンを使用することで、ツレーンおけるスタイルシートの位置 (優先順位)を変更してくたさい。デザイン概要サイドバーでは CSS スタイルシートの編集ポタンをクトックして、必要に応じて上へまナゴよ下 へ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か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かるここれてスタイルシートのインポート優先度が下かることご注意してください。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 ルールが同じセンクターを持っている場合、よど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。

デザイン概要	×	
▷ ソース		
▶ モジュール		
▽ CSSファイル		
🛃 QuickStart.css	•	<u> </u>
CommonStyles.css	6	XMLSpy でファイルを編集(Y)
<u>新しいCSSファイルを追加</u>	- -	
▽ パラメーター		L∧(U)
新しいパラメーターを追加		下へ(D)
▽ XSLT ファイル		
<u>新規XSLTファイルを追加</u>	\mathbf{x}	削除(<u>R</u>)
		全ての CSS ファイルからスタイルを混合(M)
	•	XPath で CSS ファイルを選択(<u>S</u>)

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レ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
- グローノ 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・シルスタイルはSPS デザイン全体に対して適用されます。スタイルレポネールはスタイルレポネールのグロー・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。

スタイルルポネリでは、セレクターとそのセレクターー、対応するCSS プロノティから構成されるCSS かジローノ 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 次のセノ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

グローノジレなスタイルを追加する

以下の操作によりグロー・シルスタイルを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> <u>定セグションを参照</u>(たさい)
- 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、

ᄯ

- 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・シレスタイルの編集と削除

スタイルセレクターとそのプロ、ティは、スタイルルポネリウィドウィン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たさい。スタイルコお種類の表示方法があり、それらのビューと、ビューの切り替えについては定義ビューのセクションを参照くたさい。

グロー・ジルスタイルを削除するコは、そのスタイルを選択して、スタイルルポネリソールレイコあるリセットボタンをクリックしてくたさい。

グロー・ジレスタイルの優先順位を変更する

スタイルドネリウイボウニて割り当てられたグロー・シルスタイルは 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:

```
<html>
       <head>
              <style>
                     <!--
                                    { color:blue;
                     h1
                                      font-size:16pt;
                                    1
                     h2
                                    { color:blue;
                                      font-size:14pt;
                                    }
                             { color:red;}
                      .red
                      .green { color:green;}
                     .green { color:lime;}
                     -->
              </style>
       </head>
       <body/>
</html>
```

グロー シレスタイルがAuthentic View ならびにHTMLドキュメトにて表示される順序は、スタイルルポネールのグロー・シレソレ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は意味があり、2つのセノクターにと同じノードが選択される場合、 とい下部に表示されているセノクターが、とい高い優先順位を持つことはないます。例えば、上記のルールがHTMLドキュメトに表示されて おり、HTMLドキュメト内に<h1 class="green">という要素と、3つのグロー・シレスタイルがこの要素に一致する場合: h1 セノクタ ーとつの.green ケラスセノクターを持つ場合。色 greenを持つ.green セノクターの後に発生し、優先順位を与えられなけな、and 色 limeを持つ.green セノクターのプロッティが適用されます。(ケラスセノクターよし、一ドセノクターよりま高い優先順序が与えられるた め、green セノクターゴは、h1 セノクターでが適用されます。、h1 セノクターよりま高い優先順位が与えられます。 Authenter マオントロンターがする位置に関わらず、h1 セノクターよりまに、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 しかしながら、<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ッティを有するセノクターが存在しないた。 め、<h1>要素に適用されます。

グロー・シリスタイルの優先順位を変更するコよそのスタイルが選択された状態で、スタイルレポネリソール・トーエある上へならひに下へボタを使用します。例えば、greenグロー・シリスタイルが.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・ティントに、優先度を持つようしています。

クラスセレクターは型のセレクターよりにでいたことに注意してくたさい。 つまし、セレクターの順序が.red .green h1 h2 に変更されると、h1 とh2 は、green のままです。

三 関連項目

- · <u>スタイルポネリ</u>
- <u>外部 CSS スタイルシート</u>
- ローカルスタイル
- <u>CSS 切卡</u>十

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、コンポーネントに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・ジルのスタイルルールならび にすい部 CSS スタイルシート より氏高い 優先度を持つこと ゴムルます。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・ハツール、 ー おっは スタイル サイドバート こて定義されます(グロー・ジルスタイルの定義と外部 CSS スタイルシートの定義は スタイルルポントリ で行われ る点に注意してくたさい)。

フォーマトツール、ー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、フォーマナ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(以下のスクノーンショナを参照)。

書式		▼ ×
標準	- B I <u>U</u> S	• • 🔍 🖓 🖓 📑 🖷 🗏 🗄 🏀

div やh1、pre といた定義済みのHTML フォーマナルに対して、テキストのスタイル、背景色、テキストの配置、ハイパーレンを適用することができます。詳細については、フォーマナッツール、一のセクションを参照くたさい。

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、ローカルにおお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。

スタイル		×			
占 📲 📲 🛛	님 4월 4 🗏 📠 🗙				
<u>スタイル:</u>	周性 値				
 1 コンテンツ 	⊞ IE				
	·⊞ UI				
	🕀 XSL-FO				
	田 テキスト				
	田 テーブル				
	田 フォント				
	田 ボックス				
	田 共通				
	田 効果				
	田 印刷媒体				
	⊕ 色				
	⊡ 詳細				
	田 順序付きリスト				

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。

スタイル				×
占 🕂 🍓 🕴 🚠	×			_
	属性	値		A
- 16 テキスト	フォント			
2 自動計算	font-family	Arial		
17 コンテンツ	font-size	10pt	$\mathbf{\nabla}$	
	font-weight	normal, bold		
<u> こ 1 コンホホックス</u>				
L 1 1メーソ - 11 約茲				
2 U2 P74				
- 7717 1724				
				$\overline{}$

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

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。

三 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 サポートのノベルを調節する出力ドキュメトの設定に関する説明

8.4.4 スタイルの値の設定

スタイルサイドバー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コンポーネトに対してスタイルプロ・ティを定義することができます(以下のスクノーシンヨルを参照)。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コンポーネントを選択することもできます。複数種類のコンポーネントが選択されている場合、スタイル定義ペイントこて定義した値が、選択されているコンポーネントの全インスタンスンプして適用されること」ないます。

スタイル			x
占 📲	PATH ×		
スタイル :	属性	値	_
4 テキスト	中 フォント		
<u>-1 自動計算</u>	font-family	MS Pゴシック	
<u>□1 編集74</u>	font-size	large	
3 段洛	font-style	inherit	
	font-variant	normal	
	font-weight	lighter	
	曰 共通		
	class	explanation	
	- 臣		
	color	blue	📼 🛞 🔄
			×

スタイルプロノティグループ

スタイル定義ペイノーて編集することのできるスタイルプロ・ティイは、以下のスクノーシンヨントで示されるように、グループごとこ管理されます。

Х АТИ	×
🔓 🕂 🏭 I 🚠	×
<u>スタイル:</u>	属性 值 人
 1 コンテンツ 	⊕ IE
	🕀 UI
	🕀 XSL-FO
	田 テキスト
	田 テーブル
	田 フォント
	田 ボックス
	田 共通
	田 効果
	田 印刷媒体
	⊕ 色
	① 詳細
	田 順序付きリスト
	v

値のあるものを表示、全て展開と、全て縮退、ソールドーボタンシールドーボタンを使用することで、プロパティ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ットを参照)。

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。

ha 142 🐙 1 . 🗛 1 ×			
スタイル: 属性		値	^
 ■ 1 コンテンツ ① IE 			
·⊞ UI			
H XS	L-FO		
· 田 テキ	F7F		
· 🕀 🖵	ーブル		
· 🕀 7a	シト		
田ボ	ックス		
田共	通		
田 効:	果		
·⊕ €∏,	刷媒体		
日色			
	background		
	attachment		
	- color	navy	I 📼 🕥 👘
	image		•
	- position	@position	PATH
	- repeat		
· · · · · · · · · · · · · · · · · · ·	color	no-repeat	<u>~</u> 😯
· 田 詳	術田	repeat	
	序付きリスト	repeat-x	
		inherit	
			Y

• 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、テ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、スタイル値をルセナ)するコよスタイルサイドバーのソールレイにあるノルクトポタンをクトックしてくたさい。その他にも、プロノティの値カラムをダブルクトックして、Delete おけま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

- <u>スタイルサイドバー</u>
- CSS スタイルをローカルに定義する
- <u>CSS スタイルをグローイ ジルニ定義する</u>
- <u>外部 CSS スタイルシート</u>
- <u>CSS 圳 小</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、スタイルプロノティの値を 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 ドキュメトト あるノードへのアクセスを行うことが可能になります。スタイルの定義を XML データから取得するさますではよく、XML データのコンテンソや構造により、異なるスタイルを与えることも可能になります。例えば XPath 2.0/3.0 のif...else ステート メントを使用することで、シーケンスにおける要素の位置 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、以下のスタイルプロ、ティーズ扎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、ティ値に対する静的モードと動的 (XPath) モード

<u>XPath 式の有効化</u>プロ ディコよ 2つのモードが用意されて 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/color1 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・ティでは、静的モードがデフォルトで選択されています。プロ・ティを動的(XPath)モードに切り 替えるコよ、そのプロ・ティを選択してサイドバーのソールレーニあるXPath アイコンをクリックします(以下のスクリーンショナを参照)。

スタイル	×
占 🕂 🏭 📕 🚠	×
2,91N : 🖊	居性
 1 コンテンツ 	⊕ IE
/	⊕ UI
/	⊕ XSL-FO
/	표 テキスト
/	田 テーブル
/	田 フォント
静的モードと動的	田 ボックス
モードの切り替え	田 共通
	田 効果
	④ 印刷媒体
	🖓 色
	🕀 background
	attachment
	color if (position 0 mod 2 = 0) then 'red' e 👬
	image
	position
	repeat XPath 条件式の編集 💽
	color 💌 🔞
	⊞ 詳細
	田 順序付きリスト

静的な値がそのプロ・ティーズ対して既に入力されている場合、その値がアプされ、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」をクトックすることで編集を完了します。

プロ・ディーズ北てXPath 条件式が入力されると、そのプロ・ディの値カラムこXPath 条件式の編集ボタム表示されます(上のスクノ ーンショナを参照)。このボタン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 うコよ、ツール・イニある XPath アイコンをクリックしてくたさし、。定義された XPath 条件式がフルされ、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、元に戻すコマノボコマンドを使用してください)。SPSを編集した後に再度プロンティを選択しても、過去に選択されたモードにてプロンティは開かれます。

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へ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・ティが既に動的モードになっている時にソールドーの XPath アイエ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てとご注意してくたさい。
- プロ、ティーズ対して 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 条件式の編集ボタンをクトックすることで XPath 式の編集ダイアログ ヘアクセスすることができます。ダイアログゴは、そのプロ・ティーズ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 条件式の入力おけは編集を行った後は「OK」をクル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た値をプロイ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 しる XML ドキュメント のコンテンソン特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なります。 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 ー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。

- 三 関連項目
- CSS スタイルと作業
- <u>XPath ダイアログ</u>
- <u>7911</u>
- <u>プロ/ ディ</u>
- <u>CSS ザポト</u>

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:

- <u>自動計算</u>
- <u>(コンテンソ)プレースヤレダ</u>
- 段落(ブロック)デザイン要素
- <u>テーブルセル</u>

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ノティーこと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ソースドキュメトのdesc-syte属性には、デフォルトの合成スタイルは含まれます。

<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ハティーて値を入力する際に、合成スタイルの切り替えを行うことができます(以下を参照くたさい)。

メモ: XMLドキュメントを妥当なものとするために、合成スタイルヘ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パティ

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 ユーザーによりこの合成スタイルを編集することができるよう」ないます。

以下の操作により、デザインコ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。

スキーマツリー × ↑□ ▼ ⑩ × ● ⑬ × ● ⑬ × ● ⑬ × ● ⑬ × ● ⑬ × ● ⑬ × ● ⑬ × ● ⑬ × ●	On1:De	_{ssc} (コンテンツ) <mark>(O</mark> n1:Des	SC .	
ロンルート要素 ロンルート ロンルート ロンルート ロンルート ロンルート ロンルート ロンルート ロンルート ロンルート ロンルート ロンの ロンの ロンの ロンの ロンの ロンの ロンの ロンの	プロパティ 19 1 12 48 プロパティ: コンテンツ テンプレート	 ・ ・ ・	(いいえ ■ @desc-style @class ▼	

2. コンテンソコンポーネントの合成スタイルプロッティー表示されているコンボドックスに、コンテキスト要素の属性が表示されます(上のスクノーンショナの右下にあるプロッティナンド・ゲを参照くたさい)。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たさい、コンボドックスコン空のエントリーも表示されている点に注目してくたさい。

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、てセルするコよプロ、ティサイドバーのソール・トースあるXPath アイエンをクトックして、表示されるXPath ダイアログにて 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ルでは、それに準ずる値)になります。

例えば、以下に示されるn1:Person 要素のバコンテンツプレースヤルダーて作成された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 とう子要素が含まれておし、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むます。XPath 条件式によし返された値は、 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 にお生成された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:

<span style="{if (number(n1:Shares) gt 1000) then 'color:red' else
'color:green'}">

HTML 出力では、条件の評価結果により、以下のどちらかか生成されます:

おは

```
<span style="color:green">
```

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
- <u>XPath ダイアログ</u>
- <u>2911</u>
- <u>プレ ም</u>ィ
- <u>CSS 圳 小</u>

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トの編集を行うこともできます。
- パースされていないエンティティ URI: XML ドキュメントがベースとしている DTD に ペースされていない 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、イメージベッイパーレンケを URI のターゲート URI として使用することができます。
- <u>スクレプトの使用</u>. StyleVision には 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。
- <u>HTML のインポート</u>:HTML ファイルを Style Vision にインポートすることができるほか、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 からの新規作成: 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 におけるスタイルレールを作成することができます。SPS コンポーネントを修正して、コンテンツやフォーマトをSPS に加えることができます。
- 三 関連項目
- プロノディサイドバー

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、その中でいースされていないエンティティを宣言した場合、そのエンティティー関連付けられているURIをSPS内にある イメージやいイパートンクのターゲナ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にな 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ドキュメト、SPSドキュメトを適切に編集する必要がみます:

- 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、パースされていたい いっていた (ii) (エンティティを宣言する必要があります。
- 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 必要があます。
- 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・イパートレクのターゲオ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サブセナロコンースされていないエンティティが2つ含まれており、1つコはGIF表記がもう1つコはLNK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 イパーリンク

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ッイパーレン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。

イメージ	の挿7	<u></u>	
靜的	動的	静的ならびに動的 インライン データ	
アドレス	ξ(E):	争的 http://www.altova.com/pix/Marketing/logop	ng
	[· 絶対パス 参照…	単純な静的アドレスの入力に、 上のフィールドを使用してください。

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、エノティティ名に関連付けられる名前がエノティティ名の代わりに使用されます。

- メモ 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
- <u>イメージの使用</u>
- 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 デザイノよ HTML の出力を行うすめのXSLT ファイルや、PDF ならびにFO ファイルを出力するすめのXSLFO コマンドを伴う既存のXSLT ファイルをベースコ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び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、おは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、<u>デザ个ビュー</u>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ゾル、スタイルや表示方法の修正や新たなスタイルの追加、自動計算と条件テンプレートといた。Style Visionの機能を使用することができます。
- 6. SPS を保存して、作業用 XML ファイルを使用した多種の出力の書式のプレビューを行うことができます。更に生成されたファイルの保存 コマンドにより、スタイルシートと出力ファイルの生成を行うことができます。

サンプル

以下に記されるサンプルは、(マイ)ドキュメナフォルダー 以下にある、C:\Documents and Settings\<username>\My Documents\Altova\StyleVision2021\StyleVisionExamples/Tutorial/NewFromXSLT になって、 のフォルダーゴは SimpleExample.xslt、SimpleExample.xsltx.SimpleExample.Xsltx.SimpleExample.Xsltx.SimpleExample.Xsltx.SimpleExample.Xsltx.Si

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>
                <Year id="2008">180000</Year>
                <Year id="2009">140000</Year>
                <Year id="2010">100000</Year>
        </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。

<pre>?xml version="1.0" encoding="UTF-8"?> xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"</pre>
mins:xs="http://www.w3.org/2001/XMLSchema" xmins:tn="
crp://www.ws.org/2005/xpath=tunctions">
explored methods will version 1.0 encourne of the indence yes //
<pre></pre>
cheads
<pre><title>Simple Example for New From XSLT</title></pre>
 kody>
<pre></pre> <
<pre><xsl:template match="Data"></xsl:template></pre>
<pre><xs1:for-each_select="region"></xs1:for-each_select="region"></pre>
<h1 style="color:red"></h1>
<pre><pre><si:apply-templates select="@id"></si:apply-templates></pre></pre>
Asistylesheets
, An experimental second se

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:

イニシャルドキュメントセクション	プロパティを編集	ヘッダー ワッターを追加
□\$XML (コンテンツ)		
Data (単一)		
O Data O Region		
Sh1 = id < ≤ select="." />	= id \$h1	
O Region O Data		

SPS はおつのデンプレートが作成されている点に注目してくたさい。HTML プレビュートごりい替えを行うと、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。



デザインビュートコント1 要素を選択し、(スタイルサイドバートある色プロ/ディグループにて)colorの値をblack へ変更します。その後スキーマソーからYear 要素をドラッグして、以下のスクレーン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。

Data (単一)
O Data O Region
Image: Shirt state state Image: Shirt state
id Year
◎ ▶ = は (コンテンツ) (= は (コンテンツ)
O Region Data

ドキュメトのコンテンソ、構造、表示方法プロ・ティーその他の変更を加え、プレビューーこてその出力を確認した後に、<u>生成されたファイルの</u> 保存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。

- 三 関連項目
- SPS ファイル: エレテンツ
- <u>SPS ファイル 構造</u>
- SPS ファイル プレゼテーション

9.3 ユーザー定義 XPath 関数

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、コンポボックスなど)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け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。

デザインツリー ×
🕮 🗸 🚬 🎆
🚯_Altovasps
🖽 🔁 スクリプト
田 🗰 メインテンプレート
🖽 🖏 グローバルテンプレート 🛃
💼 デザインフラグメント 🛨
□ f XPath 関数群 →
🗹 📝 🕺 sps:factorial(num)
デザインツリー スキーマツリー

ユーザー定義 XPath 関数を作成するコよ XPath 関数アイテムこある 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(以下のスクレーンショナを参照)。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リックします。XPath 関数ダイアログが表示され、関数の定義を編集することができます(以下のスクレーンショナを参照)。

ユーザー定義 XPath 関数

328 追加機能

🛂 XPath 関数群		×
関数名:	Stock	
戻り値の型(オプション):]
]
	型(オプション) 出現	
	- u w	
関数のボディー(XPath) \$XML/Trades/Stoc	条件式) K[@name=\$XML/Trades/Selection/Stock]	
	XPath	
	OK ++>>セル	

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ごかます。

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 名前空間にて作成されます。この名前空 間はまsps: とうプレフィックスが与えられており、ユーザー定義 XPath 関数を呼び出すはこの名前空間プレフィックスを使用する必要が あます(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 関数エト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(以下のスクレーンショナを参照)。

スキーマツリー ×
+E - 🔞 🗙 🐖 🚰
🭓 Y-X
田田 名前空間
👘 🖏 グローバルテンプレート 🚽
┃ ── T 全てのグローバル要素
田⁄⁄⁄ 全てのグローバル型
電デザインフラグメント →
□□ 🔏 XPath 関数群 🛨
🚽 🖌 🛃 sps:Stock (Selection,Stock Info)
f _{il} sps:StockName0
[L]
スキーマツリー デザインツリー

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 らユーザー定義 XPath 関数を、自動計算内部にある以下の様な XPath 条件式によい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「評価されます:

- 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ーまけまデザインソ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バーにあるアイテムの削除アイエンをクリックします。まけよXPath 関数を右っクリックして、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)の両方が必要にないます。

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リストから型を選択することにとりオブションとして指定することもできます。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た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か変換される訳ではおよい点に注意し てくたさい、型のE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 はありません。 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ゴンルます。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。

以下のスクレージョナに示される 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 関数からよ 3つの入力、 デ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、 値か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。

🔕 XPath 関数群			×
関数名:	Stock		
戻り値の型(オプション):			•
- パラメーター 「「「」」			
- <u> 2 2 </u> 2前	型(オプション)	出現	
一関数のボディー(XPath 図ングル (アー・) (2)	条件式) 		
\$XML/Trades/Stoc	kų@name=\$XML/Trades/Selec	tion/Stock]	
I			
			XPath
		ОК	キャンセル

関数の定義

関数の定義を行うコよ、スキーマソーまけまデザインソノーにあるXPath 関数アイテムの アイコンをクトックします(上のスクノーシショナを参照)。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 関数をダブルクトックしてくたさい。関数の名前を入力して、その定義を関数のボディーペイノに入力します。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、ノ<u>ラメーターどン</u>ケケンスと<u>ノラメーターとノード</u>のセグションを参照くたさい)。関数の戻い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うことです。XPath 条件式によりノードの位置を特定する必要がある場合、以下の方法のどれかによりエンテキストノードを得ることができます。

- 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、Style Vision 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、パラメーターとノード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た。以下のスクノーシンコントにて示される sps:Stock 関数により、/Trades/Selection/Stock 要素のエレテンソニマッチする name 属性の値を持った、/Trades/Stock 要素だけが 返されることしています。

332 追加機能

🛂 XPath 関数群			×
関数名:	Stock		
戻り値の型(オプション):			•
パラメーター			
	型(オプション)	出現	
_ 関数のボディー(×Path	条件式)		
\$XML/Trades/Stoc	≿k[@name=\$XML/Trades/Se	lection/Stock]	
			VPath
		OK	キャンセル

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 様にモジューリルとすることができます:

- · \$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 ことが可能してなり、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 関数はは任意の数の ラメーターを割り当てることができます。関数の ラメーターは XPath 関数ダイアログの ラメー ターペイノこて定義されます(以下のスクノーンショナを参照)。これら 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 関数は以下のようこ動作します:

- 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。

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 ーンションを参照くたさい)。

🛂 XPath 関数群			×
関数名:	Stock		
戻り値の型(オプション):			<u> </u>
パラメーター			
見冒			×
名前	型(オプション)	出現	
ContextStock		10	
StockInfo		10	
ー 回避のポディー(XPath	1-24世)		
ScontextStock	ame=\$Selection]/\$Stockinf	'n	
	······	-	
			XPath
		OK	キャンセル

上のスクレーシンコナトに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 条件式により呼びたされた場合を考えてみましょう:

sps:Stock(\$XML, Node1, Node2)

これら3つの引数(\$XML、Node1、Node2)は、関数のラメーターにて定義された順序に従い、それぞれ \$contextstock、\$Selection、\$StockInfo へ渡されます。

関数呼び出しにて記述された各引数はコンマにより分離されていることに注目してくたさい。これにより、コ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、ラメーターへ渡されることしています。 ノラメーターペイン内にある、ラメーターの順序は、ノ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

オプションとして、ユーザー定義関数における・ラ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、ラ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コはエラーが返されます。この機能により、(関数呼び出しの引 数から得られる)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 関数の各ノラメーターは シーケンスとして考えることができます。ノラメーターの出現プロノティにより、関数呼び出しの引数からそのノラメーター(されしてどれだけの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、ラメーター/引数の分離に使用されているのか、ませよ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

ここでパラメーターとシーケンスの関係と、パラメーターとシーケンスがXPath条件式でどのように使用されるかを理解しておくくきでしょう。以下のようここ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('style Vision Examples', 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は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らごと口留意してくたさい。

ノマ条件式は単一のラメーターおけま単一の引数として読み取られるため、//Person/@salaryと、ケナノマ(ロケーター)条件式の周りでかってを使用する必要は必ずしもみません。ノマ条件式は単一の、ラメーター、引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ります。

336 追加機能

🔕 XPath 関数群			×
関数名:	ThreeAverage		
戻り値の型(オプション):	xs:decimal		•
パラメーター			
			×
名前	型(オプション)	出現	
a	xs:integer	T 10	•
Ь	xs:integer	10	
с	xs:integer	1 0	
□関数のボディー(XPath)	条件式)		
avg((\$a,\$b,\$c))			
			1 1
			XPath
		ОК	キャンセル

上のスクノーンショナでは、3つの、ラメーターが、ハラメーターペイノこで、定義されており、これらの、ラメーターを関数ボディーペインで使用する ことしより、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ハラメーターは、それそれ単一のシーケンスとして考えることができます。出現プロハティでは、それそれのシーケンスにて許されることしてなるアイテムの数が指定されます。上のスクレーシンマルに示される定義では、各ハラメーターがシングルトンシーケンス (つまリアイテムが1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1つ含まれるシーケンスが使用されることしています。型プロハティにより、シーケンスのアイテムに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 デメーターにより与えられ、それらの 平均が求められます。これらの XPath / デメーターによりシーケンスが構成されるため、これらの デ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かあります。ランタイムにて与えられる(3つの デ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 シーケンスでむ、場合、エラーか返されます。

上のスクレーンショナに示されるXPath 関数のThreeAverage()を呼び出すためのXPath ノウメーターの例を以下に示します。デザイン ビュートこ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8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、デメータに対応する、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ために重複しているしかしなが5重複はエラーではよい)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。

ノラメーターの出現プロノティー最低1つがセナされた場合、1つまた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)。

🛂 XPath 関数群			×
関数名:	Average		
戻り値の型(オプション):	xs:decimal		<u> </u>
目目			×
名前	型(オブション)	出現	
a	xs:integer	💌 最低1つ	▼
Ь	xs:integer	10	
с	xs:integer	10	
ー関数のボディー(XPath	条件式)		
arg((count(\$a),\$b	\$c))		
0.6((000m ((40), 40)	•••//		
			XPath
		OK	キャンセル
			11.

上にある定義では、最初の、ラメーターマ対して1つ以上のアイテムが含まれるシーケンスが定義され、残り2つの、ラメーターマ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、ラメーターマ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、ラメーターマ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たさい、

- 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。

上のスクノーシショナーにて示される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)。

その他の留意するへき点

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の関数セナト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 という引数か求める階乗の数がとすると、関数を構成する XPath 条件式は以下のようことがます:

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、デ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。

以下のスクノージョナに示される、3つの・ラメーターとともに定義された Stock() と、ウユーザー定義 XPath 関数を考えてみましょう

🛂 XPath 関数群				X
関数名:	Stock			
戻り値の型(オプション):				▼
パラメーター				
				×
名前	型(オブ)	/ヨン)	出現	
ContextStock			10	▼
Selection			10	T
StockInfo			10	•
「関数のボディー(XPath \$ContextStock[@n	∩条件式) name=\$Selection]/\$S	tockin fo		
				XPath
		[OK	キャンセル

関数ボディーニおお定義は\$ContextStock[@name=\$Selection]/\$StockInfo どこっており、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ことゴンます:

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ントの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み」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ノラメーターに対して、それぞれ3つの別数値が与えられていることに注目してください。

- · \$ContextStock = \$XML/Trades/Stock (/Trades/Stock 要素)
- · \$Selection = \$XML/Trades/Selection/Stock (/Trades/Selection/Stock 要素)
- \$StockInfo = @name

関数定義におたる XPath 条件式は以下のようてなります:

\$ContextStock[@name=\$Selection]/\$StockInfo

引数が渡されることで、関数定義におっるXPath 条件式は以下のようことります:

\$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、ノードセトであるとら点に留意してくたさい。

これにより、XPathノ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ノードか特定、返されること」なります。

9.4 日付の使用

ソースドキュメン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スキーマイスあるxs:date やxs:dateTime データ型を 使用することで、XPath 2.0/3.0 (<u>下の例を参照</u>を参照)。StyleVision では、xs:date 封はxs:dateTime データ型に対して以下のような機能がサポートされます。 入力フォーマント 機能により日付フォーマントの範囲を広げます。

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 日付のフォ <u>ーマル</u>を保証することができます。

XPath 2.0 における日付の計算

自動計算におけるXPath 2.0条件式では、日付形式のデータ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() Due Time 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。

自動計算におっる XPath 2.0 条件式の別を以下に示します。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 という日付型によに定義されており、その要素のエレテンパ 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ります。

- □ 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 ノードの値はな 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り、場合によって 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トを定義することができます。

- xs:date
- xs:dateTime
- xs:duration
- xs:gYear
- xs:gYearMonth
- xs:gMonth
- xs:gMonthDay
- xs:gDay

値フォーマトを使用して日付ノードのフォーマトを行う

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)。

値フォーマット	×
ー型 'date'のフォーマット	
 ● XMLスキーマ値としてフォーマット 	
 ×MLスキーマ値のオブション ×MLインスタンス内のデータタイプ: date 出力ドキュメント内のフォーマット: DD Month YYYY < フィールドの挿入(!) > フィールドオブション(E) エラーへ移動(E) ✓ 同じフォーマットを XSLT 出力へも適用(フォーマットオブション全ては適用できません) 	•
OK (<u>0</u>)	жи(C)

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 値フォーマト構文 のセクションを参 照付选い。

値フォーマナによる自動計算のフォーマナ

日付フォーマト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トを行うために値フォーマトを使用することができます。以下の操作によりこ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。
- 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 かたのしてください。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)。

値	ৗ৴৵৾৾৾৾৾৾৵৴৾৾৾৾৾৾	×
	ーコンテンジフィール ドのフォーマット	
	○ 書式なし	
	※ XML スキーマ値としてフォーマット	
	○ インライン XBRL 値としてフォーマット	
	XML スキーマ値のオブション	
	XML-心スタンス内のデータタイプ: integer (or related)	
	出力ドキュメント内のフォーマット: +###0 マ ##0	
	○000 < フィールドの挿入(L) > フィールドマ(##0) フィールドマ(##0)	
	■ 同じフォーマットを XSLT 出力へも適用(フォーマットオブション全ては適用できません)	
	OK (<u>D</u>) キャンセル(<u>C</u>)	

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。
 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(たえい)。

値フォーマトを出力へ適用する

定義された値フォーマナは 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ナ定義を(全てでは無い ですか) 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か利用できない場合、Authentic Viewの出力に対してのみ値フォーマナが適用されます。

- 三 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(デザイビューのダブとして用意されている) JavaScript エディターこて定義 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 とかできます。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たい。

GUI では、SPS にて定義された全てのJavaScript 関数がデザインソレーウイドウのスクレプトエトリー以下に表示されます(以下のスクレーシンコナを参照)。以下にあるスクレーシンコナではAverate、ImageOut、ImageOver、そして Buttons とう4つの JavaScript 関数が表示されており、SPS にて JavaScript が有効」なってます。



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。デザインドキュメト内にあるコンポー ネントを選択することで、プロ・ティサイド・バーのイベトプロ・ティグループにあるイベト・ハンドラープロ・ティカら、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・ハンドラーニズ扎て JavaScript 関数を割り当てる方法については、関数のイベント・ハンドラ 一への割り当てセクショントコに記述されます。

モジュール SPS 内のスクレプト

<u>SPS モジュールが別のSPS モジュールに追加される</u>と、追加されたモジュール内にあるスクレプトが参照元 SPS でも利用できるような り、参照元 SPS のエンポーネント に対してプロ・ティサイド バーカらイベント ハンドラーとして使用することができるよう エジュール 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たさい。

- 三 関連項目
- デザインビュー
- <u>デザインソト</u>
- <u>モジュールSPS</u>

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)。



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。

- 三 関連項目
- デザイビュー
- イベトハンドラーへ割り当て

関数をイベントハンドラーとして割り当てる 9.5.2

StyleVision GUI では SPS エノポーネトイがHTML にてレンダレグされた際に発生するイベ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。グローノ VI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onClick のような)イベトーズもしてイベト・ハンドラーをセナすることができます。プロ・ティサイドバーでは、JavaScript エディ ターにて定義されたグローノシル関数を、イベトプロノティグループにある各イベトのドロップダウンボックスからイベトノンドラーとして利用する ことかできます(以下のスクノーンショナを参照)。

プロパティ		×
la 🕂 📲		
プロパティ:	白 イベント	_
段落	onclick Average 0	
	ondbiclick Average0	<u> </u>
	onkeydown Buttons0	
	onkeypress ImageOver0	
	onkeyup	7
	onmousedown	•
	onmousemove	•
	onmouseout	
	onmouseover	_
	onmouseup	•

以下の操作により、関数をイベトハンドラーに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レポーネトのイベトが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

- <u>デザイバュー</u>
- JavaScript 関数の定義

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 ファイルへアクセスすることができます:

- ユーザー定義要素おゴはユーザー定義 XML ブロックを作成する これらのデザインオブシェクト はお外部 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リプトを追加する</u> このようして作成されたスクリプトは、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>とユーザー定義 XML ブロックを使用することで、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 ズムを使用することで、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。

ユーザー定義要素の編集
要素の名前と、二重引用符(ダブルウォート)で囲まれた属性の繰り返しを入力してください。
例えば、HTML にて50%の幅を持った「水平線」要素を入力するには:
hr width="50%"
動的属性の値に対しては、括弧で囲まれた XPath 条件式を入力します:
hr width=''{concat(width-in-pct, '%')}''
script src="hie:///C:/Users/test.js" type="text/javascript"
<script src="file:///C:/Users/test.js" type="text/javascript"></script>
OK ++>>tz/

- 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。

<u>ユーザー定義 XML ブロック</u>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) <u>ユーザー定義 XML ブロック</u> の代わりに <u>ユーザー定義要素</u> 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 要素に外部スクレプトを挿入することができます。JavaScript エディター にて作成された男数定義の外側に、以下にあるスクレプトフラグメントを入力してくたさい。

```
var script = ドセンナ.createElement('script');
script.type = 'text/javascript';
script.src = 'file:///c:/Users/Desktop/test.js';
var head = ドセンナ.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
- <u>ユーザー定義要素</u>
- <u>ユーザー定義 XML テキストブロック</u>
- イベトハンドラーへ割り当て

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 ドキュメント。
- 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 した XML ファイルの作成を行うことができます:

- 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 マークアップタグともにデザインビューにて表示されます。ユーザールートとう名 前のドキュメト 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イド ウェ表示されます。このスキーマが、SPS がド 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 ドキュメト のエンテンソとマークアップは、静的コン テンソとして SPS に含まれています。
- <u>スキーマとSPS デザインの作成</u>: (i) HTML ドキュメトから(スキーマソノーウインドウの)スキーマソノー上の適切な場所につよポ ーネントをドラッグするか、オブションとして(ii) 独自のノ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。 デザインウイン ドウでは、スキーマソノーにてノードを構築するために使用された HTML コンテンンが、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 コンテンソは、スキーマノード 無して表示されます。
- 3. デザイボキュメントにて、必要に応じてノードのフォーマナヤ処理ルールの調整、静的コンテンン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. スキーマソノーと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ノーマ対応する XML スキーマ
- SPS デザイン内にあるエンテンソプレースホルダとともに作成されたスキーマのエンテンソと、スキーマをベースはた構造を伴う XML データ ファイル
- SPS (.sps ファイル)ませまデザインをベースしたXSLT 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ドキュメントが新たな 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。

新規 SPS ドキュメトが作成されます。デザインビューゴよ StyleVision にて使用することのできる定義済み HTML フォーマナのマーク アップとともJ ジキュメントが表示されます(以下のスクノーンショナを参照)。

A Sample Article
P 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.
PIt 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ドキュメナがメインテンプレート内に表示されていることに注目してください。グロー・ジレ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ろ相前のドキュメント要素とともに、ユーザー定義スキーマか作成されます(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パ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 ドキュメト のドキュメトノードへ 適用されます。スキーマソレーウィド ウェ て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 ドキュメト は作成されます。 この時点でデザイビューはエメインテンプレート内部のHTML ドキュメト コンポーネント か静的コンポーネント とて SPS 内に表示さ れます。これら HTML コンポーネントは、Authentic View にて読み取り専用の非 XML コンテンソコンポーネント とて処理されます。 XSLT スタイルシートには、これらのHTML コンポーネントがテラル結果要素とて含まれます。この時点のスキーマイコよドキュメント要素のユーザールートしか含まれておらず、一時的な内部 XMLドキュメント にお キュメント 要素のユーザールート だけ が含まれます。

以下にある、ずれかの方法により、ユーザー定義スキーマーンで選択した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。
- 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市ルダにより、XMLドキュメントから得られたデータが、出力にて)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。

三 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なで 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へ変換するか、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

このセクションでは、選択対象に対して行われる一般的な処理について記述します。テーブルならびにコストの特別な場合については要素/ 属性としてテーブルやリストを作成のセクションを参照くたさい。

選択を行うコよ HTML ドキュントコンポーネントをクリックするかテキスト文字列をハイライトします。複数のコンポーネントを選択するコよ、 ドラッグにより複数のコンポーネントをハイライトします。StyleVision にて複数のコンポーネントが イライトされた場合、それらコンポーネント 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たさい。

スキーマットーこおける位置

目的のスキーマソノーノードへ選択対象をデラッグすると、マウスカーノルが以下にあるとちらかのシンボルに変わます。

- 新たな兄弟を作成シンボル↓にてドロップすると、ドロップされたノードの兄弟ノードとして選択対象が作成されます。
- 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、そのノードの兄弟おとは子ノードとして選択対象を作成するか決定することしています。

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カ表示され、(i)作成されるノードは要素になるのか属性になるのか、(ii)選択対象をノードに変換するか、ノードの周りを選択対象で囲むのか、を選択することしています(以下のスクノーシンョットを参照)。



以下の点に留意してくたさい

- 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、要素/属性としてテーブルやリストを作成のセクションを参照くたさい。

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。ノードを右クトックして、コンテキストメニューから目的のコマンドを選択してください。スキーマソノーサイドノ・チークノールレーアイコンからも、コマンド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おコよソールレー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、既にあるいなか下正となってしまう場合もある点に注意してください。

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。

- 三 関連項目
- <u>ユーザ-定義スキーマ</u>

9.6.3 要素/属性としてテーブルやリストを作成

HTMLドキュメト内にあるテーブルならびビリストをXMLスキーマの要素おけば属性へ変換することで、テーブルまけはリ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おける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。

Convert selection to elements Convert selection to attributes Surround selection with element Convert selected table/list to elements Convert selected table/list to attributes

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る動的テーブルなのかを選択します(以下のスクノーンショットを参照)。



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. オプランを選択した後には変換をクルクして完了します。

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。
- 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おける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。

Convert selection to <u>e</u> lements
Convert selection to <u>a</u> ttributes
Surround selection with element
Convert selected table/list to elements
Convert selected table/list to attributes

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か静的リストお台類的リストになるかを選択します(以下のスクリーンショットを参照)。



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. オプノンを選択した後には、変換をクトックして完了します。

□ 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」コマンド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 レポートをSPS には作成する場合、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 まけは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。ウェブサーィー上のHTML ファイルを更新するけは、(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 ファイルをサーィ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け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 アプリケーションアイルロ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 ファイルませはXML ファイル そして XSLT-for HTML ともにサーバー上 「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) 変換の出力がウェブイン 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 ませは 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 おけは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。

- 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
- 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みます。変換を行う際に 問題が発生する場合、Altova XML のセオアップ方オフか記述されている 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に以下のファイルが含まれます:

- Readme.doc
- SPSFilename.aspx
- SPSFilename.xslt
- SPSFilename.cs

SPSFilename.aspx が出力ドキュメナのURL どむます。SPSFilename.aspx により、SPSFilename.cs. に含まれている C# コードか実 行され、この C# コードにより(ファイル、おけよ DB から取得された) XML コンテンンが読み取られ、SPSFilename.xslt ファイルとされに 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ノロ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 コンテンソの変換が行われます。変換によりHTML ドキュメトが作成され、 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 のコンテンパ変更されると、 SPSFilename.aspx の表示(おゴム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ングが行われます。

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、ローカルオストにASPX アプリケーションをセオアップします。詳細については ASPX アプリケーション フォルダームある Readme.doc ファイルを参照くたさい。このフォルダーならパーファイルは SPS ファイルがアウ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お生成されます。

RaptorXML のインストール

最新バージョンのAltova XML がインストールされていることを確認してくたさい。Altova XML 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、Style Vision に搭載された機能の、くつかは、Altova XML の、ージョンイントプロナポートされたいという点に注意してくたさい。

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たさい。ステップ5 により、IIS が有効になっているか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:

- 1. 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」ています:

- 1. Windows のスタートポシカジョントロールパネル | 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。
- 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 ファイル名がく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か再度生成されます。

灹 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。

9.8 PXF ファイル: SPS とその関連ファイルのコンテナー

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 フォーマトは SPS デザインを(スキーマア・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 を出力フォーマト へ変換する XSLT ファイルといす。) 関連するファイルとともし ッケージ化するために、Altova には開発されました。PXF ファイルフォーマトを使用する ことにより、 Authentic View の編集や、 Authentic View から出力を生成するために必要な全てのファイルを、1つのファイルとして配布す ることが可能にないます。

このセクションではPXF ファイルの使用方法について、2つのパトに分けて説明します:

- <u>PXF ファイルの作成</u>
- <u>PXF ファイルの編集</u>
- <u>PXF 7ะ4เมารีวีน4</u>

灹

PXF 機能に関する以下の点に注意してくたさい

- XSLT 2.0 おは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

9.8.1 PXF ファイルの作成

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、メニューオプションから「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(以下のスクレーシンコナを参照)。



SPS フォーマットは Style Vision デザイン 1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 説明を行うため SPS フォーマナ についてば説明しません。 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 細を参照くたさい。

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 <u>ンを保存 コマンド</u>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(以下のスクレーシンヨットを参照)。
table XML Form (リコンテンツを構成	
"ザイン時ファイル その他のファイル	
《スならびにファイル・名が StyleVision により認識されているイメージ ファイルなどのファ こ埋め込むことができます。 PXF ファイルに埋め込むファイルを選択してください。	ゥイルは、自動的に PXF ファイル
グローバル構成	
SPS ファイルの埋込み	
イメージの埋込み	✓
CSS ファイルの埋込み	
XSLTファイルの埋込み	✓
SPS モジュールの理込み Authority カフタノ ポかいた畑かジュ	✓
Authentic カスタム ホタンを注意のため スキーマソース \$XML (メイン)	V
スキーマ ファイルの埋込み	V
作業用 XML ファイルの埋込み	
一時 XML ファイルの埋込み	
XSLT ファイルを生成し、XMLSpy や Authentic Desktop にて以下を作成 ダ HTML(H) ダ RTF(R) ダ PDF(P) ダ Word 2007+(W)	
- ファイルを保存した後は、デザイン概要からこのダイアログにアクセスして、オプション	の変更を行うことができます。 OK キャンセル

デザイン時ファイルタブのグローンN構成セクションでは、デザインは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 が生成する出力フォーマナを選択してくたさい。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、出力フォーマナを生成と閲覧を行うためのソールドーボタンがAuthentic View にて有効になりま す(以下のスクレーンショナを参照)。

メモ StyleVision の高し、デジョンでのみサポートされている(高から低い順: Enterprise、Professional、Basic)出力のすめのXSLT ファイルは、PXF ファイル内で作成され、PXF ファイルが低い、デジョンで開かれる場合、PXF ファイルを保存すると、低い、デジョンでサポートされる出力のすめのXSLT ファイルは、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、まけよ、保存を中断して、サポートされないフォーマナを保管します。

その他のファイルタブでは、デザインココ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(以下のスクレーンショントを参照)。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。以下のスクレーシンコナでは、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。

🕑 Portable XML Form (PXF) の構成 🛛 💽
Portable XML Form のコンテンツを構成
デザイン時ファイル その他のファイル
XPath 条件式内で定義されているイメージ ファイルやその他のファイル名は、デザイン時において StyleVision にとって 未知のものであり、自動的には PXF ファイルに埋めこまれません。これらのファイルを PXF ファイルに埋め込むには、以 下から行ってください。ファイルの XPath 条件式から得られるファイルパスは、相対パスでなければならず、以下にある 様なフォルダー構造で表せられなければなりません。
PXF 付随ファイル
ドキュメントを追加 アーカイブから削除
ロ 📾 PXF ファイル Ali 名前 サイズ Paws Items https://www.items.htmp
-
4 b
DVE コーズルを/見方した/2011 ごぜ / 「畑市わたこのガノフロガーフタレコレア」ナペシー、の本市を生きていがったナナ
FAF ファイアルを1本1+した1友は、テリイン194天からとのダイアログにアクビスして、オフンヨンの変更を11プレビがしでます。
OK キャンセル

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オダンをクトック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 ファイルが配置されているフォルダーか開かれます。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。上のスクレーシ ショントからは、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NewsItems.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、その他のファイルに対する参照を相対/次で、更に付属ファイルのペイン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、上のスクノーンショントにある NewsItems.bmp には、Images/NewsItems.bmpという相対/次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種類の方法があります。

- 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 StyleVision 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コよ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(以下のスクノーシンコオを参照)。

デザイン概要 ×		
∇ Portable XML Form		
埋め込まれたファイルの構成		
埋め込まれたファイルの更新/展開		
▷ ソース		
▶ モジュール		
▶ CSSファイル		
▶ パラメーター		
▶ XSLT ファイル		

埋め込まれたファイルの構成

デザイン概要サイドバーにあ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 か表示されます。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

デザイン概要サイドバーにある埋め込まれたファイルの更新リンクをクリックすると、Portable XML Form (PXF)の更新ダイアログ か表示されます(以下のスクレーンションを参照)。

🚯 StyleVision Portable XML Form (PXF) 更新	\mathbf{X}
このダイアログでは更新されたソースファイルを PXF へ統合することができます	
・ 現住のPAFかめる団直から、作取当りなシースファイルを快楽	
○ オリジナルの位置にあるソースファイルを検索	
注意:現在の PXF をバックアップするよう推奨します。更新を元に戻すことはできません。	
Ka. Kute	
クローハル構成 SPS ファイルの更新	
イメージの更新	
CSS ファイルの更新	
インボートされた XSLT ファイルの更新	
SPS モジュールの更新	
Authentic カスタムボタンの更新	
スキーマソース \$XML (メイン)	
人名エマファイルの更新	
TF美田 XML ファイルの更新 テンプレート VML ファイルの更新	
「JJJVFFAME JY11000更新	
- 王U展開… 単新 キャンセ	<i>.</i> , ,
	111

ソースファイルの取得を、現在 PXF ファイル的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、おけよオト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 こチェックを入れて、 更新 ボタンをクト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、 オトジナルの 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 ファイルゴは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ため、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<u>Altova 製品のAuthentic View</u> にて開く事ができます。XMLSpy にて PXF をどのように使用することができるかの使用シナリオを以下に示します::

- 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。

- メモ 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.
- 三 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 により生成された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になります。このセクションでは自動化機能について説明します。

StyleVision のファイル生成機能

StyleVision によりSPS デザインか作成されると、使用している StyleVision のGUI から XSLT ならひに出力ファイルを生成すること ができるようてなります。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 出力するためのXSLT ファイル。
- HTML 出力

上のリ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 を伴わな ワアイルの生成

XSLT スタイルシートと目的の出力ファイルを生成するコよ、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 を開かずにStyle Visionのファイル生成機能を呼び出す Style Vision Server は、多種の出力を作成することができます。
- <u>RaptorXML</u>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ドアロンの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 ドキュメトを処理し書式を出 カするすめ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にお使用されることがで きます。(RaptorXMLは、SPSを入力・ウメーターとして受け取らえい場合が防ます) 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 インターフェイスに加え、 RaptorXMLは、COM、Java、と、NET のすめのインターフェイスを提供し、このすめ、これらの環境内からの呼び出しが簡単 しないました。変換のすめの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 StyleVision Server を使用して、複数の変換を事前に設定された トリガーゴ従し(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:

- <u>Style Vision 実行可能ファイル</u>を呼び出す。これにより Style Vision 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。

コマボラインの使用方法

コマドラインを使用するコン2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 おこ は<u>Style Vision Server</u>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。
- 複数のコマボをバッチファイルに入力し、以下の例にあるようない シチ処理を行います。

```
@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 のスケジュール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に従うかたちで Style Vision コマンド 実行することができます。 す。コマンドを直接入力、おけは実行するタスクとしてバッ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」なります

StyleVision [<SPS File>] [<options>]

where

StyleVision	StyleVision アブリケーションフォルダーン存在するStyleVision を呼び出します。
<sps file=""></sps>	SPS ファイルを指定する
<options></options>	下にリスト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StyleVision 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 が正確にコードされるには、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

オプシンの入力の順序に決まりはかほせん。FO、RTF、PDF、Word 2007+ 出力に関連するオプションは、Enterprise エディション、おけは、Enterprise とProfessional エディションのみで使用することができます。これらのオプションは、下の Enterprise エディション、おけは、Enterprise とProfessional エディションリスト内で説明されています。

• XSLT ファイル出力

-OutXSLT= <file></file>	XSLT-for-HTMLを指定されたファイルに書き込む
-OutXSLRTF= <file></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></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 ンのみ)
- OutXSLWord2007= <file></file>	XSLT-for-W ord 2007+ を指定されナファイルに書き込む (Enterprise エディションのみ)

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

- パンは絶対的、ませよ、相対的でバックスラッシュを使用します。
- ファイル名、おさよ、パマスペースが含まれる場合、パマ全体はお用符で囲まれる必要があります。例: "c:\My Files\MyXML.xml" or "c:\MyFiles\My XML.xml".
- コマイ、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 StyleVision Server

StyleVision Server を、Windows、Linux とMac OS システム上で、コマドライン イクターフェイス(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。 変換のナメのXSLT スタイパシートは、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 のこし ではよく、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 ファイルを入力とてとます)。StyleVision Server はAltova FlowForce ワークフローの一部とて使用されることが奨励されます。FlowForce ワークフローは、変換ショブを設定済みのトリガーご従い 開始します:FlowForce ジョブがトリガーされるとやホワーク内で、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グシンを参照してくたさい。

StyleVision Server CLI に関する詳しい説明は、次を参照してたさい、StyleVision Server ドキュメナ。

出力ファイル

StyleVision Server は、1つませは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:

• PXF内で指定されたXMLとXSLT-for-HTMLファイル おけよ 代替のXMLファイルを使用したHTML(.html)ファイル

10.2 RaptorXML の使用

Altova RaptorXMLは、最新の標準と並列計算の標準のすめの最適化のすめご構築された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 データを処理するす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検証をサポトする RaptorXML+XBRL Server エディション。

RaptorXML に関する詳しい説明は、次を参照してくたさい<u>Altova Web サイト</u>。

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た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」コマイドを用いて、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てという言してください。
- 2. RaptorXMLを使用して、XMLドキュメントを変換するために生成された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 には、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沿して立ちます。
- 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、処理の自動化やスケジュール処理などを行うことができます。

XML ソースドキュメントからF出力ドキュメント(HTML などの生成を自動化するためにRaptorXML を使用する方法に関しては次の セグショ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 に関する詳細は、次を参照してくたさい、 RaptorXML ユーザードキュメント。

10.2.1 PDF 出力

XMLドキュメイからPDF出力を生成するコンコンのステップを踏む必要があります::

- 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。
- (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」コマドによりファイルとして保存することができます。

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 を参照くたい。

FOPとXSLT

FOP オプションにより、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 ドキュメント 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 内蔵のエンジノは、Style Vision とRaptorXML によりサポートされるすべてのXSLT 機能をサポートしな、可能性 かあります。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、エラーカ発生する可能性があります。RaptorXML+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 の生成を行うようこしてくたさい。

PDF の シチ処理

XSLT 変換の最初のステップのためにRaptorXML を使用して、PDF を生成する素早し簡単な方法です。 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、シ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番目のコマイでは、 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、次を参照してくだい、処理の自動化。

10.3 FlowForce Server を使用した自動化

Windows、Linux、とMac OS システム上で使用することのできるAltova のFlowForce Server を使用して、ネルワーク上で変換を自動化することができます。このプロセスは以下のように作動します:

- 1. (ファイル Flow Force ヘデプロイするコマイを使用して) Style Vision から PXF ファイルは.transformation ファイルとて Flow Force Server にデプロイ されます。.transformation ファイルは SPS (内でデザインされたよお) に変換を実行するために必要な全てのファイルと情報を含んでいます。下の図では、デプロイは接続ラインにと表示されていま す。)
- 2. .transformation ファイルがFlowForce Server にデルイされた後、.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イズはい、変換を生成するジョブをFlowForce 内でシ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1ホー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こ) StyleVision、まけよ、HTTP アドレスによリアクセスすることのできるFlow Force ジョブをFlowForce Web 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、次を参照してくけざい、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 プします (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 マルングを実行のためにAltova MapForce Server に転送するFlowForce ジョブも作成することができるため、ワークフロー内でのMapForce Server の役割に表示されています)。



FlowForce ジョブこと呼び出されない場合、StyleVision をコマイトラインから呼び出すことができます。使用方法は、<u>StyleVision</u> Server ドキュメート
で説明されています。

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ドのシーケンスを含んで、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ドは、最初から始まし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。

- 一連のコマドを自動的に実行する場合(下を参照)。
- コマイカ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か実行されることを予定する場合。

コマドのシーケンスを持ついシテファイル

実行されるコマドのシーケンスは以下のようこ入力されます

@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、シチファイルは、StyleVision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

このセクションでは StyleVision ツール・、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。セク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。
- <u>XPath 式 の編集ダイアログ</u>の説明。
- すべての<u>ツールドーとそれらのアイコン</u>、そしでツ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。

ユーザーリファレンスのセグションでは個々のコマンドに関する記述がなされている一方で、関連するセクションでは Style Vision の機能で使用されているメカニズムの詳細にていて説明されます。メカニズムは以下のグループにとV構成されています

- <u>SPS ファイルコンテンソ</u>
- <u>SPS ファイル構造</u>
- <u>SPS ファイルの高度な機能</u>
- <u>SPS ファイル表示方法</u>
- <u>SPS ファイル追加機能</u>

三 照関連項目

- <u>+-+IPII07/1/20+</u>

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か含まれ、各エレポーネントはそれそ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。:

- <u>XMLドキュメナ内のノード</u>
- XMLドキュメント コンテンソ
- データ入力デバス
- 定義済みのフォーマホ
- XPath オブジェクト
- URI オブジェクト

これらコンポーネントに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) <u>編集メニュー</u>のコマド、おけは(ii) 標準的な Windows のショートカナにより切り取り、コピー、貼り付け、おけば削除する
- 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ドキュメント内にある要素や属性はSPS デザインドキュメント内でタグとして表示されます。各ノードはは開始タグと終了タグか管まれます。開始タグませま終了タグをダブルクリックすることでそのノードが縮退され、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レテンが隠されます。縮退されたノードをダブルクリックすることで、ノードが展開され、内部に含まれているエンテンンが表示されます。

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

<u>On1:Name</u> (コンテンツ) On1:Name	On1:Name
----------------------------------	----------

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

href (コンテンツ) = href	= href
---------------------	--------

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たさい、、テンプレートとデザインフラブメント、

XML ドキュメント コンテンソ XML ドキュメント コンテンソは2 種類のプレースホルダームと表示されます:

- (contents)
- (rest-of-contents)

contents パース市りダは単ーノードのエレテンンを表し、ノード内に含まれるすべてのテキスト エレテンンが出力されます。ノードが属性ノ ードおけよ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。ノードが混合エレテンンや要素エレテンンだけで構成される要素ノー ドの場合、子孫ノードのテキスト コレテンンが出力されます。XSLT の用語に置き換えると、contents パレース市り以ば、select 属性の値 によりノードが指定された、sl:apply-templates 要素と等し、振る舞しを行します。

メモ contents プレースヤレ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 プレースヤルダにより、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ベルテンプレート(その要素に対して定義されている場合)、おけは要素のデンオルトテン プレート(テキストのみの要素からはデキストだ」を出力し、子要素のテンプレートを適用)となります。例えば、title、author、isbn、pubdate という子要素を含んだ book という要素を考えてみましい。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) は出力されない ことしています。しかし、book の定義内の title 要素直後に その他のコンテンツ プレースヤルダが含まれている場合、その他の子要素 (author、isbn、pubdate)、グローベルテンプレート(その要 素に対する定義が存在する場合) おけは要素のデンオルトテンプレートも適用されることで、

データ入力デバイス

デザイノニてデータ入力デッイ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 はお以下のデータ入力デッイスを割り当てることができます:

• *入力スール*(単一行と複数行)



ニンボボックス



• *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、縮退されてしるタグも、ダブルクトックにより展開することができます。データ 入力デバイスの細かな使用方法については、データ入力デバイスを参照ください。

定義済みのフォーマト

定義済みフォーマナは薄、紫色のタグで表示され、ダブルクトクによ、展開、縮退することができます。



上のスクノーンショナでは、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に2 つのメカニズムか用意されています。

条件テンプレート

? 1 2 0n1:Address

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 ーからプロ ティサイドバー自動計算の編集をおこなうダイアログへアクセス</u>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をベースコンたオブジェクトには以下の3 種類があります:

- 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。

- 三 関連項目
- ・ツールド
- <u>デザインサイドバー</u>
- 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(アのスクリーンショナ)はXPath 式の編集を作成、テストするオンジニ使用されます。 条件付きの処理 ま たま、スタイルとフロンテ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 のナダのXPath 1.0、XSLT 2.0 のナダのXPath 2.0、XSLT 3.0 のナダのXPath 3.1) の<u>XSLT</u> ハージョン に対応するXPath 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 ゴよ <u>SPS の XSLT バージョン</u>を切 り替えます。

ダイアログのレイアウト

XPath 式の編集にコお以下のペインか含まれています(アのスクレーンショントを参照)(i) 式ペイン(左上)(ii) ソースペイン(右上) (iii) 結果ペイン(ア)。ビルダーモードでは結果ペインは入力ヘルレーペインに比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 式か割り当てられた作業 XML ファイルに対して評価される評価モード。 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。結果内のノードをクリックしてダイアログのノースペイン内のノードに移動することができます。
- 割り当てられた作業 XML ファイルに適用し XPath 式をデバッグすることのできるデバッグモード。ブレークポイントを設定し、評価をステップパイステップで行うことができます。各ステップで変数のコンテンツを確認することができ、評価の追加アスペイトをチェックすることのできるカスタムのファッチ式を設定することができます。で

ウイドウのソール いった角にある「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グタンのサブセグタ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 、 の「ビルダーモード」 ボタンを使用して<u>式ビルダー</u>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ントを参照)



評価モード内でエ/**ジリュエーター** ポタンをクリックします(アのスクリーンショットを参照)。エ/ジリュエーター ゴは次のペインが存在します(アの スクリーンショットを参照):(i) 式ペイン(左上)(ii) ソースペイン(右上)(iii) 結果ペイン(ア)。

Sedit XPath Expression	- 1	□ ×
▶ • ■ २० ०० २० ••• ••	🕎 👷 Builder 📗 Evaluator	
/data/@* 📮 \$)	KML (Data.xml) 💌	
	<> data	
	werify = "true"	
	= last-edited = "2008-05-02"	
	first-entry = "2008-04-10"	
	sample-string = "abc"	
	xsi:noNamespaceSchemaLocation = "Data.xsd"	
		t various
	⊕ <> strings	
	⊕ <> numbers	-
		•
</th <th><!-- = Abc</th--><th></th></th>	= Abc</th <th></th>	
▼▲ □@□ ٩,3,3,X ≂ ⊽		Items: 5
= verify	true	
= last-edited	2008-05-02	
= first-entry	2008-04-10	
= sample-string	abc	
= xsi:noNamespaceSchemaLocation Data.xsd		
Using Java Using .NET	ОК	Cancel

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ント)。

以下の点に注意してくたさい

- XML ファイルに対して式を評価するコよ そのファイルはSPS のノースの一つのファイルと作業 とて割り当てられる必要があります。
- (入力中に評価」iアイコンをソール ←から選択すると),結果は式の入力中に表示されます。おけよ(ツール ←の左上の「評価/デバッグの開始」(F5)をクリックすると表示されます。
- ソーズソー内のノードのXPath ロケーター式を入力するコよノースペイン内のノ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)コメト(iii)属性(iv)要素。ソースペインの下の状況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クトックします。このコンテキ ストノードを式のためコ保存するコよ、(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増えてくたさい、式のた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り、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」パタンは 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ンションを参照してください。

- (よ読みやすくするけかにつまを複数行で表示するにはリターンキーを使用します。
- 式フィールド内のテキストのサイズを大きくする小さくするコよ、式フィールドをクトックし、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、や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、やを挿入する XPath 式内の場所にカーソルをポイトします(ii) ソーズソレー内でターゲオとするノード内にカーソルをポイトします。式内の選択されたノードのロケーター、やを入力します。ロケーター、やはギュメントのレートノードから始まる絶対パマです。

結果ペイン

結果ペイノは下のスクノーシショナ内で表示されています。結果ペイン固有ののソールレーが存在することに注意してください。

Edit XPath Expression	-		×
▶ - ■ २→ <> >>	🕎 👷 Builder 📗 Evaluator		
/data/@* 📮 SX	(ML (Data.xml) 💌		
	<> data		
	····· = verify = "true"		
	= last-edited = "2008-05-02"		
	first-entry = "2008-04-10"		
	sample-string = "abc"		
	ssi:noNamespaceSchemaLocation = "Data	.xsd"	
		nlight vario	ous
	∃ <> numbers		-
			F
0	der = abc	E	1 6.
		.e	9 50
▼▲□◙□٩۶३Х≍⊽		lt	ems: 5
= verify	true		
= last-edited	2008-05-02		
= first-entry	2008-04-10		
= sample-string	abc		
= xsi:noNamespaceSchemaLocation	Data.xsd		-
Using Java Using .NET	ОК	Can	cel

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つの列には構成されています:(i)ノード名、おけはデータ型(ii)ノードのコンテンソ。
- XPath 式が要素おけ属性などのノードを返す場合ノードのエレテンソ全体がシードの値とて表示されるかを選択することができます。これを行うには、完全な結果を表示するたオノニゴル替えてください。
- 式により生成されるノテラルとは異なり、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。
- 結果サブラインの列、おけよ、値列の両方をエピーすることができます。全ての列をコピーするはよ、サブラインを右クトックし「全ての 列を含みコピーする」をオノご切り替えます。(おけよ、結果ペインのソールドー内のアイコンを使用してコマンドをオンとオフに切り替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、(そのサブラインのナックは)「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ナックは)を選択します。

<u>結果ペインのソールバー</u>

結果パインのソールバーコはナビゲート、検索、およびコピー機能を提供するアイコンか含まれています。これらのアイコンは、左から順番に以下て説明されています。対応するコマンドは結果リストアイテムのコ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	クトップボードに選択された結果アイテムの植力ラムをコピーします。全てのカラムをコピーするコは、全てのカ ラムを含みコピーするコマンドに切り替えてくたさい(下を参照)。
<i>全てのメッセージをクリップボ</i> ード1 <i>二</i> 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。
<i>全てのカラムを含みコ</i> ピーす る	(i)全てのカラム、おけは(ii)値カラムのコ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。
クア	結果のリストをクリアする
複数のラインの結果を単一 のラインに折りたたむ	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、(新しいラインませは文字を含む場合)、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ソ全体を表示します。

11.2.2 デバッガ—

デバッガーによりXMLファイルと作業のコンテキスト内でXPath 式をデバッグすることができます。「評価/デバッグの開始」ドロップダウンメ ニュー内で「デバッグの開始」にアクセスします(アのスグリーンショット)。これによりデバッグモードに設定されます。式を作成するオメのヘル プのオメののビルダーと(式をデバッグするオメの)エバリュエーター間を切り替えることができます。デバッグを開始するオメリニ「評価/デバッグの 開始」(F5)をクトックします。



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)をクトックしてデバッグを開始することができます。

<u>デバッグモードをセメアップするためのボタン</u>

N	評価/デバッグの開始(F5)	デバッガーを開始します。
----------	----------------	--------------

*	ビルダーに切り替える	式の作成を手助けするコンテキストを考慮する入力ヘルレーを提供する式ビル ダーモードに切り替えます。
	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

デッジグモード内では、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):

- ペイン内に個別のダ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ン表示される変数とウォッチ式ペイン

Image: Section of the section of t	🔕 XPath 式の編集	— D >	<
<pre>/presswatch/newsitems/newsitem//source /presswatch/newsitems/newsitem/source SXXL (QuickStart.xml) = SXXL (QuickStart.xm</pre>	🐚 🔹 🔳 💫 🔗 💎 📼 📼 🕎 % EU/9- ! I/(U1I-9	-	
▼▲□喧□ ۹ 身 身 X 末 호 アイテム:4 ◇ newsitem ◇ newsitem ◇ newsitem ◇ newsitem	/presswatch/newsitems/newsitem/source	SXML (QuickStart.xml) ▼ □ <> presswatch □ <> selection □ <> newsitems <> newsitem [1] <> newsitem [2] <> newsitem [3] <> newsitem [4] <> team	
	▼▲□ @ □	マムロロヘタタ+- # コールスタック ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	

デバッガーモードには以下の機能が搭載されています

• XPath 式かどのように評価さえるかをステップごとご確認するためにXPath 評価プロセスにステップインすることかできます。ステップイン(F11) ツール レ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か黄色で パライ

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・オ)。例えば、上のスクリーンショ・オでは、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クポイトを設定し、これらの箇所で結果をチェックします。ブレークポイン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) ツールレ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も素早く行うことができます。
- トレースポイ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トレースポイトを設定します。(ブレークポイト以外)評価は 一時停止しませんが、結果ペイン内のリスト内トトレースポイ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ントなどの情報をチェックすることができます。この機能はブレークポイントで役に立ちます。
- 変数とフォ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バッグポイントペインのコールスタックタブ内で表示されます。
- ブレークポイトと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 式のデバッグのための大まかなステップは以下の通りです:

- 1. XPath 式を式ペインに入力します。
- 2. ブレークポイト、おさか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」をクリックすると、式内にブレークポイトがマークされていない限り、評価か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」を繰り返しクリックしてブレークポイ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/オーン・機能を使用して評価をステップルイステップニイトはす。

₽ ®	デバッガーの開始(F5)	デバッガーを開始します。評価はブレークポイントのために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。
\$	ステップア 가 (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・トします。
•	ブレークポイントの挿入/削除(F9)	カーノルはポイトする箇所の式ステップでブレークポイントを挿入/削除します。
	トレースポイントの挿入/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。

<u> デバッグのためのボタン</u>

<u>ステップイン、アウト、およびオーバーの評価ステップ</u>

ステップイン機能にお評価をステップ・イステップで行うことができます。このコマンドの各クトックにお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(下のスクレーンショントを参照)。ステップアウト機能にお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、ステップオーノンー機能は低いレベルのステップにステップオーノントし、同じレベルの次のステップ 「こ移動することができます。ステップ機能を下のスクレーンショント内で表示されている式を使用して、3つのステップボタン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なつの newsitem ノードを表示しています。

🔕 XPath 式の編集	— 🗆	Х
🛯 🍬 🕶 🗏 🐎 🔗 💎 📼 📼 🔛 👷 EU/3- ! I/(UII-5	-	
/presswatch/newsitems/newsitem/source	SXML (QuickStart.xml) SXML (QuickStart.xml) State State	
$\nabla \land \Box \Box \Box \land \land \land \land X = \overline{\nabla}$ $P \land \overline{\tau} L: 4$ \land newsitem \land newsitem \land newsitem \land newsitem	マイ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	Debugpoints 💊 🗆 – Jh 🛛 🕸 🤊 🖉
Java の使用 Net の使用	OK ++>>t)	ŀ

ブレークポイント

ブークポイトは「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すめのポイトです。ブーク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。ブレークポイトを設定するこ は、式内のブレークポイトを設定する箇所にカーンルをポイトし、「ブレークポイントの挿入/削除」(F9)ツールドーボタンをクレックしま す。ブレークポイトは赤い点線のオーバーラインでマークされています。ブレークポイトを削除するコよ、選択し「ブレークポイント/削除」 (F9)をクレックします。

下のデバングポイン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レーで表示されています(アのスクレーンショナを 参照。これにお尻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、下のスクレーシショナ内では、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レ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ノルをポイントし、ツール・・ボタン「トレースポイントの挿入/削除」(Shift+F9)をクリックします。トレースポイントは青い点線のオーバーラインでマークされています(アのスクリーンショントを参照)。トレースポイントの挿入/削除」(F9)をクリックします。



メモ ブレークポイトとトレースポイトか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バーラインが作成されます。

下のデッジゲポイトも参照してください。

変数、ウォッチ式、コールスダック

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。

 XPath 式の編集 			D X
🍬 🕶 📕 🐎 🔗 🔗 📼 📼 🔛 🛞 ビルダー ! エバリュエーター			
<pre>1 for \$i in //newsitem[metainfo/relevance="NanoNull"] 2 return (\$i/headline, \$i/source)</pre>	Since (constant and the second sec	itart.xsd" on 2.0 of NanoPower" I version 2.0 of its marke	t-leading I
▼▲□回回 ۹ 多 S X 末 交 アイテム: 1 ◇ headline NanoNull Inc Launches Version 2.0 of NanoPower	▼▲□@□< ♀ ♀ ♀ + - □ ♥ Si/metainfo/enteredBy/@id="NED" □ ♥ Si □ ♥ Si ↓ newsitem	# コールスタック 2 AxisStep 1 Expr 0 SimpleForExpr	● Debugpoints 🌏 コールスタック 《》
Java の使用 Net の使用		OK	キャンセル

<u>変数</u>

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に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 上のブレークポイントで処理が一時停止されています。 \$i 変数によ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 が最初のノードである現在の値と共に表示されています。

<u>ウォッチ式</u>

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。上のスクレージェット内のサンプルでは、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お上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かどの条件が満たしているかを示しています。
- 特定のエレテキスト内のデータを検索する場合。例えば Company 要素コンテキスト 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ウォッチ式 @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はペインのソール・一内の、上のスクリーンショナ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は、ツール・一内で「選択されたウォッチエントリの削除」をクトックしてくたさい。デバッグ中に式が例えば、変数のこの範囲以外など」何らかの理由で正確に評価されない場合、ウォッチ式にあ色で表示されます。

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 ンショット内の右下のペイン)。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せ呼び出し 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ィッグポイトペインのディッグポイ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トと(青い、丸印で表示される)トレースポイトを表示しています(下のスクリーンショナトのの右下のペイン)。各ダイアログポイントはラインと文字番号と共にコストされています。例えば、AxisStep@2:12 は式ペイン内の式のライン2 と文字 12 にディッグポイトか存在することを意味します。

3 Edit XPath Expression — 🗆	×
n 🖡 🕶 📕 🗞 🐶 🐼 🖙 📼 🔛 👷 Builder 📲 Evaluator	
1 for \$i in //newsitem[metainfo/relevance="NanoNull"]	•
2 return (\$1/neadline, \$1/source) -⊡ <> newsitem [1]	
- C <> newsitem [2]	_
————————————————————————————————————	
→ ⇒ ♦ aateine	
Source manual wine	
→ synopsis share of handhard metaling angle short are day to close at essent at close of the synopsis share of the synopsis short are day to close at essent at at es	
Selevance [1] "Stockmarket"	
→ → c >> enteredBy	
= id = "ABE"	-
<! = Rbc</p	Ð
	—
Ansight and a significant of the addition	
□ Traces □ □ \$i □	
trace(SimpleForExpr) () trace(SimpleForExpr)	
· · · · · · · · · · · · · · · · · · ·	
Using Java Using .NET OK Cancel	

以下の機能に注意してくたさい

- ブレーケポイトのナダニデ・ドグポイントペイン内のブレークポイントの条件をダブルクトックしてブレークの条件を入力し、条件のナメリニ式を入力し 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クポイントは有効化されます。例えば、上のスクレーシショナではブレーク条件 \$i/metainfo/enteredBy/@id="ABE" はid ABE を持つチームメンバーにより入力されたニュースアイテムのそれぞれのヘンドライン上のブレークポイントを有効化します。スクレーショナによのブレークポイントで一時中止されている評価を示しています(子のブレークポイントでのカッチ式がfalse を返すことに注意してくたさい。)
- 対応するコンテキストメニューのためコ個別のブレークポイトを有効化/無効化することができます。

ペイン内のソールドーコマンド

デバッグモード内の式の編集ダイアログのペインコよナビゲート、検索、およびコピーの機能を提供するポタンが含まれています(*上のスクレーン* ションナ)。これらのポ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コ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	クトップボードに選択された結果アイテムの植力ラムをコピーします。全てのカラムをコピーするコス、全てのカ ラムを含みコピーするコマンドに切り替えてくたさい(下を参照)。
<i>全てのメッセージをクリップボ</i> ード1 <i>二</i> 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。
<i>全てのカラムを含みコピーす</i> る	(i)全てのカラム、おさは(ii)値カラムのコ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 のラインに折りたたた	結果アイテムの植力ラムか複数ラインのテキストを含む場合、(新しいラインませは文字を含む場合)、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ソ全体を表示します。

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] ポタンカウリックされると、XPath 条件式を構築するための入力ヘルレーペインが表示されます。これら入力ヘルレート内にあるエイリーをダブルクリックすることで、テキスト ボックスのカーノル位置にそのアイテムか挿入されます。

 XPath 条件式の編集 				×
XPath 2 条件式: count(newsitem) 				
スキーマ属性や要素の選択	演算子/式の選択	₹	関数の選択	₹
選択: \$XML/presswatch/newsitems 選択: \$XML/presswatch/newsitems 日日名前空間 日日本XML(3-1>) 日レート要素 日く) presswatch 日く) newsitems 日く) newsitems 日く) newsitems 日く) newsitem	日軸 田比較 田条件 田Quantified/for 田論理 田ノード 田数値 田シーケンス 田Type	~	 田 Accessors 田 Boolean 田 コンストラクター 田 コンテキスト 田 変換 田 日付と時間 田 日 田 日 七 の他 田 ノード 田 数値 田 QName 田 シーケンス 田 文字列 田 URI 	*
□ 把对 XPath (A)	•	4	•	4
Javaの使用Netの使用			OK(0)	キャンセル(C)

入力へい やペインコお以下の3 種類があります:

- XPath 式内に要素と属性を挿入するするのスキーマソレー。相対的な XPath チェックボックスカ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 パスはコンテキストノードに対して相対的に挿入されます。コンテキストノードはスキーマソレーペインの下に表示されています。絶対 XPath 式にお キュメントルートから始まり、 Relative 相対的な XPath チェックボックスカチェックされていない場合選択されたノードのすっというすった。
- 演算子と式のための入力へルゲーペイン。以下が含まれます:(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ト順、おけは関数カテゴリ 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けは、フライトを選択してオプ ションを選択します。
- アルファベナ順、おさば関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レー。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 おさま フライト を選択してオブションを選択します。 The Names/Types オブションにより関数の引数を選択することができ、名前、おさま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工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、おけま、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、そのドロップダウンメニューから名前 おけま型を選択します。
- ペイン、演算子、条件式、おさ、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 プレースホルダ # 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) プレースホルダを含む関数の条件式が挿入のためにダブ 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ホルダ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、これらは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っこ内にかざし、 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 式の作成を手助けれます。

- <u>コンテキストノードとスキーマッリー</u> ソースペル内の選択テキストボックスはすくにコン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。スキーマソノー内のノードをダブルクトックし、XPath 式に挿入します。If the Relative XPath チェックボックスカ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、 や式と 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 ツ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。
- <u>XPath 関数の挿入</u>

関数の選択 ペイノは、XPath 関数をアルファベト順に、おけよ、機能別のグループ別にリストしています。(ペインの上のアイコンをクリックして、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。関数にコつ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。署名内の引数は、カ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、おけよ、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、おけよ、指定された型の1つのアイテムを示します)。引数は、名前、おけはデータ型ごとご表示されます。ペインの上の下れる、またします。引数は、名前、おけはデータ型ごとご表示されます。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、または、 date アイテムであることを示しています。関数の上にマンスをかざすと、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(がさい)。

直接のテキストエ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 クレーンションを参照)。

♦ presswat<u>c</u>h

presswatch/selection

f(x) fn:upper-<u>c</u>ase(xs:string?)

Ĵ_, an<u>c</u>estor-or-self∷ ∣

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。

ath 2 条件式:					
- 構築 評価		演算子/式の選択		関数の選択	-
翻: \$XML/presswatch/newsitems/newsite ● \$ selection ● newsitems ● \$ newsitem ● \$	m/me	 ■軸 ■比較 ■条件 ■Quantified/for ■論理 ■ノード ●数値 ■シーケンス ■ Type 	*	 Accessors Boolean コンストラクター コンテキスト ・ ・	
		1.			

[それ以外]条件分岐を使用する詳細については「次を参照してくたさい 条件テンプレート。

11.3 ツールバー

ツールーでは様々な Style Vision コマイが用意されています。ツール ー お以下のように構成されます:

- <u>書式</u>
- <u>テ</u>ーブル
- デザイン要素の挿入
- デザインフィルター
- <u>標準</u>

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。

ツールドーの位置

ツール・Hはア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のケイド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、ウイバウの両端ませま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、一領域にソール、一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・をフロート状態にするコま、ツール、一領域のかにソール、そボラッグし、ウィボ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、個々のソールレ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 チオにて、ツール チのチェクボックスを ٠ クリックすることで、個々のソールレ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。

ツールドーボタンを追加/削除

ツール・ーコある個々の村ダノは追加ませる当時できます。以下の操作により、ツール・ーから村ダンを追加ませる当時ですることがで 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 「追加または前除するホタンに倒進付けられたウールハーにて、(ワールバーがワ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。
 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)。

ボタンの追加/削除(A) 🔻	テーブル 🕨	✔ 囲 テーブルの挿入(T)
	デザイン要素の挿入 ▶	✓ ■ テーブルの削除(D)
	カスタマイズ	🔽 🖅 テーブル ヘッダー列の追加
		🔽 🖻 テーブル フッター列の追加
		✓ ⑥ [↑] テーブル ヘッダー行の追加
		🔽 🗗 テーブル フッター行の追加
		✓ 置 行の挿入(R)

- 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)。
- 4. 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か表示されます。

ボタンのノスト以下にある[ツールバーをリセット]アイテムにより、そのソールデーをStyleVision. をインストールた状態に戻すことが できます。リセルされると、そのソール・トーコあるすべてのがなっか表示されます。

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 メモ 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。

🖃 関連項目

Altova StyleVision 2021 Basic Edition
• <u>ユザイノターフェイス</u>

11.3.1 書式

[書式] ツール いはデザインビューで表示され、一般的なインラインならび ご ロックのフォーマ・ナ プロ・ ティを、SPS のデザイン こて選択されたアイテムへ割り当てるすっかのコマンド かきまれています。



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

対応する状シをクリックすることで、太字、イタリック、下線そして取り消し線のインライン テキスト プロイ ディが、現在デザインビューで選択されている対象に対して適用されます。ツール いーポタンからよ、スタイル、フォント サイズ、フォントの色と背景色を指定することもできます。

配置 配置プロ・ティ(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。

- 三 関連項目
- ・ツールト
- 書式とスタイル
- 入力フォーマナ

11.3.2 テーブル

[テーブル] ツール・ー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ドが収められています。テーブルツール・一内にあるこれらのコマナドを以下のスクレージョナに示します。



行と列の操作

SPS のテーブル(静的ならび、動的)に対して行おけありの挿入、追加、削除を行うことができます。行おける別は現在のカーノル位置の 直前に挿入、おけは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] ボタイにより行われます。[テーブル ヘッダー行/列の追加] おけは[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

現在カーノル的配置されている SPS テーブルセルを、隣接する4 つのセルの1 コン結合することができます。結合操作は HTML のテーブ ルンおける span 操作と似たものです。「右/左/上/下のセルを結合] ボタン、かこれらの操作のためご使用されます。更に [セル を水平方向に分割] おまま[セルを垂直方向に分割] ボタイより、カールは配置されている. SPS テーブルのセルを、それぞれ水平おまま重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ま、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] ボタンをクトックすることで、セルの境界線を表示おままにすことができます。

テーブルの操作、プロパティ、表示

静的ませは動的なテーブル内部にカーソルが配置された状態でテーブルの削除をクリックすることで、そのテーブルが削除されます。テーブル のマーケアップ表示は、[テーブル マークアップの表示] ポタイことり切り替わります。[テーブルのプロパティ] コマイドにより、そ のテーブルのプロペティを定義するケメのダイアログが表示されます。

- □ 関連項目
- ・ツールド
- <u>テーブル</u>
- <u>テーブルメニュー</u>

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 デザイン内部ヘデザイン要素を挿入するコマイドや、関連するコマイドに対応したア イエンが含まれています。ツール・・・アイエンから挿入することのできるデザイン要素を以下のスクレーンショットに示します。ツール・・・・・コは3 種類のアイテムが収められています:

- 1. デザイン要素,:コンテキストノードに影響を受けます(ツール・日本るみとの要素です)
- 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. グリンドに関連する操作:デザインの補助に使用されます



デザイン要素

デザイン要素は「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・トア イエルカらデザイン要素を挿入することかできます

- 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が表示されます。ダイアログにはコンテキストノードがハイトされた状態のスキーマッリーが表示されます。 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 す。
- 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 ツ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、<u>XPath 式 の編集ダイアログ</u> 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 おとはステップ4 でデザイン要素が挿入されます。

レイアナ要素

デザイン要素を挿入ツール・イコよ(i)レイアナコンテナー、(ii)レイアナボクスそして(iii)ラインを挿入するさめの3つのレイアナ 要素コマイが収められています。レイアナボクスとライノ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。

レイアオ コンテナーを挿入するコよ [レイアウト コンテナーの挿入] アイコを選択し、レイアオ コンテナーを挿入するデザイン上 の場所をクトックしてください。レイアオ コンテナーの大きさこして尋ねられ、選択によしイアオ コンテナーが挿入されます。レイアオ ボッ クスを挿入するコよ [レイアウト ボックスの挿入] アイコンをクトックし、レイアオ ボックスを挿入するレイアオ コンテナー上の場所を クトックしてください。レイアオ ボックスの挿入] アイコンをクトックし、レイアオ ボックスを挿入するレイアオ コンテナー上の場所を できます。ラインを挿入するコよ [ラインの挿入] アイコンをクトックして、レイアオ コンテナー内にあるラインの開始点までカーノルを移動し てくたさい。 クトックによりラインの開始点がセナされ、目的の終了点までカーノルをドラッグしてくたさい。終了点でカーノルをドロップすること で、指定された開始点から終了点までのラインの挿入されます。

レイアナトコンテナーやレイアナトボックスの大きさを変更するコよ、レイアナトコンテナーおコンイアナトボックスの右端おコよ下端までカーノルを移動させ、境界線をドラッグしてくたさい。レイアナトボックスを移動するコよ、レイアナトボックスの上端おコンを端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グによルレイアナトボックスを新たな場所へ移動してくたさい。

グルドに関連する操作

[グリッドの表示] コマドにより、グリドの表示/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オシィアナ ボックスやラインが、グリドド線やグリッド線の交差点に対してスナップされます。 グリドのプロ ティレオプションダイプログ([ツール | オプション])のデザインタブにてセルすることができます。

□ 関連項目

- ・ツールド
- 挿入メニュー
- <u> レイアサンテナー</u>

11.3.4 デザインフィルター

デザインフィルターツールバー(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レングを行うさめのコマンドが収められています。ツールバーにある各アイコンにて以下で説明します。.

68 68 6) 📫 🚓 🖷	EmailPerson 🔹

アイコン	⊐⋜ンバ	説明
ē	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コンをクリックすることで、そのテンプレートだけ放表示されます。
	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、グローイ VL 名前付き、レイアト) が表示されます。

アイコン	ب ۲×≻⊏	説明
Ð	インポートされたテンプレ ートを表示	インポート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/無効はします。
J	グロー・ゾル <i>テ</i> ンプレート を表示/隠す	グロー・ジレ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ノストが表示されます(以下のスクレーンショナを参照)。

🔁 Template1	-
🗰 (
🕰 n1:bold	
n1:italic	
📸 Template1	

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ノニて選択されます。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が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か表示されてい ないような状況で利用することができます。

□ 関連項目

- <u><u>y</u>-<u>Ju</u><u></u></u>
- <u>デザインソー</u>

11.3.5 標準

標準ッールバーには重要なファイルベ編集機能に関するコマイトが含まれています。簡単な説明とともここれらのアイコンを以下に示します。コマイド名をクリックすることで、リファレンスセクションにあるコマイドの詳細な説明を確認することができます。

アイコン	マンド	ショートカット	説明
t ₽	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 ドキュメトを新付ご作成します。ドロップ ダウンをクトックすることで、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。
Ĩ	<u>開</u> 人	Ctrl+O	既存のSPSドキュメトを開きます。
	<u>デザインを保存</u>	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	大トロ	ショートカット	説明
Ø	<u>すべて保存</u>	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。
6	<u>印刷</u>	Ctrl+P	作業用 XML ファイルのAuthentic View を印刷します。
C.	<u>印刷プレビュー</u>		作業用 XML ファイルのAuthentic View の印刷プレビューを表示します。
Ж	<u>切り取り</u>	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	選択対象をクリップボードにコピーします。
A	<u>貼り付け</u>	Ctrl+P	クトップボードの内容をカーソ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。
5	<u>元に戻す</u>	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。
<i>₿</i> %	<u>検索</u>	Ctrl+F	Authentic View や出力ビューインあるテキストの検索を行います。
Å	<u>次を検索</u>	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	<u>XSLT 1.0</u>		XSLT 1.0 がスタイルシート言語として設定されます。
XSLT 20	<u>XSLT 2.0</u>		XSLT 2.0 がスタイルシート言語と、て設定されます。
XSLT 3.0	<u>XSLT 3.0</u>		XSLT 3.0 がスタイルシート言語と、て設定されます。
ABC	<u>スペルチェック</u>		SPSドキュメントにてスペルチェンを行します。

□ 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を新たっ作成します。
- 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、おけま再ロードします。
- <u>デザインの保存、名前をコナて保存、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すめのコ マドです。
- <u>Mobile Together デザインファイルと、てエクスポートする</u>を使用して、アクティブな SPS ファイルから Mobile Together デザイン を生成することができます。
- <u>生成されたファイルの保存</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。
- <u>Web デザイン</u>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 ディ 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ています。プレビューイン表示されている内容を印刷するかのコマンドです。
- 最後に開かれたファイル、終了それぞれ最後に開いたファイルを開く、プログラムを終了するためのコマンドです。

11.4.1 新規作成

[新規作成] コマド上へマウスカーノルを配置すると、SPS ドキュメントを新たに作成するためのサブメニューカ表示されます(以下のスク リーンショントを参照):



- 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 デザク概要サイバーへ 追加され、(スキーマソノーサイババー(ある) スキーマツノー(スキーマの構造が表示されます。 デザインビュー では 空のテンプレ ートとともに SPSが作成されます。新規 SPS はURL やグロー・ ジレリノースを介したファイルからも作成することができます (下記参照)。
- <u>HTML ファイル</u>からノードごとご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(ルート要素)を持ったソノーがスキーマツノーに表示され、HTML ファイルがデザクノ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 からの新規作成」を参照くだれ、
- <u>デザイの静的テキストとてMSW ord ドキュメトのエレテンルを</u>含む新規SPS。
- <u>デザインの静的テキストとてMS Excel ドキュメトのエレテンルを</u>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)といたノー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表示され、(i)フレー・ハイのドキュメトデザイン、まけは(ii)フォームをベースはしたギキュメトデザイン(エノポーネ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。

新規デザインの	光作成
StyleVisionは 2 つのデザイン種類は ほぼすべてのデザイ)	種類のドキュメント デザインをサポートします:フリーハンドとフォームを基にしたものです。以下の説明を読んで 1 つ選んでください。2 ドキュメントの同一ページ内でも混合することができます。 両方のデザインにて、テキスト、テーブル、イメージ、編集フィールドなど、 ンアイテムが使用可能です。
<u>◎</u> フリーハンド@)ドキュメントを作成
	フリーハンドデザインは本、新聞、ドキュメンテーション、レポートのデザインを行う際に有効です。テキスト、イメージ、テーブルなどを ページ内で温合をせることができ、個々のサイズに従って自動的に配置されます。ページは自動的に生成され、テキストはページ に収まるようにラップされ、テーブルの自動幅設定を使うことでページの大きさに合うようにテーブルの大きさを変更することができま す。
◎ フォームを基(こしたドキュメントを作成(絶対位置)
Har -	フォームを基にしたデザインは、絶対位置で指定されたページにて編集コントロール、ラベル、イメージまたはテーブルを含む入力フ ォームを作成するのに最適です。通常すべてのページが個別にデザインされます。
Ľ	用紙上に既存のフォームをお持ちの場合は、その用紙をスキャンしてデザインの春写真イメージ ファイルとして使用することができま す。この方法を使えば春写真イメージにある位置に従ってデザイン アイテムを配置することができ、オリジナルのフォーム位置に一 致させることができます。
	青写真イメージを選択 参照(B)
	OK(0) キャンセル(C)

フレーット・「クトーン・アード・コントコンテンンが出力メディアオブシェクト(印刷物や画面) にフィット するようコンイア・ナ されま す。ドキュメントコンテンン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。

フォームをベースコンナドキュメントロン単一のレイアナ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。レイアナンテナーコス、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Iボキュメントコンテンルを配置 することしています。既にあるフォームのデザインをSPS にて再現したし、場合、オバンナルのフォームイメージを青写真イメージ 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 とグローバルリ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ひ、コファイルを保存ダイアログによってしま URL やグロー・ジレリノ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 スクレーシンョナを参照)。[URL にスイッチ]、おさま[グロー・ジレリソースにスイッチ] を選択することで、別の選択プロセスが表示されま す。

③ 静的アドレスの	参照	
ファイルの場所(1):	StyleVisionExamples	- G 👂 🗁 🖽 -
最近表示した場所	API HTML5 IDEPlugin Authentic Images Import	AddValidWithVariables altova article AutoCalc
デスクトップ	 IndustryStandards Music Archive RichEdit Tutorial XBRL Examples address AddValidOfValues 	BarcodeDemo BarcodeDemo BarcodeDemo BiggestCities BiggestCities BiggestCitiesPerCity BiggestCitiesPerContinent
ユンビュ シ () ネットワーク	 ✓ Ⅲ ファイル名(N): ファイルの種類(T): すべてのファイル (* URL (こ切り替え(U)) 	 開((O) **) ゲローバル リソースに切り替え(G)

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] コマドをクトックします。URL モードのファイルを開くダイアログが表示されます(以下のスクノーンショナを参 照)。

-					? <mark>×</mark>
ファイル URL(U):				•
開(形式: ③ 自動(A)) XML(X)	🔘 DTD(D)		ファイルのロード ③ キャッシュ /プロキシを使用(P)	◎ 再ロード(R)
ログイン情報 ユーサー(E):	MyDocs	パスワード(W):	•••••	アプリケーシ パスワートを	aン間で 保存
利用可能なフ サーバー	7-1/J http://vietspste	st/		•	参照(B)
Microsol	ft(R) SharePoint(R)	Server			
,				新規フォルダー(N)	
ファイル ダイア	ログに切り替え(F)	(ชื่อหาร์พ มีพ หลือ	:切り替え(G)	開く	キャンセル

- サードーURL フィールドにアクセスするURL を入力します(上のスクレーシンョナを参照)。サードーがMicrosoft(R) SharePoint(S) Server の場合、Microsoft(R) SharePoint(R) Server チェックボックスにチェックを入れてくたさい。この 種類のサーバーを利用する場合、下記のMicrosoft(R) SharePoint(R) Server に関するメモを参照くたい。
- 3. サーバーが、マワードを要求する場合、[ユーザー] どいマワード]のフィールドに必要な情報を入力してくたさい。
- 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。

開<				? <mark>×</mark>
ファイル URL(U):	ftp://go	l.tuwien.ac.at/opsys/Free	eBSD/releases/i386/ISO-IMAGES/READM	1E.TXT 👻
開<形式: ◎ 自動(A)	© ×ml(×)	OTD(D)	ファイルのロード ③ キャッシュ /プロキシを使用(P) () 再	ī□~ド(R)
ログイン情報 ユーザー(E):		א-עגו <i>ו</i>	アプリケーション間で パスワードを保存	
利用可能なファイ)	U			
ש–וז– ו Microsoft(R	ftp://gd.tuwien.)SharePoint(R)	ac.at Server	▼₹	新四(B)
	B 100 € B 100	9.1 9.2 9.3 README.TXT		•
			新規フォルダー(N)	削除(L)
วราม มีกราช	に切り替え(F)	グローバル リソースに切り替え	٤(G) 🕅 🛁	FPC-HZIL

ファイルURL が、ファイルURL フィールドに表示されます(上のスクレーンショナを参照)。この段階で、[開く] ボタンが選択可能 になります。

- 6. [開く] ボタンをクリックして、ファイルをロードします。開かれたファイルがダインウインドウン表示されます。
- メモ 参照機能はWebDAVなびに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になります。
- 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 こともみます。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 に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ケトリ構造では、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のシンボルとて表示されます(以下のスクレーシンコットを参照)。

開く		? 🔀
77-1/ม URL (U) : http://vie	etspstest/Documents/New%	20Text%20Document.txt 🔹
開(形式:	💿 DTD (D)	ファイルのロード ④ キャッシュ /プロキシを使用(P) 〇 再ロード(R)
ログイン情報 ユーザー(E):	パスワード(W) :	アプリケーション間で パスワートを保存
利用可能なファイル サーバー URL <u>http://vietspstest</u> I Microsoft(R) SharePoint(R) S	erver	▼ 参照(B)
	.xml iment.txt	
FormServerTem	チェックアウト(0) チェックイン(I) チェックアウトの取消(▼ (U) 行規フォルダー(N) 削除(L)
ファイルダイアログにスイッチ(F)	グローパルリソース(こスイッチ(G)	開く キャンセル //

ファイルを右クリックして表示されるコンテキスト メニューイゴよ そのファイルに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:

4 8 859	チェックイン状態。チェックアオ可能。
	他のユーザーによりチェックアナ状態。チェックアナ不可能。
5	ローカルマネルてチェックアナ。編集後にチェックイン可能。

- ファイルカチェックアナされている状態で、Altova アプリケーションにて編集した後に「ファイル | 保存 (Ctrl+S)] により保存することができます。
- URLを開くダイアログではエレテキストメニューのおか、アプリケーションのメインウインドウィーあるファイルタブ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、編集されたファイルをチェックインすることができます。

AutoCalc.sps

- 他のユーザーによりファイルカ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。
- ファイルカローカルでチェックアナされている場合、コンテキストメニューカら[チェックアナの取消] を選択することで、チェックアナを取り消すことができます。変更することはレーファイルカ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した場合、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。ファイルは既 にチェックアナされた状態だと認識されます。Microsoft® SharePoint® Server をサポートしている Altova アプルケーションで行う 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 おはPXF ファイルを開くことができます。Windows システムの ダイアログを開くダ イアログ が表示され、sps おはよ,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・ドにより、ディスクニ保存されている SPS ファイルを開き直します。ファイルカ最後に保存されてからなされたすべての変更 か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] コマイドにより、現在アクティブな SPS ドキュメイカ閉じられます。複数のファイルを同時に開くことはできるものの、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たさい。メインウィイウの右上にある閉じるボタンをクトックすることでも、アクティブなドキュメ ントを閉じることができます。ドキュメノ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 ドキュメントか閉じられます。開かれているドキュメント内に保存されていない 変更点かある場合、これらの変更を保存するか尋ねられます。

▼ URL とグロー・ジレリソースを使用してファイルを選択ませま保存する

「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 おさおブロー・ シリノノースを使用して必要なファイルを選択おさまファ イルを保存することができます(アのスクリーンショナを照))。選択プロセスに移動するために「URL に切り替える」 おさよ「グロ ー・ シリノノース」をクトックしてください。

Open			—
Look in:	🕌 Examples	- 3 🕫 🛛	୭▼
(Pa	Name	Date modified	Size ^
-	📾 cond-address.xsd	01/07/2014 12:10 AM	4 KB
Recent Places	🕞 Conditional.sps	01/07/2014 12:10 AM	41 KB
	📶 Conditional.xml	01/07/2014 12:10 AM	4 KB
	🔤 Conditional.xsd	01/07/2014 12:10 AM	5 KB
Desktop	🕞 Conditional-Final.sps	01/07/2014 12:10 AM	47 KB 🗏
-	🕞 DBSample.sps	01/07/2014 12:10 AM	25 KB
	🖲 DebuggerClient.htm	01/07/2014 12:10 AM	9 KB
Libraries	🛃 EU.bmp	01/07/2014 12:10 AM	1 KB
	Examples.spp	01/07/2014 12:10 AM	9 KB
	ExpReport.sps	01/07/2014 12:10 AM	166 KB
Computer	ExpReport.xml	01/07/2014 12:10 AM	2 KB
	ExpReport.xsd	01/07/2014 12:10 AM	7 KB
	ExpReport.xslt	01/07/2014 12:10 AM	168 KB
Network	💽 exterior.gif	01/07/2014 12:10 AM	18 KB 👻
	File name: ExpReport.xml		Open
	Files of type: All Files (*.*)		Cancel
	Switch to URL	Switch to Global Resources	н.

URL を使用してファイルを切り替える

URL から開くおけまれ存するファイルを選択するコキリ下を行ってくたさい

1. 「URL に切り替える」コマンドをクリックします。これにより、開くまは一保存ダイアログの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)。

)pen			×
File URL:			•
Open as: O Auto	© DTD	File load	ad
Identification User: MyDocs	Password:	Remember password between application	l starts
Available files Server URL: http://vietsp	stest/	- Brov	wse
V This is a Microsoft® Sha	arePoint® Server		
,		New Folder De	lete
Switch to File Dialog	Switch to Global Resou	ces Open Ca	ncel

- アクセスする URL をサーメー URL フィールドに入力します(上のスクレーシショナ)。サードーが 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。この種類のサーメート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. サードーが タワードにより保護されている場合、ユーザーとノ タワード フィーリドに入力してください。
- 4. 「参照」をクルクして、サーバーのディレクトリ構造をナビゲートします。
- 5. フォルダーツ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。

Open				
File URL: http://gd.tuwien.ac.at/vietspstest/_catalogs/lt/Forms/DispForm.aspx -				
Open as: Auto	File load			
Identification User: TestUser Password: ••••••	Remember password between application starts			
Available files Server URL: http://gd.tuwien.ac.at/ Image: This is a Microsoft® SharePoint® Server	■ Browse			
gd.tuwien.ac.at wietspstest catalogs wietspice wietspice				
	New Folder Delete			
Switch to File Dialog Switch to Global Resources	Open Cancel //			

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」ませは「保存」ボタンを使用できるようてないます。

6. 「開く」をクリックしてファイルをロードするか、「保存」をクリックして保存します。

<u>以下の点に注意してください</u>

- WebDAVをサポートするサードと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ュまゴまファイルが以前コロ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は、便に処理スピードを向上することができます)。また、電子出版まゴ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、ファイルアイコンココンアイルのチェッククレノチェックアナの状

o

態を表示するシンボルがおます(下のスクリーンショット)。

Open	×
File URL: http://vietspstest/Docs/Documents/flc/AutoCa	alc.sps 👻
Open as: Auto OXML ODTD OUse	ad e cache/proxy 🔘 Reload
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/	■ Browse
This is a management of a control Documents Origination of the control Documents Origination of the control Origination of the contro Origination of the co	
Berdiw Forms Undo Check Out	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

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:

	チェックル状態。
Î	チェックアオ可能。他のユーザーによりチェックアオ状態。
ß	チェックアナ不可能。ローカルに対してチェックアナ。

- ファイルをチェックアナした後、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」ダイアログのコ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(上のスクレーンショット参照) おさよ 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リックすることことはポップアップするコンテキストメニューを 使用して(アのスクレーンショント)。

🛃 AutoCalc.sps	Persons.xml 💼 ExpReport.xml
900095355555555555555555555555555555555	1

- ファイルが他のユーザーによりチェックアナされている場合、チェックアナすることができません。
- 他のユーザーによりファイルカ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てきません。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。
- 変更することなしに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ンドは、以下の通りです:「チェッククレン」および「チェックアナトを元に戻す」です。
- ▼ グロー・ ジリノノースを使用してファイルを選択ませま保存する

グロー・ ジリノノースを使用してファイルを開くおけは保存する場合、「グロー・ ジリノノース」をクトックします。 グロー・ ジレーノースを選択 するけっかのダイアログか表示されます。 これらのダイアログは以下のセグションで説明されています… グロー・ ジリノノースの説明に関しては このドキュメントのセグションを参照してください。

- 三 関連項目
- <u>ファイル</u>新規作成
- <u>メインウィドウ</u>

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) とて保存します。

[すべて保存] (Ctrl+Shift+S) コマイ 🔐 により、開かれているすべての SPS ドキュメト か保存されます。

▼ URL とグローイジレリノースを使用してファイルを選択ませは保存する

「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 おうまグロー・ ジレハノースを使用して必要なファイルを選択おうまファ イルを保存することができます(アのスクノーンショナ参照))。選択プロセスに移動するために「URL に切り替える」 おうま「グロ ー・ ジレノソース」をクトックしてくたさい。

Open					×
Look in:	\mu Examples	•	G 🤌 🖻	۶ 🛄 ד	
Ca	Name	Date modified	d	Size	•
	🐻 cond-address.xsd	01/07/2014 12	2:10 AM	4 KB	
Recent Places	🔄 Conditional.sps	01/07/2014 12	2:10 AM	41 KB	
	📶 Conditional.xml	01/07/2014 12	2:10 AM	4 KB	
	🐻 Conditional.xsd	01/07/2014 12	2:10 AM	5 KB	
Desktop	ᡖ Conditional-Final.sps	01/07/2014 12	2:10 AM	47 KB	=
-	🕞 DBSample.sps	01/07/2014 12	2:10 AM	25 KB	
1	DebuggerClient.htm	01/07/2014 12	2:10 AM	9 KB	
Libraries	EU.bmp	01/07/2014 12	2:10 AM	1 KB	
	🕞 Examples.spp	01/07/2014 12	2:10 AM	9 KB	
	🕞 ExpReport.sps	01/07/2014 12	2:10 AM	166 KB	
Computer	ExpReport.xml	01/07/2014 12	2:10 AM	2 KB	
	ExpReport.xsd	01/07/2014 12	2:10 AM	7 KB	
	ExpReport.xslt	01/07/2014 12	2:10 AM	168 KB	
Network	💽 exterior.gif	01/07/2014 12	2:10 AM	18 KB	-
	File name: ExpReport xml			- Op	en
	Files of type: All Files (*.*)			 Can 	icel
	Switch to URL	Switch to Globa	Resources		.11

URL を使用してファイルを切り替える

URL から開くおけまれ存するファイルを選択するコキリ下を行ってくたさい

1. 「URL に切り替える」コマンドをクリックします。これにより、開くまは一保存ダイアログの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)。

Open	—
File URL:	•
Open as: O Auto	File load
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/	✓ Browse
✓ This is a Microsoft® SharePoint® Server	
	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

- 2. アクセスする URL をサーメー URL フィールドに入力します(上のスクレーシショナ)。サーメーが 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 のサーメート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. サードーが タワードにより保護されている場合、ユーザーとノ タワード フィーリドに入力してください。
- 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リックします。

Open 🔀				
File URL: http://gd.tuwien.ac.at/vietspstest/_catalogs/lt/Forms/DispForm.aspx -				
Open as: O Auto				
Identification User: TestUser Password: ••••••• Between application starts				
Available files Server URL: http://gd.tuwien.ac.at/ This is a Microsoft® SharePoint® Server				
Image: state of the orthogonal Image: state orthogona Image: s				
New Folder Delete				
Switch to File Dialog Switch to Global Resources Open Cancel				

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」ませは「保存」ボタンを使用できるようこないます。

6. 「開く」をクリックしてファイルをロードするか、「保存」をクリックして保存します。

<u>以下の点に注意してください</u>

- WebDAVをサポーするサーベーと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、ローカルのキャシュまゴよファイルが以前し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とし使に処理スピードを向上することができます)。また、電子出版まゴ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
- 0

▼ 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リ構造では、ファイルアイエノコはファイルのチェッククレノチェックアナの状

態を表示するシンボルがおます(下のスクリーンショット)。

Open	—
File URL: http://vietspstest/Docs/Documents/flc/Au	utoCalc.sps 👻
Open as: Auto OXML ODTD ©	le load) Use cache/proxy 💿 Reload
Identification User: MyDocs Password: ••••••	Remember password between application starts
Available files Server URL: http://vietspstest/	■ Browse
This is a Microsoft o Shaler Onto Server Documents Or chr AutoCalc.sps Check Out Bank_Java.ur Check In Check In Check Out	
Marketing	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel //

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:

	チェックル状態。
Î	チェックアオ可能。他のユーザーによりチェックアオ状態。
ß	チェックアナ不可能。ローカルに対してチェックアナ。

- ファイルをチェックアナした後、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」ダイアログのコ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(上のスクノーンショット参照) おさよ、アプリケーションのメインウィバウのファイルタブを右クリックすることことはポップアップするコンテキストメニューを 使用して(アのスクノーンショッケ)。



- ファイルが他のユーザーによりチェックアナされている場合、チェックアナすることができません。
- 他のユーザーによりアイルカ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。これによりアイルを変更す

ることないサーバーに戻すことができます。

- 変更することなしに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した場合、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ンドは、以下の通りです:「チェッククレン」および「チェックアナトを元に戻す」です。
- ▼ グロー・ ジリノノースを使用してファイルを選択ませょ保存する

グロー・ ジリノノースを使用してファイルを開くおけは保存する場合、「グロー・ ジリノノース」をクトックします。 グロー・ ジレーノースを選択 するけっかのダイアログか表示されます。 これらのダイアログは以下のセグションで説明されています… グロー・ ジリノノースの説明に関しては このドキュメントのセグションを参照してください。

- 関連項目
- <u>ファイル 閉じる</u>

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 ファイルまけはPXF (Portable XML Form) ファイルとして保存することができ ます。コマイをクルクすると、デザインの保存ダイアログが表示されます(以下のスクレーシンヨナトを参照)。目的のフォーマナを選択して、 [OK] をクルクしてください。



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、デザインをベースにした HTML 出力を行うのに必要なすべてのファイルを簡単にや取りすることが可 能になります。

SPS とて保存する

SPS オプションを選択すると、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「<u>デザインを保存</u> コマンド」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。

三 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 Mobile Together 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 Mobile Together デザインファ イルはモ イルデ イスのナダに Mobile Together アプリ内で ノリューションを実行する かいき用されます。 例えば、 Mobile Together ソ リューションは、スマート フォンなどのモ イルデ イスで、データベースの内容を閲覧し編集するために開くことができます。 Mobile Together ソリューションは Altova Mobile Together Designer 内でデザインすることができます。 このコマンドは、 Mobile Together Designer 内 で編集することができる、 SPS デザインを Mobile Together デザイン で変換することができます。 詳細に関しては、 <u>Mobile Together ウェ</u> フペーンを参照してくたさい。 オデノョンダイアログ(「ツール | オブノョン」)の Mobile Together Design タブ内で変換オプノョンを使用す 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ズを持つとは限りません。このコマンドを実行下の 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たさい。以下のデザイン機能は MobileTogether デザイノニエク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・シリテンプレート(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]コマイにあるサブメニューコよ以下のファイルを保存するすめのオプションが含まれています(以下のスク リーンショナを参照)。生成されたファイルの利用方法については、使用方法 | 生成された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XSLTHTML 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] コマンドにより、HTML ファイルか生成されます。デザイノニて複数のドキュメントの出力 が指定されて しる場合、複数のHTML ファイルが生成されます。この操作にコは2 つの入力ファイルが必要しています:

- 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 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しなります。
- 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、このコマンドが有効になります。スキーマッソー・サイドバートこで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ージは応する、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] コマイにより、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たトリガーに従うかたちで変換 を生成するさめのジョブを、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たさい。

.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 こなりますが、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 ファイルから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を選択してくたさい。

以下の点に注意してくたさい

- 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] コマボを実行する前に 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たさい。FlowForce Server のセオ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たさい(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] コマイドにより、変換のデプロイダイアログが表示されます(以下のスクレーンショナを参照)。

		×
コイする FlowForce 管理インターフ:	ェイスのホスト名とポートを	入力してくださ
127.0.0.1	ポート	8082
root		
••••		
/public/AutoCalc.transformation	n	参照(B)
パスの先頭にはスラッシュ文字が	必要です。	
作成に Web ブラウザーを開く(J)	ОК	キャンセル
	 ロイする FlowForce 管理インターフ: 127.0.0.1 root ・・・・ /public/AutoCalc.transformation パスの先頭にはスラッシュ文字が /作成に Web ブラウザーを開く(1) 	 ロイする FlowForce 管理インターフェイスのホスト名とボートを 127.0.0.1 ポート root ・・・・ /public/AutoCalc.transformation パスの先頭にはスラッシュ文字が必要です。 /作成(こ Web ブラウザーを開く(1)

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ングをデプロイする場合、「SSL を使用」チェンクボックスを選択します。これは、FlowForce Server か既に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たさい。

[OK] をクリックすることで、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] チェックボックスニ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ショブ 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)。

<u>StyleVision Server の複数のバージョン</u>

.transformation ファイルをデプロイするサーバ上に、FlowForce Server 管理下で Style Vision Server の複数のバージョンが 作動している場合、(Windows サーバーのみに適用可能)、「Select Style Vision Server の選択」ダイアログ内で、このマメピングが 実行される Stylevision Server バージョンを指定してください。使用するバージョンを手動で選択することかき、おこ、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 バージョンのためにFlowForce Server インネールディントリに .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 とてインネールする際、StyleVision Server .tool ファイル省動的にのディントリに追加されます。FlowForce 内に.tool ファイルな存される 対よ 以下のとおだです: C:\Program Files\Altova\FlowForceServer2021\tools。FlowForce Server 管理下で実行する StyleVision Serverの追加バージョンが存在する場合、.tool ファイルが上記のディントリに手動で追 加される必要がある場合がかます。StyleVision Server の.tool ファイルは以下で検索することができます: C:\Program Files\Altova\StyleVisionServer2021\tetc. **メモ** FlowForce Server の使用方法については、FlowForce ドキュメノ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] コマド [Jは[ASPX Web アプリケーションの生成] コマド かサブメニュー ご含まれています。後者のコマド に より、ASPX アプ ケーションを動作させる かい 必要なすべてのファイルが、指定され こフォルダー 内部に生成されます。 ブラウザー こより 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 パッケージ内にある XSLT ファイル こよ 動的に変換され こノース データベースや XML ファイルのデータが 張示されます。 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。

は関する詳しい説明は、次を参照してくたさい、ASPX インターフェイスによるWeb アプリケーション。

11.4.9 プロパティ

プロパティコマンドにより、プロパティダイアログ SPS のプロパティをセットするためのプロパティダイアログが表示されます

出力

出力ダブでは以下のプロノティをセントすることができます

- 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 ドキュメント全体、おさよ 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、すなオカ同じレベルでの、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 サポート: IE 6 前のInternet Explorer (IE) ・・ジョン内のCSS サポートは、一部のアスペトで不十分な箇所がある可能性があます。CSS サポートは、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 プロ 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*ント)。IE 5、IE 7、ま たまれIE 9 を選択することができます。(IE 9 互換性を適用するコは IE 9 おまれ以降がインストールされている必要がありま す)。IE サポートの指定されたレベルは、すく「こHTML プレビューで使用することができます。新規のSPS ドキュメントは、IE 7 互換性を選択して作成します。Altova Style Vision の古し V デジョンで作成されたま SPS ドキュメントを(プロ) ディダイアロ 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、シデーターおよびSPS内で使用されるXSLT、シージョンを指定することができます。

StyleVision には XSD 1.0 パデーターとXSD 1.1 パ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 ドキュメントの/xs:schema/@vc:minVersion 属性が1.1 に設定されている場合、XSD 1.1 バデーターを使用します。それ以外の場合、XSD 1.0 バ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 バデーターを使用する。

このダブからアクティブなドキュメトのために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 属性が存在しない場合、スキーマからデータが読み取られるようにするコよこのオプションが選択されていることが奨励されます。



11.4.10 印刷プレビュー、印刷

[印刷プレビュー] コマド 山」はデザインビューならけにAuthentic View にて有効になります(Authentic View はEnterprise ならびに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クティブの場合は) SPS デザイン、 おさは(Authentic View かアクティブの場合は) 作業用 XML ファイルのAuthentic View のプレビューか含まれるウィ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い、タグなしで、おさはタグとといて表示されます。



印刷プビューウィボウでは、ページ上部にあるソールレー、おさはページ下部にあるナビゲーションアイコンから、以下の操作を行うことができます。印刷プビューツールレースあるコマボを左から順に以下に示します。

- 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たさい。

Authentic View ならびに出カプレビュータブでは 🗐 か有効になります。選択されナビューの作業用 XML ファイルが、ビューイズして 指定されたページのセルアップしたからで印刷されます。 Authentic View のページ セルアップは印刷プレビュー ウィンド 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。

□ 関連項目

• ファイル プロパティ

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リストが、ファイル名や、マ情報とともに表示されます。エトリーの1 つをクリックすることで、そのファイルか新たなタブニてメイン ウィンドウェ開かれます。

6	印刷(P)	Ctrl+P
6	1 C:¥Users¥¥NanonullOrg.sps	
	2 C:¥Users¥¥BookmarksStatic.sps	
6	3 C:¥Users¥¥OrgChart.pxf	
6	4 C:¥Users¥¥RichEdit_Native.sps	
6	5 C:¥Users¥¥QuickStart.sps	
6	6 C:¥Users¥¥Conditional.sps	
6	7 C:¥Users¥¥Conditional-Final.sps	
6	8 C:¥Users¥¥AutoCalc.sps	
	終了(X)	

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は開かれている場合、これらのファイルを保存するか尋ねられます。

- 三 関連項目
- <u>メインウィンドウ</u>

11.5 編集メニュー

[編集] メニューーコは SPS ドキュメンの編集をサポートするオングのコマン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) といす 標準的な編集コマンドの他にも、以下のコマン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。
- <u>検索、次を検索、置き換</u>: SPS や とXSLT スタイルシート プレビュー内にあるテキストの検索を行います。
- <u>スタイルシートの、ラメーター</u>: SPS に対してグローイ いに宣言された ・ラメーターの編集をします。
- <u>マーケアップの縮退/展開</u>: SPS デザインコンポーネト タグを縮退/展開します。

コマボイはコンポーネトやカーノルの挿入位置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もアクセスできます。更にしくつかのコマボはキーボードのショートカナやソールレーのアイコン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やカーノルの位置により利用できないロマボはグレーアナされます。

□ 関連項目

• <u>">-11~</u>

11.5.1 元に戻す、やり直し、すべて選択

[元に戻す] (Ctrl+Z) コマボ 🔄 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はなく、 行オ: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) コマボ 🖸 により、それまで元に戻された操作をや、値すことができます。[元に戻す] と[や値し] コマボを使用 することで、コマボの履歴を行き来することができます。

[すべて選択] コマボにより、デザインドキュメトウイボウェあるすべてのコレテンンが選択されます。

🖃 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 🎒 デザインビュー、JavaScript エディター、とHTML-のための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にないます

デザインビュー、HTML プレビュー、

検索 コマイ・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 サイドバーか有効化されます。

以下の点に注意してくたさい

- デザインビューでは静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コは「単語の完全マッチ」をチェックしてくたさい。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コよ [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たさい。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 を検索できるようこなります。

XSLT-for-HTML & JavaScript Editor

XSLT-for+HTML おけよ JavaScript エディタータブ内の検索コマンドをクリックすると以下のダイアログか表示されます。

V www		~ 😵 🗙
Aa Abc 🔸 🚱 🗉 들	2の1	

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オノニジル替えられます(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 fitlistして、単語の完全マッチがすれた 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。
- 正規表現オインが替えられると、検索用語は、正規表現として読み取られます。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、ドキュメント内の一致が、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。アンカーの検索の別い替えは、最初の選択がサーンルの位置に対して相対的かを決定します。アンカーの検索がオノゴ切り替えられると、現在選択されている一致が選択され、現在のカーンルの場所の次の一致に一致します。アンカーの検索がオフェ切い替えられているし、現在選択されている一致はボキュメントの最初から数えて最初の一致が一致します。
- 選択範囲内での検索: オイン切り替えられると、現在のテ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、ドキュメン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ションをオスン切り替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・ド

次を検索(F3) コマド 「 は リケエスト されナテキストの次の発生を検索するオークリニ最後に使用されナー検索コマドを繰り返します。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 lt Basic エディションではサポートされていません)内で置換 コマイ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。これを行うコよ、最初に 正規表現 オプシュンをオノゴル 潜え まず(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ルプするオームに、検索用語フィールドの右にある 正規表現 ビルダー ポタンをクトックします。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たさい。

<u>正規表現メタ文字</u>

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カレダーです。
λ(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。
\(abc \)	↓ というメタ文字は、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、おけよ、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、シグル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 採文字列がSam、1 た式か存在することを意味し、おこ、置換文字列内に、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 た式か存在することを意味し、おこ、置換文字列内では、タグ付けされた式は、1 と参照されています。検 素と置換式が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 が使用できるようことはます。例えば、(に) 文字 セナの始まりではなく、 (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 おさな 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ために使用できます。
[^]	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。(セオ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 に一致するよど。
+	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	キャトッジターン(CR)。CR(\r)お」ままLF(\n)を使用して新規のラインを検索おままれ作成します。	
\n	改行(LF)。CR(\r)ませまLF(\n)を使用して新規のラインを検索ませま作成します。	
\t	ダブ文字	
11	正規表現内に表示される文字をエスケープするために使用してくたさい。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。しかしなから、通常の検索と置換え操作ではテキストを操作できなレ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 つのオペーション内の特定の要素を、囲まれたコンテンソニ影響を与えずに、名前を 書き換える必要かある場合か挙じたれます。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 に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コードリストはよ、ブックのリストか含まれています。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</year>
   </book>
   <book id="2">
      <author>Franz Kafka</author>
      <title>The Metamorphosis</title>
      <category>Fiction</category>
      <year>1912</year>
   </book>
   <book id="3">
      <author>Herman Melville</author>
      <title>Moby Dick</title>
      <category>Fiction</category>
      <year>1851</year>
   </book>
</books>
```

必要条件を満たすコよ以下のステップを踏んてくたさい

- 1. 検索と置換ダイアログボックスを開くには、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。



(既知ではない)各要素の内部のテキストに一致するには、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」は複数の文字 .+ 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ります。

- 4. 置換フィールド内に、次のテキストを入力します。

 くgenre>\1</genre> 。この正規表現置換えのテキストを定義します。

 検

 素フィールドから既にタグ付けされた条件式に、シグルファレンスするために

 い1 が使用されていることに注意してください。

 すなオカ、

 このエレテキストでは、

 \1 「現在一致するくcategory〉要素の内部のテキスト」を意味します。
- 5. 全てを置き換える 2 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コードリストはは製品のノストが含まれています。各製品はは以下の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。(すなオム、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。

必要条件を満たすコよ以下のステップを踏んてくたさい

- 1. 検索と置換ダイアログボックスを開くには、Ctrl+H を押します。
- 2. 正規表現の使用 📩 をクルクします。

1 2	<pre><?xml version="1.0" encoding="UTF-8"?> =<pre>cproducts xmlns:xsi="http://www.w3.org/2001/XML</pre></pre>	★×
	products.xsd"> <product id="\1" size="\2"></product> <product size="\1"></product> <product <="" size="\1" th=""><th>⊊b ⊊ab</th></product>	⊊b ⊊ab
3	<pre><pre>product id="1" size="10"/></pre></pre>	
4	<pre></pre>	▲ ▶
5	<product id="3" size="30"></product> "	
6	<product id="4" size="40"></product>	
7	<product id="5" size="50"></product>	
8	<product id="6" size="60"></product>	
9		

- 6. 全てを置き換える 画 をクルクして、結果を確認します。すべてのproduct 要素は属性 size が属性のid 前に置かれ るように更新されました。
- 三 関連項目

• <u>ツールド</u>

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、ラメーターコマド 🗐 により、 ・ 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。 コマド はデザイ ンビューならび こ Authentic View で利用することができます。 このコマドを選択すると、 以下に示される ・ ラメーターの編集ダイアログ か表示されます。

パラメーターの編集				
パラメーター名は先頭の '\$' なしで入力してください。				
見習	X			
パラメーター名	デフォルト値			
state1	CA			
state2	NY			
	- 1			
	OK ++>>tz,1			

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イン概要サイドバーでも宣言することができます。
- 三 関連項目
- <u>15x-9</u>

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けは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ナ、SPS メカニズムコマルで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。
一連のタグか縮退される様子を以下のスクノーンショナに示します。縮退されたタグをダブルクトックすることで、タグか展開されます。



タグを縮退することで、編集のニーズに従い最適な表示を行うことができます。

- 三 関連項目
- シンボル
- <u>テンプレートフィルター</u>

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さざのコマンドが含まれています(以下のスクノーンショントを参照)。個々のソールバー、デザインサイドバー、デザインフィルター、ステータスパーの表示を切り替えることができます。

	Toolbars
6	Design <u>O</u> verview
Le	<u>S</u> chema Tree
-	<u>D</u> esign Tree
6	Style <u>R</u> epository
Ŀð	Context Properties
L-À	<u>C</u> ontext Styles
	Design <u>Fi</u> lter 🕨 🕨
~	Status <u>B</u> ar

- 三 関連項目
- <u>ユーザーインターフェイス</u>
- <u>ツール</u>ー

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、一の表示を切り替えるためのサブメニューか表示されます(以下のスクリーンショントを参照)。

>	<u>F</u> ormatting
~	<u>T</u> able
~	Insert Design Element
~	<u>D</u> esign Filter
~	<u>G</u> lobal Resources
>	Standard Toolbar

有効」なっているソールドーが表示されます。上のスクレーンショナでは、すべてのソールドーが表示されています。ツールドーをクトックすることで、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ついては、次のセクションを参照してくたさい、レファレンス、ツ ールドー。

ステータスバー

アプリケーションウィン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」コマンドにより別をえることができます。

- 三 関連項目
- ・ツールド
- <u>ツール カスタマイズ</u>

11.6.2 デザインサイドバー

[表示] メニューコマイには各サイドバーの表示を切り替えるためのコマイが含まれています(以下のスクレーンショナを参照)。

	Toolbars	
-0	Design <u>O</u> verview	
ნ	<u>S</u> chema Tree	
-	<u>D</u> esign Tree	
6	Style <u>R</u> epository	
낭	Context Properties	
Ŀ <u>A</u>	<u>C</u> ontext Styles	
	Design Eilter	
~	Status <u>B</u> ar	

サイドバーが有効」なっている(コマイドのアイエイニレームがついている)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。このコマイドは、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、あるビュートこで行われた変更は、新たっ変更 されるまで、そのビュートネリして保持されます。

- 三 関連項目
- <u>ユーザークターフェイス デザインサイドバー</u>

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レンプするすめのコマンドがサブメニューに 収められています。デザインか非常に長い、おすす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	۲<>⊥	説明
ē	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コンをクリックすることで、そのテンプレートだけが表示されます。
Ē	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・ VL 名前付き、レイアト) か表示されます。
Ē	インポートされたテンプレ ートを表示	へポートされたテンプレートの表示を有効/無効にします。
	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。
വ	グロー・ ヾル テンプレート を表示 ⁄隠す	グロー・ジレテンプレートの表示を有効/無効にします。
æ	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。

これらのコマイ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まか、Ctrl キーとマウスのスクロールホイールにより、拡大/縮小を行うこともできます。

- □ 関連項目
- ツールー デザインフィルター

11.7 挿入メニュー

[挿入] メニューイコよ、様々なデザインコンポーネントを SPS ヘ挿入するオメクロ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</u> ノ<u>バーアイコン</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 メニューからも利用できます。コンテキスト、メニューでは、SPSのカーノル位置に挿入できない ロマンがはグレーアウト されます。

メモ [挿入] コマドはSPS の構築に使用されるため、これらのコマドはデザインビューでのみ利用できます。

- 三 関連項目
- <u>ユーザーイクターフェイス</u>
- ・<u>ツール〜</u>

関連項目:

<u>コン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・一個々のツールバーについて詳細に記述されます。

タブで囲むメニュー・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] コマドにより、カーノル位置に(コンテンツ プレースホルダが挿入されます。(コンテンツ プレースホ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」ないます。

- 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 にコングロー、ジレ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これ、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ルールか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グロー、ジレテンプレートが存在しては、限り、ませは(コンテンツ)プレースホルダやその他のエンテンツ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たいかぎり、(コンテンツ)プレースホルダでは、属性の値が出力されたいということに言してくたさい。
- 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。

ノードのタグコーナリルを配置して、右クトック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 <u>へと要素の挿入 ツールドー</u>にある[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) プレース市い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ケテを使用することで、(コンテンツ)プレースホ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、ノードのエンテンメに適用されます。

コンテンソの置換え

スキーマッノーから別のノードが(コンテンツ) プレースオ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) プレースオルダが新し いノードには置き換えられます。

コンテンソの削除

(コンテンツ) プレース市ルダを選択し、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
- <u>テキストとして XML エレテンンを挿入</u>
- <u>出力の構造</u>
- その他のエレテンソ

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。このプレースホルダは、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 ルールニ対応します。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たさい。要素タグ内部を右 クトックして、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) プレースホルダを挿入 することもできます。

(その他のエンテンツ) プレースカルダを使用することで、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こなはす。

(その他のエレテンツ) プレース市以多を選択し、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

- <u>テキストとてXML コンテンクを挿入</u>
- 出力の構造
- その他のエンテンソ
- <u>エレテンソ</u>

11.7.3 フォームコントロール

[フォーム コントロール] コマボヘマンズを移動すると、各種フォームコトロール(<u>データ入力デバイス</u>)を挿入するオメのコマボカ表示されます(以下のズクノーンショントを参照)。

下 入力フィールド(I)
 下 複数行フィールド(M)
 「 チェック ボックス(H)…
 □ コンボ ボックス(O)…

- ラジオボタン(R)...
- ボタン(B)

[フォーム コントロール] コマボヘマウスを移動すると、各種フォームコナトロールデータ入力デバスの使用のセグタンを参照ください。フォームコナロールを作成すると、そのコナトロールを選択し、 プロノティサイドバートにてプロノティの編集を行うことができます。

デザインの挿入ポイントを右クトックして、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</u> <u>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トックして、デザイン上の位置をクトックすることで、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。
- 三 関連項目
- データ入力デバイスの使用
- <u>プロ/ ም</u>

11.7.4 自動計算

[自動計算]を使うことで、XPath 条件式を使った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、おけ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、右クルクするか[挿入] メニュー コマンドを使用してくたさい、[挿入 | 自動計算] にマウスカーノルを移動すると、どのようこ 自動計算が挿入されるかを選択するためのメニューが表示されます(以下のスクノーンショントを参照)。デザイン要素の挿入 ツール・トームあ 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イに表示されること」なります。

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(以下のスクノーシンョナに示される) XPath 式の編集ダイアログ(アのスクノーンショナ) ダイアログが表示されます。

人キーマ病性や要素の選択 選択: \$XML/presswatch/newsitems 選択: \$XML/presswatch/newsitems 田田名前空間 日 日子参加 第XML (メイン) 日子の一下要素 日子の一下要素 日子の中wsitems 日子のnewsitems 田〇 newsitems 日子の目子の目子の目子の目子の目子の目子の目子の目子の目子の目子の目子の目子の目子	演算子/式の選択 田軸 田比較 田条件 田Quantified/for 田論理 田ノード 田数値 田シーケンス 田Type		関数の選択 田 Accessors 田 Boolean 田 コンストラクター 田 コンテキスト 田 変換 田 日付と時間 田 ID 田 その他 田 ノード 田 妙値 田 QName 田 シーケンス 田 文字列 田 URI	
I 絶対 XPath (A)	4	F.	4	Þ

構築されている条件式のコ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しずれかの)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たい、場合) コンテキストノ ードにつけして相対的、ませは、(「絶対 XPath] にチェックが入っている場合) 絶対的なもの」さいます。

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。

- 三 関連項目
- <u>XPath 式 の編集ダイアログ</u>
- <u>自動計算</u>

11.7.5 段落、特殊な段落

[段落] コマイ
[1] により、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、まけは精約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。カーノルの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)か挿入されます。段落は、<u>デザイン要素の挿入 ツール・</u>トームある[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、選択されたノードに対して定義済みのフォーマナを割り当てることができます。ツール ~のコンボボックスから ら、定義済みのフォーマナを選択することができます。

各種段落ははそれぞれ個別のフォーマナ機能が含まれています。pre フォーマナトでは、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たさい。

🗉 関連項目

- <u>定義済みのフォーマナ</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(以下のスクレーンショナを参照)。 デザイン要素の挿入 ツールレ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コは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。

それそれのタブの詳細を以下に記します。

静的

イメージのURI がアドレスフィールドに直接入力されます(以下のスクレーシショナを参照)。以下のスクレーシショナでは、 http://www.altova.com/pix/Marketing/logo.png とうURI が使用されています。

イメーシ	の挿入		
静的	動的	静的ならびに動的 インライン	データ
アドレ	オ ス(E): h	的 ttp://www.altova.com/pix/Mai]絶対パス 参照…	keting/logopng 単純な静的アドレスの入力に、 上のフィールドを使用してください。
			OK キャンセル

URI は絶対パマまけは相対パマとして入力することができます(「相対パマ チェックボックスことが選択)。相対 URI が入力された場合、そのパマは SPS ファイルが配置されている場所に対して相対的パコンドます。「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、(絶対まけよ相対パマの) URI を入力することできます。

動的

XPath 条件式によりイメージのURI が返されます。以下のスクレージョナでは、@deptlogo とら XPath 条件式が使用されています。 これはイメージのURI がコンテキスト 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 がコンテキスト ノードしなります。

イメージの挿入		
静的動的	静的ならびに動的 インラインデー	9
1	防的	
アドレス(E) :	@deptlogo	
	XPath を編集…	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。
E	XSLT 変換に対して解析対象外工	ンティティとして扱う
		OK キャンセル

[XPath を編集] ボタンをクリックすることで、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 をベースはしており、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 ティティか含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ていては:ノーースされて いないエンティティ URI のセクションを参照してくたさい。

静的と動的

静的ならびし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。

イメーシ	の挿入				×
静的	動的	静的ならびに動的	インラインデー	-タ	
	青争白	5		動的	青争自匀
アドレ	ス(E) : ከ	ttp://www.altova.com	n/pix/	@deptname	/logo.png
2つの間 作成し	静的ならで します。	びに1つの動的表現に	よりアドレスを	XPath を編集	
				🔲 動的なパートを XSLT 変換に対し	て解析対象外エンティティとして扱う
					OK キャンセル

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、上のスクレージタト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 どみます。動的な部分においては 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 封はBase-64 によりエンコードされナテキスト とて 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、エンコードされナ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] ユボボックスによ リ、ソース XML にて、StyleVision 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 フォーマナ] ユズボボック スでは、生成するイメージファイルのフォーマトか指定されます(イメージファイルはエンコードされナテキスト 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)。

イメージの挿入	x
静的 動的 静的ならびに動的 インラインデータ	
インラインデータ XPath : //images/marketinglogo	
イメージフォーマット:png マ Iンコーディング:base64 マ XPath を編集… イメージファイル設定…]
OK キャンセル	

[イメージ ファイル設定] ポタンをクト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し、ビッ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 前か生成されます。

- □ 関連項目
- <u>イメージの使用</u>

11.7.7 水平線

[水平線] コマイドにより、カーノル挿入位置に水平線が挿入されます。SPS コンポーネントが選択された状態でこのコマイドを選択すること はてきません。水平線のプロ・ティをセントするコよ、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・ティー値を指定してください。

プロパティ			×
님 내 내	PRTH		
プロパティ ライン テンプレート	Authentic HTML align color noshade size width U Iベント	navy 2	

以下のプロケ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
- <u>プロノティサイドバー</u>
- CSS スタイルと作業

11.7.8 テーブル

[テーブルの挿入]コマボを選択すると、テーブルの作成ダイアログが表示されます(以下のスクレーンショナを参照)。

StyleVis	sion
?	固定された数の行/カラムの静的テーブル、または入力データをベースにした動的テーブルを作成しますか?
	静的テーブル 動的テーブル キャンセル

静的なテーブルと動的なテーブルから、目的のテーブルに対応するがシ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、<u>静的</u> SPS テーブルと動的 SPS テーブルのセク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] メニューコマイドや、デザイン要素の挿入ツール・イニあるテーブルの挿入アイエン 間か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] コマドにより、静的ませば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。

以下の操作によりストを作成します

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)。

StyleVis	sion			×
?	入力データをベージ	スに <mark>静</mark> 的または動B	りテーブルを作成し	ますか?
	静的リスト	動的リスト	キャンセル	

[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。 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。

O XPath セレクター	×
XPathコンテキスト: \$XML/n1:OrgChart/n1:Office/n1:Depa	rtment
 スキーマ パス 	
スキーマの属性または要素を選択:	
On l:Name □ () n 1:Office () n 1:Name	^
·⊞ () n 1:Desc	
On 1:Location	
· ⊕ () n 1:Address	
() n 1:Phone	
💛 n 1:Fax	E
🕐 n 1:EMail	
□ () n 1:Department	
TO n liverson	- -
<	P
- ● 絶対 ×Path	
◎ ユーザー定義 XPath	
×Path の編集	(E)
OK +>>	<u>ل</u> الع

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。

	•	
	:==	
°		
°	•===	

- メモ リストを挿入する箇所にカーノルが配置されている状態で、デザイン要素の挿入 ツール・トームある [箇条書きリストを挿入] アイエンをクトックすることでも、静的なリストを作成することができます。スキーマッソトームあるノードをデザイン上へドラッグすることで も、動的なリストを作成することができます。
- ∃ 関連項目
- リストの作成
- <u>タグで囲む リスト</u>

11.7.10 ブックマーク

[ブックマーク] コマイドにより、SPS 内の任意の場所へブックマーク(おけおアンカー)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。

以下の操作によバ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、右クルクで表示されるエンテキスト 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。

ブックマークの	の挿入	
静的 動的	静的ならびに動的	
	動的	
アドレス(E):	generate-id(n1:Name)	
	XPath を編集	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。
	🔲 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
		OK キャンセル

3. 「<u>ブックマークの挿入」ダイアログ</u> にて、ブックマークの名前を静的なものにするのか(静的)、XML ドキュメトから動的に取得され るのか(動的)、おけば静的ない いっと動的ない いっから構成されるのか(静的ならび)に動的) により、対応するタブを選択します。 上のスクレージショナ 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。 ブックマークか定義されます。

メモ デザイノニて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 ヘンスタンス要素内部にブックマークが作成されます。静的な名前か与えられた場合、これらブックマーク全て15同じ名前か与えられ 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 のスクリーンショナを参照)。ブックマークの参照を行うには、同じIDをノメイパーリンクのhref値として生成することができます。こ の場合、generate-id() 関数の前にプラグメト識別子の#を使用するようにしてたたい。XPath条件式は concat('#', generate-id(nodeXXX)) どれます。

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リックして、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・ティ サイドバーのブックマーク グループ以下にあるブックマーク名プロ・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。

- 三 関連項目
- <u>ブックマークの挿入</u>
- 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] コマボにより、出カドキュメト(HTML)内にある任意の場所から、出カドキュメント内にあるアンカー、ませよ外部 ドキュメントヤドキュメント プブメント へのレンを挿入することができます。

以下の操作により、イル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トの周りに作成することができるほか、ドキュメ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)レク内部のテキストを挿入すること」されます)。ハイパーレンクへ変更する SPS コンポーネト やテキスト フラグメートを選択するか、リングを挿入する箇所へカーノルを移動してくたさい。
- 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たさい。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) へのレンクを作成する場合、# シンボルを追加する必要があります。 ハイパートレクの 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「ハイパーレンク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をスキーマとて使用している場合、ハイパーレ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、次を参照してくたさい、ハースされていないエンティティ URIの使用を参照してくたさい。

ハイパーレンクの編集

ハイパーレンクの作成後には href の値を編集することができます。ハイパーレンクを右クリックして、コンテキスト メニューから[URL の編集] を選択してください。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。

ハイパーレンクの削除

ハイパーレクカ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 サンティティ URI
- <u>ブッケマークの挿入</u>

11.7.12 条件、出力ベースの条件

[条件] コマイドにより、カーノル位置おけば選択対象の周りに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さいます。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、ノードのコンテンツがStopという文字列の場合にはメードのコンテンツを赤色に、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。

以下の操作によ、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レポーネントを選択します。
- 2. メニューコマドの[挿入 | 条件] を選択するか、右クルクにとし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)。

♪ XPath 条件式の編集 XPath 2 条件式: count(newsitem)				
様発 スキーマ属性や要素の選択 選択: \$XML/presswatch/newsitems 躍 ソース 田田名前空間 日日\$\$XML(3-1>) 日レルート要素 日レノルート要素 日レク presswatch 田() newsitems 田() newsitem 田() team	演算子/式の選択 田 地 田 比較 田 全 保 4 日 し 和 田 し 和 で 田 点 和 里 し い む い で 田 点 和 理 田 レ 一 べ に い の こ い 本 代 ー の に い で い 一 、 に い の れ げ に で し こ の れ い に で し 、 の れ い に で し 、 の 、 い っ い こ い っ い こ い っ い こ い っ い こ い っ い こ い う い っ っ い っ い っ い っ い っ い っ っ っ い っ っ い っ い っ い っ い っ っ い っ い っ っ っ っ い っ い っ っ っ っ っ っ っ っ い っ っ っ っ っ っ っ っ っ っ		 関数の選択 田 Accessors 田 Boolean ロンストラクター ロンテキスト 田 アテキスト 田 子の人 田 日付と時間 田 日 田 日 七の他 田 ノード 田 QName 田 シーケンス 田 文字列 田 URI 	
一 絶対 XPath (A)	4	F	4	Þ
Javaの使用 .Netの使用			OK(0)	*#2767P(C)

構築されている条件式で使用されるコンテキスト ノードが 左端のペインはあるスキーマッソートこで イライトされます。テキスト ボックスに 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ルクすることで、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ない場合)、おさは絶対的([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、プロノティサイドバーから[条件分岐|条件トリガー]を選択し、XPath アイテムの編集ポタン・・・・ をクトックしてくたさい。XPath 条件式の編集ダイプログが表示され、条件式を編集することができます(上のスクノーシンコントを参照)。

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、目的の分岐を右クハクして、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。

- 三 関連項目
- 条件

11.7.13 テンプレート

[テンプレート] コマナにより、カーノルの挿入位置に選択されたスキーマツ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 作によテンプレートを挿入てきます

- 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)。

XPathコンテキスト: \$XML/data	
スキーマの属性または要素を選択:	
 ジース ロンルート要素 ロンルート要素 コーン・ト要素 コーン・ト要素 コーン・トマート コーン・トート ローン・トート ローン・トー・ ローン・トート ローン・ ローン・	*
4	
■ 絶対 XPath	

- 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 ー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。

選択されたノードに対する空のテンプレートが作成されます(下のスクノーンショントでは 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
- <u>テキストとして XML コンテン を挿入</u>
- 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] コマイにより、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ドをクリンするとXPath式の編集ダイアログか表示されます。
- 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。

詳細にないては、次のセクションを参照してくたさい、SPS ファイル・コンテンソ ユーザー定義テンプレート、

- 三 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
- <u>挿入| デンプレー</u>
- 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 され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」 ますは タグで囲む | 変数テンプレート」 を選択することで、カーソル位置に、 SPS で定義される変数のテンプレート か挿入されま 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 オルダか 含まれており、 これにより変数の値が出力されます(以下のスクリージンョナ を参照)。 変数テンプレートには、 必要ににじてその他の(静的ならび) ゴンテングを含めることができる ほか、 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。

以下の操作によ変数テンプレートを挿入することができます:

- 1. 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カジ挿入 | 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 リーンショントを参照)。

変数テンプレートを挿入		
新規テンプレートに対する変数を	選択してください:	
\$name (読み取り専用)		
	ОК	キャンセル

- 3. ダイアログには、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>か表示されます。変数テンプレートに対して追加する変数を選択してくたさい。
- 4. OK をクルクして完了します。
- 🖃 関連項目
- <u>挿入 デンプレート</u>
- SPS ファイル: エンテンソ 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] コマドロよ デザイノニ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 ニューニ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
- <u>デザインプラグメント</u>

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] コマドにより、デザイン内部の任意の場所にレイアナ コンテナーを挿入することができます。レイアウト コンテナーイコ おイアナ ボックスやラインを挿入することができる まか、これらの挿入コマンドはレイアナ コンテナーの選択中に有効になります。

<u>デザイン要素の挿入 ツール・</u>に含まれている対応するアイコイことり、レイア・オコンテナー、レイア・オ・ガクス そしてラインを挿入することができます。ツール・ーアイコンにとり挿入を行うコよ 目的のソール・ーアイコンをクリックして、レイア・オ 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たさい、レイアナ モジュール。

三 関連項目

- レイアウト モジュール
- ツール (一) 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ノ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、それそれのコマドの詳細を確認することができます。

- <u>目次の挿入</u>
- <u>目次ブックマーク</u>
- <u>目次ブックマーク(ウィザード)</u>
- <u>目次参照</u>
- <u>
 日次参照</u>
 <u>
 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
- <u>L/VL</u>

- レバル参照
- 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

• <u>目次(TOC)</u>

11.7.19 新規ドキュメント

[新規ドキュメントの挿入] コマドにより、新規ドキュメント テンプレート かかーノル挿入位置に挿入されます(以下のスクレーンショントを参照)。

新規ドキュメント 'Document1'	
イニシャルドキュメント セクション プロバライ	を編集 ヘッダー・フッターを追加
く新規ドキュメント 'Document1'	

新規ドキュメト テンプレートには、空のイニシャルドキュメト セグションか含まれ、エンテングをイニシャルドキュメト セグションへ入力することができるよう プルます。必要な場合には、「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] コマイを使用することで、ドキュメト セグションをイニシャルドキュメト セグ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から構成されることになります。

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だれ、<u>複数のドキュメーの出力</u>。

- 旦 関連項目
- 複数のドキュメトの出力
- <u>新規ドキュメトテンプレートの挿入</u>
- 新規ドキュメト テノプレートとデザイン構造
- · ドキュメト プロノティとスタイル

11.7.20 ユーザー定義アイテム

入力 | ユーザー定義 アイテムコマボ ヘマウスオーバーすることで、ユーザー定義要素 おけはユーザー定義 XML テキストプロック を挿入 するすっかのコマボ が表示されます。これらコレポーネントの使用方法については、<u>SPS ファイル・コンテンツ | ユーザー定義要素、XML テキ</u> <u>ストプロック</u>のセグションを参照ください。

□ 関連項目

Altova StyleVision 2021 Basic Edition

- <u>ユーザ</u>-定義要素
- <u>タゲ(囲む) ユーザー定義要素</u>
 <u>ユーザー定義 XML テキストブロック</u>
- <u>ユーザー定義テンプレート</u>

11.8 タグで囲むメニュー

[タグで囲む] メニューコよ デザイントの選択対象を様々なデザインコンポーネントにお囲むすめのコマンドが収められています。これらコ マドの、くつかは、コンポーネントをデザインへ挿入するすみの、ツール、デアイコンからもアクセスすることができます(同様のコマンドは、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] コマンドは、SPS デザインにて選択対象を右クル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 ロマンドは、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を構築するためのコマイであるため、デザインビューでのみ利用可能となります。

- □ 関連項目
- エンテンソの編集の手順
- <u>ツールド</u>

11.8.1 テンプレート

[タグで囲む | テンプレート] コマボにより、選択されたデザインコンポーネントやテキストが、選択されたスキーマッリーノードにより囲まれます。以下の操作を行なってくたさい

- 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)。

◎ スキーマセレクター	x
XPathコンテキスト: \$XML/n1:OrgChart	
スキーマの属性または要素を選択:	
● シース □ □ ♪ \$XML (メイン) □ □ / ルート要素 □ □ () n1:OrgChart □ ① n1:CompanyLogo ○ n1:Name □ ① n1:Office	*
	Ŧ
☐ 銫牡 ×Path	
OK +>>±	

- 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 ます。
- [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。

三 関連項目

- テキストとてXML エノテノンを挿入
- 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] コマイにより、選択対象をXPath 条件式によりユーザーが指定したノードに対するテ 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。
- 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エレテキス 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] をクルクして完了します。

ターゲ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。

は関する詳しい説明は、次のセクションを参照してください、SPS ファイル構造 テノプレートとデザイノファグメオ | 変数テンプレート.

目 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
- <u>デザイン要素の挿入</u>
- 挿入 デノプレート
- 挿入 ユーザー定義テンプレート

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。変数テレプレートダイアログには囲む 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。

選択対象の周りこ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、次のセクションを参照してください、SPS ファイル構造 テノプレートとデザインアグメト | 変数テンプレート.

□ 関連項目

- 挿入 デプレート
- 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] コマド 🔊 によりHTML の段落 要素が選択されたエ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、おけは(タグをクリックすることで)ノード全体を選択することで、エ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に配置されている場合、段落要素(の開始ならびに終了タグ)がその位置に挿入されます。デザイン要素の挿入 ツール・トームある 段落の挿入 アイエンを使っても挿入することができます

[特殊な段落] コマボでは、選択されたノードに定義済みのフォーマナ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 ーリレ ~のコンボボックスからも選択することができます。

各種段落にはそれぞれフォーマトか与えられています。pre フォーマトでは、改行か空白スペースに正規化されることは、そのまま改行として出力されることに注意してくたさい。

三 関連項目

- 定義済みのフォーマト
- <u>CSS スタイルと作業</u>

11.8.5 リスト

[タグで囲む | リスト] コマドにより、選択対象の周りに静的なリストとリスト アイテムか作成されます。選択対象に(Enter キーを押下することで挿入される) 改行か管まれている場合、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	
箇条書き 番号付き箇条	書き
	•
	· · · · · · · · · · · · · · · · · · ·
	•
。	
°	
o	•
L	
	UK キャンセル

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、デザイン要素の挿入ツールレーにあるリストアイエ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。
- 三 関連項目
- リストの作成
- <u>タグで囲む| リスト</u>

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] ならびに[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コンポーネト が選択された状態で有効」ないます。これらのコマドにより、ブックマークと イパーレン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 イパーレンクの処理と、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
- ハイパーレクの定義

11.8.7 条件、出力ベースの条件

[条件] コマイドにより、カーノル位置おけば選択対象の周りに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さいます。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、ノードのコンテンツがStopという文字列の場合にはメードのコンテンツを赤色に、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。

以下の操作によ、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レポーネトを選択します。
- 2. メニューコマンドの[挿入 | 条件] を選択するか、右クトックにとし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)。

XPath 条件式の編集 XPath 2条件式: count(newsitem)				×
構築 評価 スキーマ属性や要素の選択 選択: \$XML/presswatch/newsitems 型 ソース 田 図 名前空間 日 ○ 名前空間 日 ○ \$XML (メイン) 日 ✓ ルート要素 日 ○ presswatch 田 ⑦ presswatch 田 ⑦ newsitems 田 ⑦ newsitem 田 ⑦ team	演算子/式の選択 田軸 田比較 田条件 田Quantified/for 田論理 田ノード 田少一ド 田数値 田シーケンス 田 Type		関数の選択 田 Accessors 田 Boolean 田 コンストラクター 田 コンテキスト 田 日付と時間 田 日 モの他 田 ノード 田 秋値 田 QName 田 シーケンス 田 シーケンス 田 文字列 田 URI	
── 絶対 XPath (A)	<	F	4	Þ
Javaの使用Netの使用			OK(0)	キャンセル(C)

構築されている条件式で使用されるコンテキスト ノードが 左端のペインはあるスキーマッソートこで イライトされます。テキスト ボックスに 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ルクすることで、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ては、場合)、 おけは絶対的 ([絶対 XPath] チェックボックスにチェックが入っていては、場合)、 おけは絶対的 ([絶対 XPath] チェックボックス にチェックが入っていては、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インで分岐を選択し、プロノティサイドバーから[条件分岐|条件トリガー]を選択し、XPath アイテムの編集ポタン・・・・ をクトックしてくたさい。XPath 条件式の編集ダイプログが表示され、条件式を編集することができます(上のスクノーシショナを参照)。

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、目的の分岐を右クルクして、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
- <u>条件</u>

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。目次コマンドのノストを以下に示します。リンクをクリックすることで、その目次コンポーネントのセグションから使用方法の詳細を確認することができます。

- <u>目次ブックマーク</u>
- <u>目次ブックマーク(ウィザード)</u>
- <u>الأسما</u>
- レベル参照
- 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イテンプレートにより囲まれます(以下のスクレーンショナを参照)。

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 プロパテで編集 ヘッダー/フッターを追加
Table of Contents: Chapters and Their Sections
<u>[®]div</u> / <u>[™]Ψ'MyTOC'</u> / (num-lvl): (テキスト参照)()(ページ参照) / <u>[™]Ψ'MyTOC'</u> / <u>[®]div</u>
「 \$div) ^ペ ψ'MvTOC' (num-lvl)) (テキスト参昭)()(ページ发昭)< ^ペ ψ'MvTOC' < \$div
く 新規ドキュメント 'Document 1'

新規ドキュメト テンプレートはオイニシャルドキュメト セクションが含まれており、内部はオコマイが選択された際の選択対象が含まれます。上のスクレージョナでは、目次デザインコンポーネントが新規ドキュメト デンプレートにお囲まれおり、イニシャルドキュメト セクションの中にコンテンン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] コマド によりドキュメント セクションをイニシャルドキュメト セク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トから構成されるようことがます。

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クションを参照くたさい。

🖃 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 テノプレートとデザイン構造
- · ドキュメント プロノティとスタイル

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<u>ユーザー定義要素</u> が作成されます。ユーザー定 義要素の使用方法については「 <u>SPS ファイル コンテンソ| ユーザー定義要素</u>のセグションを参照ください。

□ 関連項目

- <u>ユーザー定義要素</u>
- <u>タグで囲む|ユーザー定義要素</u>
- 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

テーブル メニュー 11.9

[テーブル] メニューイコよ、静的まけは動的なテーブルを挿入し、それらテーブルの構造やプロ・ティを変更するナメのコマンドが含まれていま 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、セルのプロ・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 プロパティーには定義することができます。

テーブルコマンドは「テーブル」メニューや、テーブルソール、トイニあるアイコンからアクセスすることができます。カーノルの位置により利用でき るテーブルコマイカ変化します。テーブルの挿入コマイをクリックすることで、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、おコおうにカーノルを移動させ、目的の編集コマトを選択してくたさい、フォーマト、プロ ノティを編集するコム、適切なセル、列、行、おコムテーブルコンーノルを移動させ、スタイルサイドバーオコムプロノティサイドバーコンテーブル コポーネトに対するプロノティを定義してくたさい。

テーブルメニューでは以下のコマイを使用できます

- <u>テーブルの挿入、テーブルの削除</u> •
- テーブルヘッダーノフィターの追加 •
- 行/列の追加と挿入 •
- 行/列の削除 •
- 左、右、下、上、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

~ダーとフター

動的なテーブルを作成した場合、ヘッダーやフッターをテーブルコ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、それそれへッダーとフッタ ーを作成することができます。静的ならいに動的なテーブル内にあるへ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 キーや矢印キーにより、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

セルコおらゆる種類の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ントは標準的なフォーマント ツールによど行わ れます。

三 関連項目

11.9.1

- <u>テ</u>−ブルソ−ル ←
- <u>テーブルの作業</u>

テーブルの插入、テーブルの削除

テーブルの挿入コマボ Ш により、空のテーブルかデザインへ挿入されます。このコマボを選択することで、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、動的テーブルの作成を参照くたさい。

テーブルの構造は、行や列を追加、挿入、削除することことり、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 () により、カーノルが配置されている静的ませる動的なテーブルが削除されます。



- <u>テ</u>−ブルソ−ル ←
- 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)ヘッダー行ませま(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(マークアップとして 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あります)。同様に、テーブルフッターは(テーブルボディー下部の)フッター行ませま(テ ーブルボディー右側の)フッター列として表示されます(マークアップとしては、フッターがテーブルボディー内部に含まれることもあります)。

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)。

- **「三」 テーブル 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。
- デーブル 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。
- アーブルフッター行の追加: テーブルボディーの下にフッター行を追加します。

🗉 関連項目

- <u>テ−ブルソ−ル・</u>
- <u>テーブルとの作業</u>

11.9.3 行/列の追加と挿入

[行の追加]コマド 出により、静的おけばか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 聞により、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]コマド ()により、静的まけ、動的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 問題により、カーリ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。.



11.9.4 行/列の削除

[行の削除] コマボ 🕮 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、動的なテーブルの 両方に交して適用されます。

[列の削除] コマド 💼 により、カーノルは配置されているテーブルの列が削除されます。このコマドは、静的ならび 動的なテーブルの 両方に対して適用されます。

- □ 関連項目
- <u>テ</u>−ブルソ−ル ←
- <u>テーブルの作業</u>

11.9.5 左/右/下/上のセルを結合

[左のセルを結合] コマド 調により、カーノルが配置されているセルと、その左にあるセルが結合されます。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。このエマドに請納がならびに動的なテ 一ブルで使用することができます。

[右のセルを結合] コマド 協 により、カーノル的配置されているセルと、その右にあるセルが結合されます。両方のセルのエンテンパ 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。このコマドに講筆がならびに動的な 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 図 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。このコマドは諸約ならびに動的なテ 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 凹 により、カーノル的配置されているセレと、その上にあるセルが結合されます。両方のセルのエンテンが 連結されて新しいセルニ表示されます。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。このコマン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。

- 三 関連項目
- <u>¬</u>-ブルソ-ル[×]-
- 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 いっしょう かーノルが配置されているセルの右隣に新たなセルが作成されます。オメジナルのセル内にあるエンテンソは変更されません。オメジナルのセルニて定義されたすべてのプロ、ティイは、新たなセルニモニーされます。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。

[セル上下に分割] コマボ 闘 により、カーリル 配置されている セルの下に新たな セルド 作成されます。 オバンナルの セルタル あるコンテンソ お変更されません。 オバンナルの セル こて定義されたすべての プロ ティは、 新たな セル こもユピーされます。 このコマド は 新的ならび し動的なテーブル て利用する ことができます。

- 三 関連項目
- テーブルソール ~
- <u>テーブルとの作業</u>

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ク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より、テーブルの構造を理解するのこ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別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>
- <u>テーブルの作業</u>

11.9.8 テーブル プロパティ

2

[テーブル プロパティ] コマドは、<u>静的おけ調的 テーブル</u>内部にカーノル的配置されている時に有効になります。コマドをクリック することで、テーブルコンポーネントが選択されたプロ ティサイドバーカ表示されます(以下のスクノーシンタントを参照)。

	PRTH		
プロバティ	周性	10	
トーノル	HIML		
	i align		_
	bgcolor		I 😳
	border	1	1.000
	cellpadding	5	
	cellspacing	0	
	datafld		
	dataformatas		.
	datasrc		
	frame		
	height		
	rules		.
	summary		
	width	100%	

テーブルのプロ、ティを編集して、[OK] ボタイこと完了します。

□ 関連項目

- <u>テーブルソール ~</u>
- <u>テーブルの作業</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 ドキュメトのデーダ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クアップが XML ドキュメトから直接得ら れるほか、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。

このコマンドを選択することで、CALS/HTML テーブルの編集ダイアログが表示されます(以下のスクレーンショントを参照)。

	L テーブルの編集	- X-
	- //////L#%:	×
テーブルの型	メイン要素	
CALS	table	
CALS	informaltable	
HTML	table	
HTML	informaltable	

CALS/HTML テーブルに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

- 三 関連項目
- <u>テーブルの作業</u>
- <u>静的テーブルの作成</u>
- <u>動的テーブルの作成</u>

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ン高さ配置を行うコマンドアイコンが収められています。セルカーン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: セルコンテンパ高さをセル上部に揃えます。

¥

中央揃え: セルコンテン/高さをセル中央に揃えます。



下揃え: セルのコンテンツ高さをセル下部に揃えます。
- ∃ 関連項目
- <u> __ブルソ_ル 〜</u>
- <u>テーブルとの作業</u>

11.10 プロパティメニュー

[プロパティ] メニューゴは 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ドが収められています。以下のサブセクションではそれそれのコマ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。
- <u>定義済みのフォーマナ 文字列</u>コマド: 与えられた SPS I マルビン教値のデータ型を定義します。
- 三 関連項目
- <u>プロノティサイドバー</u>

11.10.1 リストの編集

[リスト] コマイドにより、リストカサーノル位置に挿入されます。コマイドを選択すると、リストのスタイルを選択するさめの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)。数値を伴うリストの場合、最初に表示される番号を指定することもできます。

リスト				-X
箇条書き	番号付き箇条書き]		
		•		
		•		
	<u>.</u>		ОК	キャンセル

- 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、おけまdurationのデータ型にお定義された(コンテンツ)プレースホルダ、入力フィールド、自動計算には、 値フォーマナ ダイアログでは、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、おけま 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。

ドロップダウノリスト内で使用することのできる定義済みのフォーマトには次の2種類があります。

- 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。ユーザーか定義済みの値フォ ーマトを作成すると、(アプリケーション全体にではない)現在開かれている SPS に対してこれらのフォーマトか作成されます。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か反映されま す。

定義済みの値フォーマト文字列を作成する

定義済みの植フォーマト文字列にはデータ型が与えられます。以下の操作によに定義済みの植フォーマト文字列を作成することができます

1. [プロパティ | 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

 定義済み値フォーマット文字列
定義済みのフォーマット: date 💌
ОК +>>セル(C)

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、追加おける挿入アイコンをクリックします。フォーマト文字列の編集ダ イアログが表示されます。

フォーマット文字列の編集	X
	ОК
書式文字列が空です。	キャンセル
< フィールドの挿入(I) > フィールドオブション(F) エラーへ移動(E)	

コンボボックスの矢印をクリックすると、そのデータ型に対するStyleVisionから得られた定義済みフォーマトのノストか表示されます(以下のスクノーンショナを参照)。

フォーマット文字列の編集	
	ОК
MM / DD / ነነነነ MM - DD - ነነነነ DD / MM / ነነነነ	キャンセンル
D. MM. YYYY YYY - MM - DD	

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ほか、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 を定義するさめの構文については、値フォーマト のセクションを参照ください。[フィールドの挿入] まさは[フィールドオ プション] ボタイこと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]をクトックして SPS ファイルを保存してください。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 ダイアログのオプション内に表示されるよう ています。

灹

- 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序により、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。
- カスタムフォーマト文字列を削除するコよ、定義済み値フォーマト文字列ダイアログにてそのエトリーを選択し、削除アイエンを クトックしてくたさい。

□ 関連項目

• <u>値のフォーマト(数値データ型のフォーマト)</u>

11.11 ツールメニュー

[ツール] メニューイコはスペルチェックオブノョンと、StyleVisionをカスタマイズするためのオブノョンが収められています。そ

ツールメニューのサブメニューイコお以下のコマンドが収められています

- <u>スペルチェック</u>
- <u>スペルチェックのオプション</u>
- <u>カスタム化</u>
- 🖃 関連項目
- <u>StyleVision のセットアップ</u>

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。スペルチェッカーに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たきい)。

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、おはBSD ライセスという商用ソス・ウェアにおおる再配布が認められて、るライセンス下で利用可能からう かに大きく依存しておます。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、お制約の多いライセス下で 配布されて、ます。これらの辞書は、http://www.altova.com/dictionaries。からダ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。

このコマイドをクリックすることで、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、ドキュメン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。

EMail	無視(!)
え 補:	すべて無視(G)
5 Mail Email 5M ail 5M-ail	辞書(3追加(A)
vlail Emails Em ail	変更(C)
Em-ail Emil	すべて変更(L)

[候補] ペイム表示されたエトリーを選択し、[変更] おけま[すべて変更] をクリックすることで、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かぞ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)。おた、[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「おか、[すべて無視] ボタイこより、ドキュメト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] ボタイこより、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してより、辞書に存在してよい次の単語が表示されます。[ドキュメントの再 チェック] ボタイこより、ドキュメントを最初からチェックすることができる「おか、[閉じる] ボタイこよりダイアログを閉じることができます。

オプション ポタイは 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。

- 三 関連項目
- <u>スペルチェックオプション</u>

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] コマドにより、スペルチェックのオプションを指定するオンタ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)。

スペル チェックのオプション			
スペルチェック ⑦ 常に修正を表示 ③ メイン辞書だけを使った修正を行う ⑦ 大文字で書かれた単語を無視 ③ 数値がある単語を無視 ⑦ キャメルケースの単語を分離 ユーザー辞書(D)			
辞書の言語 English (US)			
OK キャンセル			

常に修正を表示:

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。

メイン辞書だき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 タンも無効化され、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か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か無視されます。

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します。例えば "CamelCase" とう単語では、 "Case" の "C" が 大文字で示されているすっか、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ずい サっか、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からしむまる。 やりょう離して、 個々の、 やいをチェックすることによの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て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) どなっています。 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しりさうけます。

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、う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されます。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されます: C: \ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons.

Lexicons フォルダー内部は、異なる言語に対応した辞書のファイルが、く言語名>W、辞書ファイル>とう構成で各フォルダーは収められます。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 C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US)\en_US.dic C:\ProgramData\Altova\SharedBetweenVersions\SpellChecker\Lexicons\English (US) \en_US.dic

スペルチェッカーオプションダイアロゲでは、「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。 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ビナか64ビナかにかかわらず)複数のAltova 製品に お供有されます。

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 <u>http://wiki.services.openoffice.org/wiki/Dictionaries</u>や <u>http://extensions.services.openoffice.org/en/dictionaries</u>.といた場所かダウレードすることができます (OpenOffice ではOXT とら拡張子のZIP アーカイブが配布されているため、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こピーしてくたさい。Hunspell 辞書はMyspell 辞書をベースしているため、Myspell 辞書を使用することもできます)
- 複数の言語辞書をエ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ニあるレノ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(以下のスクレーシショントを参照)。

辞書の言語	
English (US)	•

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、そしてお使いのコンピューターにおお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 ユーザーこと時可された単語を収めるナダの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リストロネル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、封当り除することができます(以下のスクリーンショナを参照)。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」 ボタンをクトックすることで、この辞書へアクセスすることができます(このセクションの2番目のスクリーンショナを参照してください)。

	×
単語:	
Nanonull	追加
辞書:	
xml	削除
Nanonull	
stylesheet	
	OK
-	+
· · · · · · · · · · · · · · · · · · ·	47721

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」 ポタンをクルクしてくたさい。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て選択し、「削除」 ポタンをクルクしてくたさ 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コよ「OK」をクルクすることで、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オンがのプロンプトをポップアップします。「辞書に追加」ボタンをクリックすると、未知の単語がユーザー辞書へ追加されます。

ユーザ 辞書は以下の場所にみます: C:\Users\<user>\Documents\Altova\SpellChecker\Lexicons\user.dic

□ 関連項目

• <u>スペルチェック</u>

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」つうして Style Vision をカスタマイズすることができます。

<u>コマンド タブ</u>

カスタマイズダイアログの[コマンド] タブでは、個々のコマイドをメニュー・バーやソール・トー記置することができます。

Customize		
Commands Toolbars Keyboard Menu Options Categories: Commands: Project		
Description: Arrange windows so they overlap Close		

メニュー・バーやソール・イー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 たコキソール・イ・デッグしてくたさい。コマボを配置することができる場所にはより表示され、ドロップすることでコマボか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エンが存在する場合、ツール・イニ はそのアイエムを示され、そうでない場合にはデキスト形式のコマボが作成されます。メニュー・バーまけはソール・イーマボを追加すると、 コマボを右クリ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、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。

<u>×</u>E

- 上記のカスタマイズはアプリケーションス対して適用され、StyleVision にてドキュメントが開かれているかどうかにかかわらず適用されます。
- Style Vision がインストールされたときの状態にメニューやソール、それた水するコよ、ツール、ータブを選択し、[リセット] おっよ[す べてリセット] ポタンをクトックしてくたさい。

ツール・ケブ

[ツールバー] ダブでは、特定のソール ~を有効化ませま無効化して、ツール ~ 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 ~ をインストール時の状態 へ たと することができます。

Customize	—
Commands Toolbars Keyboard Menu Opti Toolbars: Design Filter Format Global Resources Insert Design Elements Menu Bar RichEdit Standard	Reset Reset All New Rename
Table	Delete Show text labels Close

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・ー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、コマドダブが開かれている状態でソール・ーに追加することができます。ツール・ーをギラッグすることで、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・ーのタイトルバーをダブルクトックすることで ツール・ーのド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・ータブでは、チェックボックスをクトックすることで、ツール・ーを表示/隠すことができます。ツール・ーを(ツール バータブこで)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・ーに含まれているアイテムのテキスト ラベルが表示されます。[リセット] ボタンをクトックすることで、選択されたツール・ーを Style Vision がインストールされた際の状態へルセナ することもできます。[すべてリセット] ボタンをクトックすると、すべてのソール・ーとメニュー・バーがルセナされます。

メニューバー、関する注意点

メニュー・コーマードを追加し、アイテムをメニュー・トカン削除することができます:上記のコマードを参照してくたさい。Style Vision のイン ストール時の状態にメニュー・トをノルナするゴよ、カスタマイズダイアログのソール・トタブ内のメニュー・「を選択し、リセット、ポタンをクトックします。(すべてをリセットするポタンをクトックするとソール・トをノルナします)。

キーボード ダブ

[キーボード] ダブでは、任意のStyleVision コマンドレストレーボード ショートカナトを定義おける変更することができます。

スタマイス コマンド ツールバー キーボード	メニュー オプション	
カテゴリー(C): ファイル(F) コマンド(O): 印刷プレビュー(W) 再ロード(D) 終了(A)	設定するアクセラレータ(F): デフォルト ▼ 現在のキー(U): Ctrl+O	 割り当て(A) 削除(R)
新規作成 (空)(E) 生成された FO ファイルの係 生成された HTMI ファイルの く ト 説明: 既存のドキュメントを開きます	新規ショートカット キー(N): Ctrl+C 割り当て先: コピー(C)	ৰ্বন্যেদ্রুম(S)
		<u>閉じる</u>

<u>コマボショートカナをコマボへ割り当てる</u>

- 1. カテゴノーコンボボックスから目的のコマンドかあるカテゴリーを選択します。
- 2. コマボリストボックスにでショートカナを割り当てるコマボを選択します。
- 3. 「新規ショートカナキー」入力フィールドをクリックして、コマンドを起動するさめのショートカナキーを押下します。「新規ショートカナオー」入力フィールドにショートカナカ表示されます。入力されナショートカナガ他のコマンドへ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コマンド名が表示されます(例えば上のスクノーンショナでは、Ctrl+Cが「コピー」コマンドへ既に割り当てられているさめ、ファイルを開くコマンド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には、Ctrl、Alt、おさはShiftキーのいずれのを押下してくたさい。
- 4. [割り当て] ボタンをクリックすることで、入力されたショートカトがコマンドに対して割り当てられます。ショートカトが「現在のキー」・リスト ボックスに表示されます。

ショートカトの割り当てを変更(また」は削除)する

- 1. 削除するショートカナカ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 こてクトックします。
- 3. 選択可能になった[削除] ボタンをクトックします。

<u>キーボード の割り当てをすべてルセナする</u>

- 1. [すべてリセット] ポタンをクトックすることで、インストール時のオトジナルショートカナへ割り当てがしたオされます。すべてのキ ーボード割り当てを本当にしたよ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ト割り当てがしてかったれます。

<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。 メニューバーゴよ、 ドキュメント 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 ドキュメント が開かれている時に表示される SPS メニューかあります。

カスタマイズ	×
コマンド ツールバー キーボード メニュー オフ	プション
アプリケーション フレーム メニュー:	ביביבי
表示するメニュー(S):	コンテキスト メニューの表示(C):
SPS 👻	·
リセット(R) atrova sps	リセット(E)
StyleVision Power Stylesheet	ヒント: コンテキスト メニューをクリックし、 ページを [コマンド] に変更します。次 に、ツール バーのボタンをメニュー ウイン ドウにドラッグします。
メニューのアニメーション(M): なし ・ マ メニューに彩をつける(H)	
	(閉じる)

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(スクノーンショナではSPS メニューが選択されています)。
- 2. [コマンド] ダブをクリックして、目的のコマイをメニューバー・ドラックしてください。

<u>メニューカらコマンドを削除する</u>

- 1. アプリケーションフレームメニューペインから(ドキュメントか開かれていない、状態で使用可能なメニューを表示する) デフォノレト おこ はドキュメントか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・ト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. おけばい)メニューからメニュー・バー、おけはメニューコマンドをドラッグ、おけは(い)メニューを右クトック、おけば削除」を選択します。

<u>メニューバーをしたかする</u>

- 1. アプリケーションメニューペインのコンボボックス内で地大するメニューエトリを選択します。
- 2. メニュー名の下にある[リセット] ボタンをクリックします。メニューバーをルセンしてもよめ確認するプロンプトが表示されます。

<u>コンテキストメニューをカスタマイズする(右クトックメニュー)</u>

- 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] ダブをクリックして、表示されたコンテキスト メニューヘコマンドをデラッグします。

コンテキストメニューカ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。おけよ 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。

<u>コンテキスト メニューをリセットする</u>

- 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] ボタンをクリックします。リセットしてもよめ確認するプロンプトが表示されます。

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トックするか、

• カスタマイズダイアログボックスの[閉じる] ボタンをクトックしてくたさい。

<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タンイこより、メニューカウトックされた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イあるドロップダウンリストから、目的のオプタンを選択してください。

<u>メニュー/ぷをつける</u>

メニューの周リに影をつけた場合にこのオプションを選択してください。すべてのメニューイ、影が加えられます。

オプションタブ

[オプション] タブでは、ツール・トーニ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 ~1:画面のと トを表示] オプションを有効にすることで、ツール ~ アイテムのと ト か表示されます。 画面のと ト オプ・タンコンコよ、 ショート カメ も画面に表示するかを指定するナメのサブオプションが与えられています。

- 三 関連項目
- <u>ツールド</u>
- <u>ビュー| ツールド</u>
- <u>ユーザーイノターフェイス</u>

11.11.4 ツールバーとウィンドウの復元

このコマイドにより、ツールレー、ウイイドウ、入力ヘルレー、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。変更を反映するこは、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 | ツールドー
- <u>ユーザーインターフェイス</u>

11.11.5 オプション

[オプション] コマイドにより、HTML 出力 ファイルのエンコードを指定するため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することができます(以下のスクリーンショナを参照)。

オプション	X
デザイン スキーマ プロジェクト	デフォルトエンコード XSL オプション
┌ デザイン ビュー オプション ──	
マークアップの最大幅:	500 p×
デフォルト単位:	in 💌
レイアウト コンテナー オプション	
グリッド サイズ X:	0.10in
グリッド サイズ Y:	0.10in
デザインビューと HTML 出力に じテキストを印刷出力フォーマッ になることがあります。	てテキストが含まれるレイアウト ボックスは同 トに収めるためには追加の幅や高さが必要
以下で指定される追加の幅/8 すべてに自動的に追加されます	高さは、新たに作成されるレイアウト ボックス す。
デフォルトの追加幅:	20%
デフォルトの追加高さ	20%
未知のコンテキストノード上にテ	シブレートを作成
◎ 単一のテンプレートを作成	
◎ 絶対スキーマ パス付きのデ	ンプレートを作成
 第に確認する 	
	K キャンセル 適用(A)

以下のオプランをセナすることができます

- マークアップの(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 コンテナーのグリボ サイズ。指定された長さが、対応するグリバの軸に表示されるグリバ間の幅となります。
- 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トにてテキストのレンダレングを行うために必要な追加 長さとし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、おけは絶対長さ単位として入力することができます。
- ・コ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コンテキスト内部に電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) その名前だりたで、まけまく(ii) スキーマルートからのフル や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ご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

スキーマッノーでは、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。スキーマタブこで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(以下のスクノーシンヨナを参照)。デフォノルでは、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てなります。

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 ヘッダーの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 コマイを使用してください。

MobileTogether デザイン

SPS ファイルをMobileTogether デザイン(MTD) ファイルに変換するために下にリ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MobileTogether 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)。

- (例えば太字おはは線の要素を含む、ラグラン要素) インラインフォーマトを含む構文テキストと ラグラン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トを持つ箇所で単一のラベルに変換されます。各ラベルがも元のテキスト フラグメトレン対応するフォーマトを持つ箇所で、この種類のテキストを複数のMTD ラベルご分割するオプションを設定することが できます。
- SPS にプ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プランチを変換し、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(と全てのプランチ)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] メニューイコよ、StyleVision ウィンドウがGUI に表示される方法を指定するオメのコマン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」)。ウィンドウを最大化するコは、そのウィン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] メニューコよ Style Vision ウイドウがGUI に表示される方法を指定するオメ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」)。ウィドウを最大化するコよ そのウィドウの最大化ポタンをクトックしてくたさい。

更に、現在開かれているすべてのドキュメント名か表示されます。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リックしてく ださい。

ウィンドウダイアログ

ドキュメトのノスト下部」コは「ウイドウ」ダイアログへアクセスするオメクロマイドが用意されています。このエトリーをクリックすることで「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、マストして適用することのできるコマンドが表示されます(名前をクリックすることで、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。
- 三 関連項目
- <u>ユザイノターフェイス</u>

11.13 ヘルプメニュー

StyleVision[ヘルプ] メニューイゴは StyleVision のソレプマニュアルヘアクセスするオメのコマンドや StyleVision に関する情報 ヘアクセ スするオメクロマンド、そして Altova Web サイト上のサポート ページへのルンか収められています。 ヘルプ メニューイゴは 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。

[ヘルプ] 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:

- <u>目次、インデックス 検索</u>
- 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
- <u>他のコマボ</u>

11.13.1 目次、インデックス、検索

- ▼ 目次
 - <u>説明</u>

ヘルプウイドウの左側のペイノに目次を表示した、StyleVisionの画面上のヘルプマニュアルを開きます。目次はヘルプドキューメト全体の概要を表示しています。目次のエントリをクリ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
 - <u>説明</u>

へいプウイドウの左側のペインにキーワードインデックスを表示したStyleVisionの画面上のへいプマニュアルを開きます。目次 はヘルプド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ト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1つ以上のトピックコンクされている場合は、トピックのノストカ表示されます。

- ▼ 検索
 - <u>説明</u>

へレプシンドウの左側のペインに検索ダイアログを表示したStyleVisionの画面上のヘレプマニュアルを開きます。単語を検索 するコよ、入力フィールドに検索対象を入力して、(i)「Return」を押す、おけばii)「トピックのリスト」をクリックします。を 押します。ヘレプシステムは、ヘレプドキュメント全体で全文検索を行いたメーレたしたりストを返します。アイテムを表示するためはは アイテムをダブルクリックします。

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

- ▼ ソフトウェアのライセンス認証
 - <u>説明</u>

Altova 製品ソストウェアをダウンロードすると、無料評価キーおけは購入されたライセンスキーを使用して、製品にライセンを供与、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。この手順には数分を要します。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 やを取りま す。ライセンスファイルを参照 おさはライセンスファイルへの やを入力し「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 やを入力する ダイアログに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リゴよ泳続的なライセンスキーを購入するよかの対象が 含まれて、ます。この対象をクリックすると、製品の泳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」で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 います。3つの種類の泳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 は単一のエビューター上のノストウェアのロックを解除します。インストール済みのラ イセンスをN 台のエビューターのナダゴーサートのノストウェアのロックを解除します。インストール済みのラ イセンスをN 台のエビューターのナダゴ、購入するとN 台までのエビューター上でノス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 人のユーザーゴ同時にノストウェアを使用することができま す。(10N 台のエレビューター(コノス・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ち台の異なるエビューター上でノストウェアを使用することを許可します。ソストウェアのライ センスは特定のユーザーがち台の異なるエビューター上でノストウェアを使用することを許可します。ソストウェアのライ センスを認証するココよ「新規のライセンスをアップロードする」をクリックして表示されるダイアログトカでライセンスファイ ルを参照、オゴムライセンスファイルの、父を入力し「OK」をクリックします。
- 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ンライセンスア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つ行ってくたさい

- 適切な場所にライセスファイル(.altova_licenses)を保存し、ライセスファイルをダブルクリックし、表示されるダイアログロ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、びを入力し「OK」をクトックします。
- 適切な場所にライセスアイル(.altova_licenses)を保存し、Altova LicenseServer のライセスプールニア・プロードします。以下を行うことができます:(i)製品のノストウェアライセス認 証ダイアログからAltova製品からライセンスを取得します。(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、下記のトピッグを参照してくたさい。

ソフトウェアライセンス認証ダイアログ(アのスクリーンショット)は「ヘルプ | ソフトウェア アクティベーション」をクリックすること 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ア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、ライセンスアイルを参照し選択します。「OK」をクトックしてライセンスアイルへのやを確認します、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 License Server を使用してライセンス供与する方法:ネットワーク上のAltova License Server を使用してライセンスを取得するゴネノストウェアのライセンスの認証ダイアログの下にある Altova License Server を使用するをクリックします。使用する License Server がインストールされているマシンを選択 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プライアトマンと同じサブネト上に存在する必要かあります。自動検知が作動しない場合、サーバーの名前を入力します。Altova LicenseServer はライセンスプール内でAltova 製品のためのライセンを有している必要かあります。LicenseServer 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で表示されます(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」をクリックしてライセンを取得します。

Altov	/a XMLSpy En	terprise	dition 2020 Software Activation		
Than licens Licen	k you for choo se or select an seServer or a	sing Alto Altova L valid lice	a XMLSpy Enterprise Edition 2020 and welcome to the software ac enseServer which provides a license for you. (NOTE: To use this so se from Altova.)	ivation process. You can view your assigned oftware you must be licensed via Altova	
If you do not want to use Altova LicenseServer click here to upload a license manually => Upload License					
To activate your software please enter or select the name of the Altova LicenseServer on your network.					
Altov	a LicenseServe	er: D	V02	C	
e C	A license is a	already a	isigned to you on LicenseServer at DEV02.		
	Name				
	Company	Altova	mbH		
	User count	50			
	License type	concur	ent		
	Expires in	703			
	SMP	703 day	; left		
	Return License	•	Check out License Copy Support Code	Save Close	
			Connected to Altova LicenseServer at DEV02		

マシ」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に戻すこかできません。7日過ぎると、(「ライセンスを戻す」をクリックして、マシンのライセンスをLicenseServerに戻すこかでき、このライセンスは、他のクライアントによりLicenseServerから取得することができます。LicenseServer管理者は、LicenseServerのWebUIを使用して、取得されナ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、マシン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には適用されないことに注意してくたさい。

<u>ライセンスのチェックアウト</u>

ライセンスが製品マシン上に保管されるようこ、ライセンスライセンスプールから30日間チェックアナオることができます。これにより、オフラインで作業することか可能になります。この機能はとても役に立ちます。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されていますが、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。おけよ、チェックアナされたライセンスはノストウェアのライセンスの認証ダイア ログのドジンを使用して「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コジス下をおこれます:(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」をクトックます(上のスクレーンショナ参照)。(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トされます。ライセンスのチェックアウ ト後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」がなくは「チェックイン」 ボタイご変更されます。「チェックイン」ボタンをクトックして、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オフライ ンで作業する期間をカッドーするようご確認してくたさい。 メモ ライセンスのチェックアウトを可能にするコは、LicenseServer上でチェックアウト機能が有効化されている必要があります。チェックアウトを試みる際この機能が有効化されていたい場合、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たさい。

サポートコードのコピー

「サポートコードのコピー」をクリックして、ライセンスの詳細をクリップボードにコピーしてください。これはオンラインサポートフォームを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 を使用することによ、IT 管理者は、リアルタイムでやホ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、<u>Altova Webサイトで</u>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、<u>Altova LicenseServer ドキュメートを参照してくたさい</u>。

- ▼ 注文フォーム
 - ᆿ <u>説明</u>

ソ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」ポタン、おけは「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
 - □ <u>説明</u>

Altova 製品登録ページをブラウザーのダブに表示します。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
 - <u>説明</u>

Altova サーバーは接続して、お新しいバ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

- ザポトセター
 - ᆿ <u>説明</u>

イターネット上にある Altova サポートセターへのレクとなっています。サポートセターゴは 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
 - ᆿ <u>説明</u>

イターネット上にある Altova のFAQ へのレクビなっています。FAQ データベースは Altova のサポートスタッフィンド常時更

新されています。

- ▼ コポーネトのダウロード
 - □ <u>説明</u>

イターネト上にあるAltovaのコンポーネトダウロードセターへのレクとなっています。このレク先から様々なコンポーネントソアウェアをダウロードして、Altova製品ともに使用することができます。ソアトウェアコンポーネントはXSLTやXSLFの プロセッサーからアプ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。コンポーネントダウロードセターにてご利用しているノア・ウェアは、通常無料でご利用しています。

- ▼ インターネオ上のStyleVision
 - ∃ <u>説明</u>

イターネト上にある<u>Altova Webサイト</u>へのレクゼなっています。<u>Altova Webサイト</u>では Style Vision 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 にれて
 - ∃ <u>説明</u>

スプラッシュ画面と製品の、ージョン番号が表示されます。StyleVisionの64ビオノ、ージ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)によい示されています。32ビオノ、ージョンコンサフィックスは存在しません。

付録 12

この付録はは(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クションに含め Shatat:

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

<u>技術情報</u>

StyleVision に関する技術的な情報か記述されます。

- OS とメモルの要件
- Altova XML パーサー •
- Altova XSLT とXQuery エンジン ٠
- Unicode のザポート
- イターやトの接続 •

ライセンス情報 StyleVisionの配布とライセンスに関する情報が含まれます。

- 電子的なノストウェアの配布 •
- ライセスメータレグ •
- 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、W 3C 仕様に従っています、ですから、XMLSpy の以前の・ージョン内のAltova エンジンよ児厳密です。この結果、以前のエンジンで無視されていたいさなエラーが、StyleVision によりエラーとて挙げられます。

例えば

- パス演算子の結果がソードと非-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 がードのシーケンスを評価しな 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ドキュメトおけまインスタンスドキュメトを必要に応じて修正してくたさい。

このセクションは、エンジの実装固有の機能を仕様別に整理して説明します。

- <u>XSLT 1.0</u>
- <u>XSLT 2.0</u>
- <u>XSLT 3.0</u>
- <u>XQuery 1.0</u>
- 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 (ワールド・ウイド・ウェブ・コンノーシアム) (W 3C) の<u>1999</u> <u>年 11 月 16 日版のXSLT 1.0 勧告</u> および <u>1999 年 11 月 16 日版のXPath 1.0 勧告</u> に準拠します。実装に関しての以下の 情報に注意してくたさい。

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、おけよ がHTML 出力 デフォルトで選択されている場合、XML おけよ 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 ク無しのスペースのための16 進数レフォレンス)がHTML コード内に文字の参照(or)、おけよ エンティティ参照 として挿入されます

12.1.2 XSLT 2.0

<u>ב הצלאשי</u>

- エンジン 適合性
- <u>下位互換性</u>
- <u>名前空間</u>
- スキーマ認識
- <u>実装固有の振る舞い</u>

適合性

StyleVision のXSLT 2.0 エンジノは W orld W ide W eb Consortium (ワールド・ウイド・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なるのは、XSLT 1.0 スタイルシートを処理 するためにXSLT 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 カに違いかあるかもしれないてとに注意してくたさい。

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ンストラクタ および関数を使用するため、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:スタイルシートませは、xsl:transform要素で宣言されます:

<xsl: スタイルシート 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"

```
</xsil:スタイルシート>
```

次の点に注意してくたさい

- XSLT 2.0 エンジノは (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、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で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 のかかは 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')でおよて fn:string('Hello') とて検証します。

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 エンジが、特定のXSLT 2.0 関数の振る舞いの実装-特定のアスペトをどのように扱うかの説明です。

xsl:result-document

追加してサポトされるエロードは以下の通りです(Altova-固有): x-base16tobinary おび 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パス および(ii) 相対パスを持つませまけされ、 file://プートコール。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 xbinarytobase64.

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パス および(ii) 絶対パを持つませまけたい file://プートコール 追加してサポートされる エンロードは以下の通りです(Altova-固有): x-binarytobase16 および xbinarytobase64.

メモ RaptorXML の先行製品である Altova XML で実装されて オンドのエンコード値は使用しな でくたさい 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 (ワールド・ウイド・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 エンジンは、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 <u>XPath 3.1 仕様</u>。

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、ドキュメト全体がソモリコードされ、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-ビットアプレでは、これは問題にないません。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

このセクション

- エンジン 適合性
- スキーマ認識
- <u>名前空間</u>
- <u>XML ソースと検証</u>
- 静的および動的な型のチェック
- ライブラ・ビジュール
- <u>外部関数</u>

- <u>照合順序</u>
- <u>数値データの精度</u>
- <u>XQuery 命令ザポト</u>

適合性

StyleVision のXQuery 1.0 エンジイよ W orld W ide W eb Consortium(ワールド・ウイド・ウェブ・コンノーシアム)(W 3C)の <u>2010 年 12 月 14 日版の XQuery 1.0 勧告</u>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。 下には、 XQuery 1.0 エンジムどのようにつんの機能を実装するかについて説明するリストが下に挙げたれています。

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
スキーマインスタレス	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たさい

- XQuery 1.0 エンジンは、上にリストされたプレフィックスを名前空間に対応する、ウナドとて認識します。
- Since the 上にフ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:はクエリプロローグ内で名前空間を宣言することがへ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")).
- クエリプロローグ内でdefault function 名前空間 式を宣言することにより、デフォルの関数 名前空間をすることにより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、プレフィックス×s:は名前空間を明確に宣言することは、また、これらのプレフィックスをクロノプローグ内で、インドすることは、使用することができます。(サンプル×s:date およびxs:year Month Duration。) XML スキーマ名前空間のために、他のプレフィッ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ります。(サンプルとdeclare 名前空間 alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04").)
- untypedAtomic, dayTimeDuration,おび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に注意してくたさい。ですから以下となります:xs: year MonthDuration。

関数のけるの名前空間、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、エラーか発生します。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、関数おけま型コンストラクタが使用されるか決定します。

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 ドキュメント内のXMLドキュメントは、整形式である必要があります。しかし、 XML スキーマロゴに 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 データのためこ生成され、 クロ 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、外部レフォレンスの存在(例、モジュールのため)、呼び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ー リのアスペ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。

クロめ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、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ーションが実行できない サングです。

ライブラ・モジュール

ライブラノモジュールは、再利用のイダ関数と変数を保管します。XQuery 1.0 エンジンオ単一の外部 XQuery ファイルに保管されて し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ナ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" };

すべての関数おとい変数は、モジュールは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。モジュールはクロレプ内の import module ステートメイを使用して X Query ファイルにインポートする際に使用されます。 import module ステートメイ は、ライブラノモジュール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";

if (\$modlib:company = "Altova")

then modlib:webaddress()

else error("No match found.")

外部関数

外部関数はサポートされていません。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

デオパトの照合順序は、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 ポートされる照合順序は、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、サポートされる照合順序の 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、デオパトの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 データ型はお小数点の後に20桁の制限が決ます。
- xs:float とxs:double データ型は15 桁の精度の制限があります。

XQuery 命令ザポー

Pragma 命令 は、サポートされていません。発生した場合、無視されフォール、シクの式が検証されます。

12.1.5 XQuery 3.1

StyleVisionのXQuery 3.1 エンジノよ W orld W ide W eb Consortium(ワールド・ウイド・ウェブ・コンノーシアム)(W 3C)の <u>2017 年 3 月 21 日版のXQuery 3.1 候補勧告</u>に準拠し、またXPath およびXQuery 関数 3.1.へのサポトを含みます。 XQuery 3.1 仕様は、3.0 仕様のスー ペーセナ です。XQuery 3.1 エンジノよ ですかXQuery 3.0 機能をサポートします。

実装固有の特性は<u>XQuery 1.0</u>同様です。

12.2 XSLT と XPath/XQuery 関数

このセクションでは、XPath およびまけは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エンジンで使用することができ、W3C 標準で定義された関数ライブラル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たさい

- W 3C 仕様にお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 X Query エンジンが、XPath/X Query 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り呼び出された場合、エラーが返されます。
- 全ての比較はUnicodeコードポイン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 とう形式でノアル化されます。

<u>xs:decimal の精度</u>

精度とは、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よく精度は小数点以下の19桁になります。

<u> 黙示的なタイムゾーン</u>

2つのdate、time、封は 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みます。値の中にタイムゾ ーンか明示的に与えられていたい場合、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

デフォルの照合順序は、Unicodeコードポイトをベースに文字列を比較するUnicodeコードポイト照合順序です。エンジルは 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序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。照合順序オプノョンの指定されていない場合、デフォルトのUnicodeコードポイスト照合順序が使用されます。

言語	URI
da: デノマーク語	da_DK
de:ドイン語	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: スペイン 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: フシス 語	<pre>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</pre>

it: イタ/ ア語	it_CH, it_IT
ja : 日本語	ja_JP
nb: ハルケー語 (ブークモール)	nb_NO
nl: オランダ語	nl_AW, nl_BE, nl_NL
nn: ハルウェー語 (ニーノシュク)	nn_NO
_{pt} : ポルガル語	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: Dシア 語	ru_MD, ru_RU, ru_UA
sv: 자ナーデン語	sv_FI, sv_SE

<u>名前空間軸</u>

名前空間軸はXPath 2.0 にて廃止されまたが、名前空間軸の使用はサポートされています。XPath 2.0 メカニズムに以名前空間情報へアクセスするコよ、in-scope-prefixes()、namespace-uri()、namespace-uri-for-prefix(関数を使用してくたい。

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 「コ収められており、altova: プレフィックスが、このセグションでは使用されます。製品の今後の、ージョンカ拡張機能への継 続的サポート、おけよ個別の関数の振る舞いは変更する可能性か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ントを参照してくたさい。

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、xstt 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 デジェイム 関数の詳細に記載されています(アのシンボルを参照してくたさい)。 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したされています。Altova 拡張関数などの、他のライブ ラカトらの関数はプレフィックスと共にリストされています。

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

<u>XSLT 関数</u>

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 ートするエディンコンの 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
- <u>位置情報</u>
- イメージ 関連した
- 数值
- シーケンス
- 文字列
- その他

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レで使用可能 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、おけば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、今後のリレ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

標準関数

distinct-nodes [altova:]

- altova:altova:distinct-nodes (country)は重複した値を持つものを除く、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、必須引数とて文字列をかします。評価された式の出力を返します。例えば altova:evaluate('//Name[1]') はドキュメント内の最初のName 要素のエレテンンを返します。式 //Name[1] は、 一重引用符を使用することにより、文字列としていなされます。

altova:evaluate 関数は オプションとて追加の引数を持つこかできます。これらの引数は、p1, p2, p3... pN の名前を持つス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 か整数である箇所のオームpx の名前と共に定義される必要が別ます。(ii) altova:evaluate 関数の引数は、(LO署名参照) 2 番目の引数からよ 数値順の 変数のノーケンスに対応した引数のノ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* である必要が別ます。

🗉 <u>サンプル</u>

```
<xsl:variable name="xpath" select="'$p3, $p2, $p1'" />
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
outputs "hi 20 10"
```

上のストに関して、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。

<u>
変数の使用方法を更に説明するサンプル</u>

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 か並べ替えの必要条件をリケエストする場合、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)["] order="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" order="ascending"/>の値になります。この sort 命令が、Order とつ要素のコンテキスト 内で発生する場合、Order 要素は Price の子の値に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。これよ、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" order="ascending"/>。上の場 合、並べ替え条件は sortkey 属性自身であり、Price まさは Date (また)

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とデフォルトの 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 に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、それぞれの引数には指定されるboolean の値に基づき保管 されます。

[<u>トップ</u>]

XBRL 関数

Altova XBRL 関数はXBRL をサポートする Altova 製品のエディシ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て必要と、入力ノードに参照されるXBRL フトノート 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ク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び時刻データ型で保存されているデーダ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 StyleVision 2021 Basic Edition

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、デジョンか拡張機能への 継続的サポート、おけば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

- ▼ 機能によりバープ化
 - <u>xs:dateTime に期間を追加して、xs:dateTimeを返す</u>
 - 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
 - フォーマトと期間の取得
 - 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、秒数を返す
 - 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、日付時刻ませよ時刻を構築する
 - 年齢に関連した関数

▼ 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
- <u>altova:parse-time</u>
- altova:time-no-TZ
- <u>altova:weekday-from-date</u>
- altova:weekday-from-dateTime
- altova:weeknumber-from-date
- altova:weeknumber-from-dateTime

[<u>トップ</u>]

xs:dateTime に期間を追加する 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 によい区切られています。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 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。 ゴ 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 を返します。
- add-months-to-dateTime [altova:]

- 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 とする XP3.1 XQ3.1

• altova:add-days-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), 10)は2014-01-25T14:00:00を返します。
- altova:add-days-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), -8)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 XQ3.1
xs:dateTime に時間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(xs:dateTime("2014-01-15T14:10:00"), 45) 2014-01-15T14:55:00 を返します。
- 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 XQ3.1
```

xs:dateTime に秒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 の引数とて与えられた xs:dateTime に追加される秒数です。結果はxs:dateTime 型です。 <u>サンプル</u>

- altova:add-seconds-to-dateTime(xs:dateTime("2014-01-15T14:00:10"), 20) 2014-01-15T14:00:30 を返します。
- altova:add-seconds-to-dateTime(xs:dateTime("2014-01-15T14:00:10"), -5) 2014-01-15T14:00:05 を返します。

[<u>トップ</u>]

xs:date に期間を追加する xp3.1 xQ3.1

こわらの関数はxs:date に期間を追加し、xs:date を返します。xs:date 型はCCYY-MM-DD フォーマナです。

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
```

日付までの期間を牛致で表示します。第20万日致は第10万日致とし、与えられてxs:date し。追加される牛致です。 xs:date 型です。

🗉 <u>サンプル</u>

- 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 XQ3.1

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。

□ <u>サンプル</u>

• 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 日付までの期間を日数で表示します。第2の引数は第1の引数として与えられたxs:date に追加される日数です。結果は xs:date 型です。

🗉 <u>サンプル</u>

- 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。

[<u>トップ</u>]

フォーマトと期間の取得 xp3.1 xq3.1

これらの関数は入力 xs:duration おはxs:string を解析し、それそれ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 とす る XP3.1 XQ3.1 第1 の引数とて提出された期間を、第2 の引数とて提出された文字列にとをフォーマナします。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) $\frac{\delta}{\delta}$ xs:duration $\frac{\delta}{\delta}$ する XP3.1 XQ3.1

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に解析され、 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]") 🕻 "P2DT2H53M11.7s" **былат**.
- 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" **友反します**。

「トップ 】

xs:time に期間を追加する xp3.1 x03.1

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 おけよ-hh:mm のフォーマトで表示しています。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
XO3.1
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
XP3.1 XO3.1
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のレポーネントは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 xq3.1

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 フォーマナは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 とてシステムシロックか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
この関数はお 数を必要とません。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。

ヨーサンプル
```

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:]

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:]

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、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たいにに

注意してくたさい。

ヨ 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、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たさい。
```

□ <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、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いてとい注
意してくたさい。

ヨーサンプル
```

• 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(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 XQ3.1

```
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。ネガティブな時間の値で期間の
結果をナビゲートすることができます。
```

□ <u>サンプル</u>

- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5D") は日間の時間の総 計である120を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H") は日間の時間の 総計プラス時間である122を返します。
- altova:hours-from-dayTimeDuration-accumulated(xs:duration("P5DT2H60M") は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

- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("PT60M") は を返します。 60
- 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 返します。
- altova:minutes-from-dayTimeDuration-accumulated(xs:duration("P1D") は一日の秒数の総 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引数にお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。ネガティブな秒の値で期間をナビゲートします。

□ <u>サンプル</u>

- 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 します。
- 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 おはxs:dateReturn から週数を返す xp3.1 xq3.1

これらの関数は、xs:dateTime おはxs:date から曜日を整数とつを返します。曜日は(米国式フォーマトを使用して)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。欧州のフォーマナでは月曜(=1)として番号付けられます。日曜=1 である米 国フォーマナは整数0 がフォーマナ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:]

altova:weekday-from-dateTime(DateTime as xs:dateTime) をxs:integer とする XP3.1 XQ3.1

時刻付きの日付を単一の引数とて、この日付の曜日を正数として返します。曜日は日曜=1 から開始して番号を付けます。(月曜 =1 とする)ヨーロッパ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(アの次の署名を参照)。

🖃 サンプル

• 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 どる XP3.1 XQ3.1

時間月の日付を最初の引数とて、この日付の曜日を正数とて返します。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です。第2 の引数がひ、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(前の次の署名を参照)。 = サンプル

- 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(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 XQ3.1

日付を最初の引数と、この日付の曜日を正数とて返します。曜日は月曜=1 から開始して番号を付けます。第2(フォーマオ)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 の引数がさい場合、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)。

🗉 <u>サンプル</u>

- 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。

[<u>トップ</u>]

xs:dateTime ませは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。週の番号付けれ、米国、欧州、イスラムのカレンダー フォーマナで使用することができます。週の始まめ異なるため、週数の番号付けれ、カレンダーのフォーマナトにより異なります。(米国フォー マナでは、日曜、欧州フォーマナでは月曜、イスラムフォーマナ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。 サポートされるカレンダーの値は次のとおしです:

- 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)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0 です。

- *□ <u>サンプル</u>*
 - 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この日付よしに一週間先です。

weeknumber-from-dateTime [altova:]

altova:weeknumber-from-dateTime(DateTime as xs:dateTime, Calendar as xs:integer)を xs:integer とする XP2 XQ1 XP3.1 XQ3.1 正数とて提出された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。 サポートされるカレンダーの値は次のとおりです:

- 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)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dateTimeの曜日は日曜日です。米国およびイスラムのカレンダー 北欧州カレンダーのこの日付よ 光一週間先です。

[<u>トップ</u>]

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 xq3.1

関数は、xs:date, xs:time おはxs:duration の構文エンポ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 の引数は、それぞれ、年、月、日を表します。xs: date 型の値を構築するかが結合されます。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あります。例えば第2 の引数(月の部分)は12 以上であってはなりません。 *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)
をxs:time とする XP3.1 XQ3.1
```

第1、第2、第3引数はそれそれ時間数(0から23)、分数(0から59)、おとび秒数(0から59)の値です。これらの値は、 xs:time型の値を作成するために結合されます。整数の値は、それぞれの部分の正しい範囲内である必要があります。例えば、秒 (分)引数は、59以上であってはなりません。値にタイムゾーンを追加するこは、この関数の他の署名を使用してくたさい(次の署名 を参照)。

```
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 とる XP3.1 XQ3.1

第1、第2、第3引数はそれそれ時間数(0から23)、分数(0から59)、および秒数(0から59)の値です。第四の引数は値の一部と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以上であってけないません。

```
🗉 <u>サンプル</u>
```

• 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 型の値を構築するかのコな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、
```

第2の引数は、月数値の部分を与えます。第2の引数(月数)が同じおけよい大きくなると、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引数に加算され、(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型を作成するけ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) をxs:dayTimeDuration とする XP3.1 XQ3.1

xs:dayTimeDuration型の値を構築するよめはよ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、それそれ期間の時間数、分数、秒数値のです。これらの3つの時間引数は、次の高し単位の値に加算され、結果は期間の全体の値を計算するよめに使用されます。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ようなに使用されます。xs:yearMonthDuration型の期間を構築するようなは、次の署名を参照し てくたさい。

」 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。

[<u>トップ</u>]

文字列入力から日付、日付時刻または時刻を構築する xp2 xq1 xp3.1 xq3.1

関数は、文字列を引数として、xs:date、xs:dateTime おはxs:time データ型を構築します。文字列は、データ型のエレポーネ ントを提出された。やーン引数をベースして分析されます。

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 ターン(コンポーネントのシーケンス)を指定します。DatePattern は下にリストされるコンポーネント指定子および狂意の文字であるコンポーネント セン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。

- D 日付
- ™ 月
- Y 丘

DatePattern の ローンは Date の ローンニー致する必要があます。出力はxs:date 型であるため、出力は常に YYYY-MM-DD 構文フォーマトレンはます。

🗉 <u>サンプル</u>

- 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(I	D ateTime as xs:string, DateTimePattern as xs:string)を
xs:dateTime とする XP2 1	KQ1 XP3.1 XQ3.1
DateTime 入力文字列をxs:d	ateTime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(エポーネトのシーケンス)	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び狂意の文
字であるエポーネトセレータと	共に説明されています。下のサンプルを参照してくたさい。
D	В
М	月
Y	年
Н	時間
m	分
s	秒

DateTimePatternのローンはDateTimeのローンに一致する必要が決ます。出力はxs:dateTime型であるため、出力は常にyyyy-MM-DDTHH:mm:ss構文フォーマトにおます。

- 🗉 <u>サンプル</u>
 - 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

XQ1 XP3.1 XQ3.1

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(エポーネン

トのシーケンズ)を指定します。TimePattern は 下にリストされるエポーネント指定子および狂意の文字であるエンポーネント セ

パ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たさい。

H 時間

m 分

s 秒
```

TimePattern の ターンは Time の ターンに一致する必要が別ます。出力はxs:time 型であるため 出力は常に YYYY-HH:mm:ss 構文フォーマナーになります。

- 🗉 <u>サンプル</u>
 - 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。

[<u>トップ</u>]

年齢に関連した関数 xP3.1 xQ3.1

関数は計算された年齢(i)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) をxs:integer とする XP3.1 XQ3.1 引数と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、あるオブシ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 年よ児大きい場合、ませま一年の場合、返される値は負の数です。

∃ <u>サンプル</u>

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 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 最初の引数とて提出された開始日から第2の引数の終了日までの日数を数えて、あるオブシェクトの年齢の年数である正数を返 します。最初の引数が1年ませま第2の引数よ戌後の日付の場合、返される値は負の数です。

🗉 <u>サンプル</u>

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 引数とシステムクロックから取得された現在の日付と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 ・
比早いまたは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か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。

🗉 <u>サンプル</u>

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) を(xs:integer)* とする XP3.1 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、おう遅くてもかまいません。 返される値は入力の日付か現 在の日付より早いかまたっぽいかを示しません。 戻される値は常に正の数です。

🖃 <u>サンプル</u>

- 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。

[<u>トップ</u>]

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 の現在のレ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 ドキュメノト内のXQuery 式。

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/xsltextensions に収められており、altova: フレフィックスが、このセクションでは使用されます。製品の今後の、デジョンか拡張機能への 継続的サポート、おけば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

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 つの引数と、位置情報を文字列とて出力します。第3 の引数 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こ、正数の値 1 から4 を使用します。(アの「位置情報出力文字列フォーマト」を参照してください)。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 とれできます。

□ <u>サンプル</u>

- 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" 方返します。

□ 位置情報出力文字列フォーマト:

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

1

2

4

4) によ、識別されます。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。

<mark>度、分、10進の秒、方角のサフィックス付き</mark>(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

<mark>10 進の度、方角のサンヘクス付き(N/S,E/W)</mark> D.DDN/S D.DDE/W <u>サンプル</u>33.33N 22.22W

3 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" <u>サンプル</u>: 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)のオーダのプラスサインは注意です。 +/-D.DD +/-D.DD <u>サンプル</u>: 33.33 -22.22

□ <u>Altova Exif 属性: 位置情報</u>

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	Е	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 XQ3.1

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ーは以下のリストの通りです。

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。

🗉 <u>サンプル</u>

- altova:parse-geolocation("33.33 -22.22") は2 つのxs: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 (48.858222222222, 24.2945) のーケンズを返します。
- altova:parse-geolocation('48°51''29.6"N 24°17''40.2"') は2つの xs:decimals (48.858222222222, 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。
- □ <u>位置情報入力文字列フォーマナ:</u>

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)を含む必要かあります。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10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ためコ使用されていると、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。このセグションのサンプルでは、入力文字列を区別するためコ使用されている引用符 は黄色い(")でい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("")でい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) のたまのプラスサインは任意です。
 +/-D°M'S.SS"
 +/-D°M'S.SS"
 <u>サンプル</u>: 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'
 <u>サンプル</u>
 +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))のオーダ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

■ <u>Altova Exif 属性: 位置情報</u>

Altova XPath/XQuery エンジンイカスタム属性 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	Е	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 <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。

□ <u>サンプル</u>

• 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。

□ <u>位置情報入力文字列フォーマナ:</u>

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)を含む必要かあります。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10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ナメロマ使用されていると、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 ブルコン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ナメロで使用されている引用符 は黄色い(") でい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("") でいイライトされています。

- 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
 <u>サンプル</u>; 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'
 <u>サンプル</u>: +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))のオック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>フォーマ水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

■ <u>Altova Exif 属性: 位置情報</u>

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	Е	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。

- 🗉 <u>サンプル</u>
 - altova:geolocation-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。
- <u>位置情報入力文字列フォーマホ:</u>

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)を含む必要かあります。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10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ナムビック用されていると、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けっか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ナムビュ使用されている引用符 は黄色い(") でい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言い("") でいイライトされています。

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"
 <u>サンプル</u>: 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'
 <u>サンプル</u> +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))のオック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>フォーマト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

■ <u>Altova Exif 属性: 位置情報</u>

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(アのテーブル参照) GPSLatitude、 GPSLatitudeRef、GPSLongitude、GPSLongitudeRef。

GPSLatitude	GPSLatitudeRef	GPSLongitude	GPSLongitudeRef	Geolocation
33 51 21.91	S	151 13 11.73	Е	33°51'21.91"S 151° 13'11.73"E

geolocations-bounding-rectangle [altova:]

al tova: geol ocati ons-boundi ng-rectangle(Geol ocati ons を xs: sequence として Geol ocati on Output String Format を xs: integer として)をxs: string とて XP3.1 XQ3.1 最初の引数とて文字列のシーケンズを取ります。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ズを返します。 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 M への)の範囲です。

関数の2番目の51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」を参照してくたさい)。

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。

□ <u>サンプル</u>

- altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 0.11832"), 1) はシーケンス("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) はーケンス("51.50939 -70.8893334", "42.5584577 16.3707266")を返します
- □ <u>位置情報入力文字列フォーマナ:</u>

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)を含む必要かあります。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。緯度がりつのフォーマトで、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ナメロク用されていると、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 ブルコン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オーメロで用されている引用符 は黄色い(")でい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("")でい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)のオックのプラスサインは任意です。
 +/-D°M'S.SS"
 +/-D°M'S.SS"
 <u>サンプル</u>: 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
 <u>サンプル</u> 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))のオーダ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>フォーマ水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

□ <u>位置情報出力文字列フォーマナ:</u>

1

2

3

Δ

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ナによりフォーマナされます。希望するフォーマナは整数 ID(1 から 4)によし識別されます。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。

<mark>度、分、10進の秒、方角のサフィックス付き</mark>(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)のオックのプラスサインは狂意です。 +/-D°M'S.SS" +/-D°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)のナック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>Altova Exif 属性: 位置情報</u>

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 タブの連結です: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	Е	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) か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か同じ場合に作成される閉じられたフィギュアを作成しない場合、フィギュアを 閉じるために、最初のポイトか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、(ア/ニノスト されるフォーマ・トの) 位置情報入力文字列により提供されます。Geolocation 引数が 多角形エリア内にある場合、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。

🗉 <u>サンプル</u>

- 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。

□ <u>位置情報入力文字列フォーマナ:</u>

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ナメロマ使用されていると、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 ブルコン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ナメロマ使用されている引用符 は黄色い(") でい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("") でい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) のオック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
 <u>サンプル</u> 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'
 <u>サンプル</u>
 +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))のオーダのプラスサインは任意です。+/-D.DD +/-D.DD
 D.DD
 <u>サンプル</u> 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

■ <u>Altova Exif 属性: 位置情報</u>

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	Е	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)に比応明されている Geolocation (最初の引数) が長方形のエリア内に存在するかを決定します。全ての引数(Geolocation、RectCorner-1 および RectCorner-2) は (アバリスト されるフォーマナの 位置情報入力文字列に比提供されます。Geolocation 引 数が長方形エリア内にある場合、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。

🗉 <u>サンプル</u>

- 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。

□ <u>位置情報入力文字列フォーマホ:</u>

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)を含む必要かあります。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10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ナメロマ使用されていると、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 ブルコン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ナメロで使用されている引用符 は黄色い(")でい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("")でい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"

<u>サンプル</u>: 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)のたまのプラスサインは任意です。
 +/-D°M.MM'
 +/-D°M.MM'
 <u>サンプル</u>
 +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))のオーダのプラスサインは任意です。+/-D.DD +/-D.DD
 D.DD
 <u>サンプル</u>: 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

■ <u>Altova Exif 属性: 位置情報</u>

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	Е	33°51'21.91"S 151° 13'11.73"E

[<u>トップ</u>]

12.2.1.4 XPath/XQuery 関数: イメージに関連

以下のイメージニ関連したXPath/XQuery 拡張関数は、StyleVisionの現在のレージョンによりサポートされています。おこ、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、おさは(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イで使用可能 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、このセグションでは使用されます。製品の今後の、デジョンか拡張機能への 継続的サポート、おけば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、今後のリレ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

suggested-image-file-extension [altova:]

altova:suggested-image-file-extension(Base64String as string)をstring? とする XP3.1 XQ3.1

イメージファイルのBase64 エン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 エンコード内の記録として返します。 返された値は、エンコード内で使用することのできるイメージ型情報を基にしたという。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。

- altova:suggested-image-file-extension(/MyImages/MobilePhone/Image20141130.01)は 'jpg'を返します。
- 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と仮定します。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パラメーターです:サイズ、回転、およびカオリティ。

• 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。アスペケト率には保持されません。高さと幅は自動的にイメージの長いおよい短い辺に逢わされるため、高さと幅の指示には関係ありません。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。
Х	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。アスペク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、回転させます。負の値はイメージを反時計回り、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よめ、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 説明を参照してくたさい)。
- 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、クオーティのトーセンテージのインジケーターではありません。サイズとウオーティのとちらかを優先すると、もう一方の優先度が下がります。フルカラーソースのイメージでは、75か通常最高値と見なされています。もし、75が満足の、く結果をもたっさざい、場合は、値を上げてく

だい。

メモ Exif データが元のイメージェ存在する場合、変換時に削除され、変換されたイメージコは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 のカオリティレベルを与えられます。
- 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 ケー率は保持されます。
- mt-transform-image(Images/Image[@id='43'], (400, 280), image-exifdata(\$XML1/\$XML1/Images/ReferenceImage)/@OrientationDegree, 75)
 このサンプルは、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400x280 ピ クセルに設定されています。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 おは6)かられそれ対応する度数 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 値の変換はないてとこ 注意してくたさい。(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-度ごとに時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。)。



□ <u>標準 Exif メタタク</u>

- 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
- 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 イメージを引数して、イメージのExif メタデータを含む Exif と 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 Jスト は以下の通りです。 ペ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 Jスト参照*) Altova 特有の属性値のペアもおと生成されます。 これらの Altova Exif 属性には以下のとおしです。

□ <u>サンプル</u>

- 属性ごアクセスするゴ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
- 全ての属性にアウセスするコよ、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コよ、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。

■ <u>Altova Exif 属性: 位置情報</u>

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	Е	33°51'21.91"S 151° 13'11.73"E

Altova Exif 属性: OrientationDegree

Altova XPath/XQuery エンジンオカスタム属性 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 おはよる)から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 値の変換はないことに 注意してくたさい。(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-度ごとに時計回りにジャンプさせぞれぞれ7、4、および5 の値を取得します。)。



<u>標準 Exif メタデータ タグのリスト</u>

- 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

[<u>トップ</u>]

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レで使用可能 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、・・ションカ拡張機能への 継続的サポート、おけば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、今後のリレ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い番号を生成します。この番号は前に生成された番号を、Increment 引数で指 定された値ごとこへクルメトします。実質的には、altova:generate-auto-number 関数は、ID 引数により指定されたカ 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こへクルメトされます。ResetOnChange 引数の値が、前の関数呼び 出しから変更されると、生成される番号の値がStartsWith 値にした大されます。自動付番は<u>altova:reset-auto-</u> number 関数を使用していた大することができます。

🗉 <u>サンプル</u>

- 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か始まり、関数が呼び出される度に1ずつインクリメトします。その後の呼び出しの4番目の'SomeString"引数がである限り、インクリメトは継続されます。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 にノセナされます。ChapterNumber の値もaltova:reset-auto-number ("ChapterNumber")ようことかされます。
- 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 引数内で名づたれた自動付番カウクーの番号をルセナします。引数内のカウター名を作成した関数の引数 に
以指定された数値にルセンされます。altova:generate-auto-number ヨ サンプル

- altova: reset-auto-number ("ChapterNumber") は <u>altova: generate-auto-number</u> 関数にお 作成されたhapterNumber とら名の自動付番カンターをルセトします。
- ChapterNumber を作成した<u>altova:generate-auto-number</u> 関数のStartsWith 引数の値に数値をJ セルします。

[<u>[トップ]</u>]

数値関数

hex-string-to-integer [altova:]

```
altova:hex-string-to-integer(HexString as xs:string) をxs:integer とる 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') はエラーを返します。
      • 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') はエラーを返します。

    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し、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。

[[<u>|</u>]]

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) スキーマコンポーネントのノストとトー スペクティブのプロッティです。 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 はお以下の2つの引数が存在します。1つのゼラ引数と他の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ボーネントをロケートします。2-引数関数は自身のQName にお職別 される特定のンポーネントの型を返します。両方の場合、戻り値は関数です。返されたエンポーネント内をナビゲートするはよ、特定のエン ポーネントのプロンティを選択する必要があります。プロンティが非動的アイテム(すなオカエンポーネントの場合)このエンポーネントのプロンティ をさらく選択してナビゲートすることが可能です。選択されたプロンティが動的なアイテムの場合、アイテムの値が返され、ナビゲートを更にご行う ことはできません。

メモ Xquery 式ではスキーマオ胡示的にインポートされる必要があます。XPath 式では、スキーマオ処理環境にインポートされる必要 かあます。 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レで使用可能 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、デジョンか拡張機能への 継続的サポート、おけば国別の関数の振る舞いは変更する可能性があることに注してくたさい。Altova 拡張機能へのサポートに関し ては、今後のリレースのドキュッシム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

▼ Schema (引数無し)

```
altova:schema() as (function(xs:string) as item()*)? XP3.1 XQ3.1
schema エポーネト全体を返します。schema エポーネトをschema エポーネトのパロ デ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、ティがエレポーネトの場合、このエレポーネトのプロ、ティのトン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けっかに繰り返すことができます。
- コ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 エパーネトのプロッティ

```
"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コンポーネト型のプロ ディが下にリストされています。

メモ Xquery 式ではスキーマオ胡示的にんポートされる必要が対はす。XPath 式では、スキーマオ処理環境にんポートされる 必要が対はす。例えば XSLT にはxslt:import 命令を使用してんポートします。

□ <u>サンプル</u>

- 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おけね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おけ複合型の最初の名 前を返します

<u> コンポーネントとそのプロパティ</u>

Assertion

プロパティ名	プロ/ 守ィ型	プロ/ 守/値
kind	文字列	"Assertion"
test	XPath プロ ゲイレコード	

⊟ Attribute Declaration

プロ/ 守ィ名	プロ/ 守/型	プロ/ 守/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 おけたComplex Type	
scope	プロ 守を持つ関数 (″class″:″Scope″, "variety": "global" or "local", "parent": the containing Complex Type or Attribute Group)	
value constraint	存在する場合、プレディを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこ は"default", "value": atomic value, "lexical form": string。"value" プロディは namespace-sensitive 型のために使用するこ とまできなことに注意してくたさい。	
inheritable	ブール値	

Attribute Group Declaration

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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) のシーケンス	
attribute wildcard	任意の属性のフィルドカード	

⊟ Attribute Use
プロ/ ディ名	プロ/ 守ィ型	プロ/ 守/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/型	プロ/ 守/値
kind	文字列	"Wildcard"
namespace constraint	プロ 守を持つ関数 ("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/ ディ名	プロ/ 守ィ型	プロ/ 守ィ値
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	任意の属性のフィルドカード	
content type	プロ、守 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

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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 守疹持つ関数 ("class":"Type Table", "alternatives": sequence of Type Alternative, "default type definition": Simple Type or Complex Type)	
scope	プロ 守⁄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 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/ ディ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守 を持つ関数 ("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/ ディ名	プロ/ 守/型	プロパティ値
---------	---------	--------

kind	文字列	ファセオの名前、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	ファセ小の列挙のみ Array(xs:anyAtomicType	列挙値を含む配列。それそれがトミック値 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)

Identity Constraint

プロ/ 守ィ名	プロ/ 守/型	プロ/ 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 (<i>″</i> key″)″unique″]″keyRef″)	
selector	XPath プロ/ ፝ኇィレコード	
fields	XPath プロ・ディレコード のシーケンス	
referenced key	(keyRef 07-5007): Identity Constraint	対応するキー制約

Model Group

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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/ ዮィ名	プロ/ 守/型	プロ/ 守ィ値
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/ ディ名	プロ/ 守/型	プロ/ 守ィ値
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/ ディ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数封出文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おけよModelGroup	

⊟ Simple Type

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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	含まれるエレポーネント	
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	(ユニオン型のナゴの) Simple Type のシーケンス	

■ Type Alternative

プロ/ 守/名	プロ/ 守/型	プロパティ値
kind	文字列	"Type Alternative"
test	XPath プロ/ ディレコード	
type definition	Simple Type おはComplex Type	

■ XPath Property Record

プロパティ名	プロ/ 守/型	プロパティ値
--------	---------	--------

namespace bindings	プロ 守を持つ関数のシーケンス("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の静的ベース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エンポーネントの型を返します。エンポーネントのプロ・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、ティがエレポーネトの場合、このエレポーネトのプロ、ティのトン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ために繰り返すことができます。
- コパ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。

メモ Xquery 式ではスキーマオ明示的にインポートされる必要があます。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。

- *□ <u>サンプル</u>*
 - import schema "" at "C:\Test\ExpReport.xsd";
 altova:schema("element declaration", xs:QName("OrgChart"))("type definition")
 ("content type")("particles")[3]!.("term")("kind")
 lお番目のparticles 工ポーネトの kind プロ テrを返します。このparticles 工ポーネトは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")]
 li各 emailType コポーネトの各 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パティ

Assertion

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Assertion"
test	XPath プロ/ ティレコード	

Attribute Declaration

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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 守控持つ関数 (″class″:″Scope″, "variety": "global" or "local", "parent": the containing Complex Type or Attribute Group)	
value constraint	存在する場合、プレ ディを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこ は"default", "value": atomic value, "lexical form": string。"value" プロ ディは namespace-sensitive 型のために使用するこ とまできないことに注意してくたさい。	
inheritable	ブール値	

⊟ Attribute Group Declaration

プロ/ ディ名	プロ/ 守ィ型	プロ/ 守/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) のシーケンス	
attribute wildcard	任意の属性のフィルドカード	

⊟ Attribute Use

プロ/ 守/名	プロ/ 守ィ型	プロ/ テー/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	

プロ/ ディ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守 を持つ関数 ("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/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	任意の属性のフィルドカード	
content type	プロ 守 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 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/ ディ名	プロ/ 守/型	プロ/ 守ィ値
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 守 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⁄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/ ディ名	プロ/ 守ィ型	プロ/ 守/値
kind	文字列	"Wildcard"
namespace constraint	プロ、守/を持つ関数 ("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/ ディ名	プロ/ 守/型	プロ/ 守/値
kind	文字列	ファセオの名前、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	ファセオの列挙のみ Array(xs:anyAtomicType	列挙値を含む配列。それそれがトミック値 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)

Identity Constraint

プロ/ 守ィ名	プロ/ 守ィ型	プロ/ 守ィ値
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 (<i>″</i> key′]″unique′́]″keyRef′́)	
selector	XPath プロ/ ティレコード	
fields	XPath プロ・ディレコード のシーケンス	

referenced key	(keyRef のためのみ): Identity Constraint	対応する た 制 約

Model Group

プロ/ 守ィ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Model Group"
compositor	文字列 (″sequence´゚゚ ″choice´゚ ″all´)	
particles	Particle のシーケンス	

Model Group Definition

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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/ 守/型	プロ/ 守/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/型	プロ/ 守ィ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数封また字列("unbounded")	
term	Element Declaration、Element Wildcard、 おゴよModelGroup	

■ Simple Type

プロ/ ዮィ名	プロ/ 守/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	含まれるエレポーネト	
base type definition	Simple Type	
facets	Sequence of Facet	
fundamental facets	(実装されていな)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/ 守/名	プロ/ 守/型	プロパティ値
kind	文字列	"Type Alternative"
test	XPath プロ/ ティレコード	
type definition	Simple Type おはComplex Type	

XPath Property Record

プロ/ 〒/名	プロ/ テティ型	プロパティ値
namespace bindings	プロ/ 守∕を持つ関数のシーケンス(″prefix″: string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の静的ベース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ドキュメトの要素おけば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」なるためにXMLドキュメトはスキーマ宣言を持つ必要があます。

🗉 <u>サンプル</u>

- 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ント (単純型ませは複合型)を取りコンポーネントのkind propertyの値を返します。

■ Assertion

プロ/ 守/名	プロ/ 守ィ型	プロ/ 守/値
kind	文字列	"Assertion"
test	XPath プロ/ ティレコード	

⊟ Attribute Declaration

プロ/ ディ名	プロ/ 守ィ型	プロ/ 守/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 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 ディを持つ関数 ("class": "Value Constraint"、"variety": "fixed" おこ は"default", "value": atomic value, "lexical form": string。"value" プロ ディは namespace-sensitive 型のために使用するこ とまできないことに注意してくたさい。	
inheritable	ブール値	

Attribute Group Declaration

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
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) のシーケンス	
attribute wildcard	任意の属性のフィルドカード	

⊟ Attribute Use

プロ/ 守ィ名	プロ/ 守/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/ ディ名	プロ/ 守/型	プロ/ 守/値
kind	文字列	"Wildcard"
namespace constraint	プロ/ 守/を持つ関数 ("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	任意の属性のフィルドカード	
content type	プロ 守 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 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/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 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⁄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 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/型	プロ/ 守ィ値
kind	文字列	"Wildcard"
namespace constraint	プロ 守 を持つ関数 ("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/ ディ名	プロ/ 守/型	プロ/ 守/値
kind	文字列	ファセオの名前、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。それそれがアトミング値 のシーケンスである場合があります。(メモ 列挙ファセトに関しては、実際の型にかか わらず "value" 値プロ・ディは文字列のシ
		ーケンスです)

Identity Constraint

プロ/ ディ名	プロ/ 守/型	プロ/ 守/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 (<i>″</i> key″]″unique″]″keyRef″)	
selector	XPath プロ/ ディレコード	
fields	XPath プロ・ディレコード のシーケンス	
referenced key	(keyRef 075007): Identity Constraint	対応するキー制約

Model Group

プロ/ 守ィ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Model Group"
compositor	文字列 (″sequence´゚゚ ″choice´゚ ″all´)	
particles	Particle のシーケンス	

■ Model Group Definition

プロパティ名	プロ/ 守/型	プロ/ 守/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/ ディ名	プロ/ 守/型	プロ/ 守ィ値
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/型	プロ/ 守ィ値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数封出文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おゴよModelGroup	

⊟ Simple Type

プロ/ ティ名	プロ/ 守/型	プロ/ 守ィ値
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	含まれることポーネント	
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	(リスト型のみのオーダの) Simple Type	
member type definitions	(ユニオン型のための) Simple Type のシーケンス	

Type Alternative

プロパティ名	プロ/ 守/型	プロ/ 守ィ値
kind	文字列	"Type Alternative"
test	XPath プロ ディレコード	
type definition	Simple Type おはComplex Type	

■ XPath Property Record

プロパティ名	プロ/ 守/型	プロパティ値
namespace bindings	プロ・ディを持つ関数のシーケンス("prefix": string, "namespace": anyURI)	
default namespace	anyURI	
base URI	anyURI	XPath 式の静的ベースURI
expression	文字列	文字列とての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ケ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/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、デジョンカ拡張機能への 継続的サポート、おけば国別の関数の振る舞いは変更する可能性があることに注してくたさい。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 (AttributeName as xs:string) をattribute()* とする 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」対して行われます。これは、コンテキストノードが親要素ノードである必要があることを意味します。 = サンプル

• 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::軸マオして行われます。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:

r = 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おります;

f = このオプションの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 E か指定されていない場合、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れた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つしか特たさい、関数と同じ効果を持ちます(前の署名)、ですが、空のシーケンスに第2の引数としては許可されていません。

□ <u>サンプル</u>

• 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",)見つからな第2引数を返します。
- elements [altova:]

altova:elements(ElementName as xs:string) をelement()* とする xP3.1 xQ3.1 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::軸に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。 = <u>サンプル</u>

• **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 ElementName 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::軸に対して行われます。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。

r = 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 = COTプレンの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 <math>f が指定されていない場合、MyElem はMyElement を返します。 f = 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.

フラッグは順序に関わ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つませよ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つしか特たない関数と同じ効果を持ちます(前の署名)、ですが、空のノーケンスに第2の引数としては許可されていません。

🗉 <u>サンプル</u>

- 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グエラーを返します。
- altova:elements("MyElement",) 見つからみ第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 つの引数を必要とます。最初の引数は1 つ、おけは1 つ以上のデータ型のアイテムのシーケンスです。第2 の引数 Condition は(1 のアトティを持つ)1 つの引数を必要と、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れて、Sequence の各アイテムか提出されます。(注意: この関数は1 つの引数のみを必要 とます。) 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 終了の結果として返されます。

🖃 <u>サンプル</u>

• 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"), (doc-available#1)) はxs:string 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 も存在しな 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。

上のサンプルについての注意点

- 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 を返する単一の引数を必要とます。(ですから、提出されたURI でのドキュメントはXMLドキュメントである必要があります。)
- doc-available 関数は altova: find-first の第2引数である condition で使用することができます。 これは1つの引数(アレティー1)のみを必要とするからであり、item()を入力(URI とて使用される文字列)とて、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たさい。アタッチされている#1 サフィックスは関数が1 2007/ティであることを表示するかのです。doc-available#1 の意味は以下のとおしです:アリティ=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、単一引数とていてします。この結果、2 200文字列の各自づは、文字列をURIとて使用し、URIIボキュメ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 つか各当する場合、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 ポジンコンは、altova:find-first 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:相対いなは、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。
- find-first-combination [altova:]

```
altova:find-first-combination((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 つまけお つ以上のデータ型のアイテムです。
- 第3の引数 Condition は (2のアケ を持つ) 2つの引数を必要と、 boolean を返す XPath 関数に対しての 参照です。

seq-01とseq-02のアイテムは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なされました。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), (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 の結果とて 返されます。以下の点に注意してくたさい(i) Condition 関数が提出された引数ペア内で繰り返され、true() 若評価しな場 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おゴゴアイテムでない場合。

🗉 <u>サンプル</u>

- 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) のシーケンを返します。
- altova:find-first-combination(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 34}) は xs:integers (11, 23) のシーケンを返します。
- 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
ご
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 つまけお 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のアイテム指定された組み合わせで、Condition内の関数の引数として、なれまた。組み合わせお、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() と評価するよう指示する最初のペアはLiova:find-first-pair の結果とて返されます。以下の点に注意してくたさい(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 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まアイテムでは、場合。

🗉 <u>サンプル</u>

- 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かないため)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 ご 関数は3 つの引数を必要します:

- 最初の2 つの引数 Seq-01 とSeq-02, は 1 つまけお つ以上のデータ型のアイテムです。
- 第3の引数 Condition は (2のアケ を持つ) 2つの引数を必要と、 boolean を返す XPath 関数に対しての 参照です。

seq-01とseq-02のアイテム指定された組み合わせで、Condition内の関数の引数としていたれまた。組み合わせお、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()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ンヨンは、はaltova:find-firstpair-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。

🗉 <u>サンプル</u>

- 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 ポジションには1 か返されます。第2 のサンプルでは、33 を返すペアがな サンカ、結果無しか返します。

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 つの別数を必要とます。最初の別数は1 つ、おは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 のアリテルを持つ)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れて、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 内インデングスポジョンを持ちます。
```

```
□ <u></u>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、sa とら名のヘ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 ひされ、代わりに、sa を入力値とま す。入力値は関数定義($sa \mod 2 = 0$)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 か存在する場合:

 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 も存在しな 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。

<u>上のサンプルこついての注意点</u>

- 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 を返する単一の引数を必要とます。(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あります。)
- 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(アリティー1)のみを必要とするからであり、item()を入力(URIと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、参照されているけって、呼び出されていない点に注意してくたさい。アタッチされている#1 サフィックスは関数が1つのアレティであることを表示するかっかです。doc-available#1の意味は以下のとおしです。アリティー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とて使用し、URIIボキュメトノードが存在するかテストするdoc-available()にいないます。1つが各当する場合、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ェンは、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:相対いなは、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、もう一つの属性が登番目の要素から取得された箇所で、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、ペアは名前別にコアルファベナ順にセナ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てない場合、ペアのジョイナト は解除されます。これは、一つのメンバーにより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(ジョイナ、ジョイトの解除)に個別に適用されます。

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) は次のアウションにより取得されます: (1+1, ()+2, 3+3, 4+())。演算子の一つか空 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になります。

• 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つの引数内の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なように結合されるように結 合されます。3番目の引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 か空の場合、SecondSequence を返します。FirstSequence か空では 場合、FirstSequence を返します。

□ <u>サンプル</u>

- 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び時刻データ型で保存されているデー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/xsltextensions に収められており、altova: フレフィックスが、このセクションでは使用されます。製品の今後の、デジョンか拡張機能への 継続的サポート、おけば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ナである 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。

🖃 <u>サンプル</u>

- 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 どる 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 キャメリケース 変換しま

す。SplitChars は IsRegex = true()の場合、おは、IsRegex = false()の場合、パ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 す。
- 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 に対しての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文字列内のPosition 引数にお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。引数にお提出されたインデックスに文字が存在してお場合、結果文字列は空です。

□ <u>サンプル</u>

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) をxs:string とする XP3.1 XQ3.1

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引数にお提出されたインデックスに文字が存在したは、場合、結果文字列は空です。

□ <u>サンプル</u>

- 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
XQ3.1
```

altova:create-hash-from-string(InputString as xs:string, HashAlgo as xs:string) asxs:string XP2 XQ1 XP3.1 XQ3.1

HashAlgo 引数によ指定されている いシュアルゴリズムを使用して InputString から いシュ文字列を生成します。次の いシ 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たさい)2番目の引数が指定されていない場合、SHA-256 / いシュアルゴリズムが使用され ます。

```
<u>サンプル</u>
```

- altova:create-hash-from-string('abc') はSHA-256 パッシュアルゴリズムを使用して生成されたパッシュ 文字列を返します。
- altova:create-hash-from-string('abc', 'md5') はMD5 パンシュアルゴリズムを使用して生成されたい シ シュ文字列を返します。
- altova: create-hash-from-string ('abc', 'MD5') は MD5 ハッシュアルゴリズ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:]

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 XQ3.1
```

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。

- 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
XP3.1 XQ3.1

```
入力文字列(最初の別数)には、配置・ラメーター(※1, ※2, etc)が含まれています。各・ラメーターは、(2番目の別数とて提出
されているフォーマトシーケンス内の対応するポジュ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、マ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
ケスか書式シーケンス内のアイテムの数よりに大きい場合発生します。
```

```
🗉 <u>サンプル</u>
```

- 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" を返します。
- 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)をxs:string とする XP3.1 XQ3.1 xs:string に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 文字を含んでいる文字列を返し ます。

🖃 <u>サンプル</u>

コ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 とする XP3.1 XQ3.1

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。

□ <u>サンプル</u>

- 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:]

```
altova:pad-string-left(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1文字です。文字列の左側ンジドされ、この数が、StringLength 引数の整数の値と等しくなるよう
にStringToPad の文字の数を増やします。StringLength 引数は任意の整数の値(正数まとは負数)を持つこができます。
```

- 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[パンズ字が長すぎます] エラーを返します。
- pad-string-right [altova:]

- 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[パボ文字が長すぎます] エラーを返します。

repeat-string [altova:]

altova:repeat-string(InputString as xs:string, Repeats as xs:integer) をxs:string とする

XP2XQ1XP3.1XQ3.1最初のInputString引数には構成される文字列、Repeats 回繰り返してを生成します。

□ <u>サンプル</u>

- 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)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 が発生した後のサブ文字列が 返されます。MainString 内でCheckString が検出されたは場合、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。

🗉 <u>サンプル</u>

- 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 か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。

- *□ <u>サンプル</u>*
 - 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 内で発生したい場合、整数 0 が返されます。第2 おけその後のStringToCheck,発生を確認すること、この関数の次の署名を確認してくたさい。

ヨ <u>サンプル</u>
```

• 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) 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 内で発生したは、場合、整数0が返されま す。

```
□ <u>サンプル</u>
```

- 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、先頭おは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、先頭おは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-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/XQuery 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 おさは(ii) Xquery ドキュメント内の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 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラレで使用可能 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、・・ジョンカ拡張機能への 継続的サポート、おけば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たさい。Altova 拡張機能へのサポートに関し ては、今後のリレ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

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い場合 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-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4

🗉 <u>サンプル</u>

- 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-エンコード済み文字列とて返します。
- 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ない場合 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-16, UTF-16LE, UTF-16BE, ISO-10646-UCS2, UTF-32, UTF-32LE, UTF-32BE, ISO-10646-UCS4

∃ <u>サンプル</u>

- altova:encode-string("Altova") はUTF-8 エノコード済み文字列 "Altova" の等価である base64Binary を返します。
- altova:encode-string("Altova", "UTF-8")はUTF-8 エロド済み文字列 "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
この関数は引数を必要とません。この関数は現在のユーザーの一時的なフォルダーへの ひを返します。

\Box  <u>サンプル</u>
```

- altova:get-temp-folder() はマンーン上で、xs:string とてC: \Users\<UserName>\AppData\Local\Temp\ と類似したのを返します。
- generate-guid [altova:]

```
altova:generate-guid() asxs:string XP2 XQ1 XP3.1 XQ3.1
ユニーケな文字列 GUID 文字列を生成します。

ヨ <u>サンプル</u>
```

- altova:generate-guid() は例えば 85F971Da-17F3-4E4E-994E-99137873ACCD を返します。
- high-res-timer [altova:]

```
altova:high-res-timer() asxs:double XP3.1 XQ3.1

秒数でシステム Yれぞりユーションタイマーの値を返します。/ Yれぞりユ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。
```

🗉 <u>サンプル</u>

- 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、おけよ 含およ 場合がおります。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 ツレを作成します。 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よめの実行を延期します。 *サンプル*

• altova:sleep(1000)は1000 ミトのため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「Jは XPath 2.0 / XQuery 関数、おけはXSLT 2.0 関数とて利用できない 関数が、くつかあ ります。そのような関数の良い例として、Java で利用することのできる sin()や cos()といた数学関数があります。XSLT スタイルシートやXQuery のカエリーてこれらの関数が利用できるのであれば、スタイルシートやクエリの適用範囲を大幅に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に軽減されます。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ご構成されます。

- <u>Java 拡張関数</u>
- .NET 拡張関数
- XSLT (文)するMSXSL スクノプト

記述の中では特に、())関連するライブラノ内の関数か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ことのようなルーノは適用され、returnにより値が返される際にどのような変換ルーノは適用されるのか、XSLT/XQueryデータオブジェクト」に対する関数の結果)について説明されます。

必要条件

拡張関数のサポートを有効にするコは XSLT 変換やXQuery の実行を行うエンピューターに 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、まけよアクセ スできる環境が整っている必要がおます。

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おか、Java のエレストラクターを呼び出したとり、Java の、静的おさまインスタンス)メノン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、ツンドとして扱われます。フィールドは静的また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して記述されます。

このセクションは以下のサブセクションには構成されます

- <u>Java:コンストラクター</u>
- <u>Java:静的メンドと静的フィールド</u>

- <u>Java: インスタンスメナンドとインスタンスフィールド</u>
- <u>データ型:XPath/XQuery からJava へ</u>
- <u>データ型: Java からXPath/XQuery へ</u>

以下の点に注意してくたさい

- Altova デストップ製品を使用している場合、Altova アプリケーションは、Java 仮想マシンへの やを(以下の順序で読み取ることにおり自動的に決知しようとします:(i) Windows レジストリ(ii) JAVA_HOME 環境変数。アプリケーションのオプションダイアログ内にカスタム やを追加することできます。このエントリは自動的に検知された他のJava VM / や以上の優先順位を有します。
- Altova サード製品がWindows マシン上で作動している場合、Java 仮想マシンへの ひはWindows レジストリから最初に読み取られます。成功したい場合、JAVA_HOME 環境変数が使用されます。
- Altova サーン製品を作動する場合、および、Linux おけまmacOS マシン上でサーンン製品を作動する場合、Java 仮想 マシンへの なが JAVA_HOME 環境変数内に保管されていることを確認してくたさい。 \bin\server おけま \bin\client ディノケリ内の jvm.dl1 ファイルを指している必要があます。

拡張関数のフォーム

XPath/XQuery 条件式におお拡張関数でよ prefix:fname()の形式を取る必要かがます。

- prefix: 部にと拡張関数が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 クラスの特定されます。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 メンドか識別され、メンドの引数が提供されます(以下の例を参照くたさい)。
 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せた場合、Java クラスの識別はクラスの前にくる
 fname() 部にて行うことはより、ビリドによりクラスから分離されることはよります(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ピューターのクラスマトレー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。

<xsl:value-of xmlns:jMath="java:java.lang.Math"
 select="jMath:cos(3.14)" />

拡張関数内にあるメンボ名(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 ンボの名前に一致する必要がおります。

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 クラスやメンドを作成した場合、(i) JAR ファイル(おけまにlass 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 おけま class)ファイルが、カレントディレクトリ(XSLT やXQuery ドキュメントが存在するディレクトリルコ配置されているか 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、ユーザー定義クラスファイルならび、ニューザー定義 JAR ファイルを参照くたさい。カレントディレクトリには無いクラスファイルやJAR ファイルへの、マレは指定しなければならないことに注意してくだ 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 スコース る XML ス キーマのトップレベルにある schema 要素 へ名前空間を追加する必要が決ます。 Style Vision にて作成された全ての SPS では、デフォルトで 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

- クラスファイルはパッケージである。XSLT おさま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)
- ケラスファイルは、シケージでは無い、XSLT おけま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 フルを参照)
- ケラスファイルは、ッケージである。XSLT おさはXQuery ファイルカランダムな場所に収められている。(下のサノプルを参照)
- ケラスファイルはッケージである。XSLT おけは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 おさはXQuery ドキュメン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、フォルダー内の全ク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こさいます:

java:classname

ニで、

java:によりユーザー定義のJava 関数が呼ばれていることか示されます(デフォルトでカレントディレクトリーある Java クラスかコードされます)。

classname は目的となるメンドのフラスか含まれているフラスの名前です。

ケラスか名前空間 URI Iこて識別され、名前空間がシンド呼び出しして使用されます。

クラスファイルがシッケージで、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 パッケージニあるCar クラスのgetVehicleType() メクドか呼び出されています。 com.altova.extfunc パッケージはJavaProjectと、う名前のフォルダーニ置かれており、XSLT ファイルも同じフォルダーニ配置されています。

```
<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>
```

```
</xsl:stylesheet>
```

クラスファイルか参照され、XSLT/XQuery ファイルかク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 ケラスのgetVehicleType() メンドが呼び出されます。以下を述べることができます:(i) Car ケラスは次のフォ ルダー内にかます: 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 内にかます。

```
<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:Car" >
<xsl:output exclude-result-prefixes="fn car xsl fo xs"/>
<xsl:template match="/">
    <a>
    <xsl:template match="/">
    <a>
    <xsl:value-of select="car:getVehicleType()"/>
</a>
```

```
</xsl:template>
```

 $\langle a \rangle$

```
</xsl:stylesheet>
```

クラスファイルかジッケージされ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 パッケージである 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 とう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います。この場合、以下のような構文でパッケージの場所をクエリ文字列とてURI内にて指定する必要があります:

java:classname[?path=uri-of-package]

ここで、

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-package 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。

<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"
xmlns:car="java:com.altova.extfunc.Car?path=file:///C:/JavaProject/" >
```

</xsl:stylesheet>

クラスファイルかジッケージではなく、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()メンドが呼び出されてます。 com.altova.extfunc パッケージはJavaProject とう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います。以下のような構文で、クラスファイルの場所をクエリ文字列としてURI内にて指定する必要があります。

java:classname[?path=uri-of-classfile]

ニで

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 は目的のメノバカ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。

```
<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:Car?path=file:///C:/JavaProject/com/altova/extfunc/" >
```

<xsl:output exclude-result-prefixes="fn car xsl xs"/>

</xsl:stylesheet>

メモ ノマカ外部関数により与えられている場合、ClassLoader によりマカ追加されます。

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 を指定する必要がみます:

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ドか呼び出されます: 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ります。
? はクラス名と な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 なが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 どなります。
!/ は、終了を表すデーをタとなります。
classNS:method() により、メノンドの呼び出しか行われます。
```

その他にも、メハボ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

```
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:template match="car"/>
```

</xsl:stylesheet>

```
メモ: 拡張関数によりなが与えられている場合、ClassLoader になが追加されます。
```

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」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、黙示的こ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 オジェイト がセンストラクター こと/作成されます。例えば、java.util.Date クラス ご対する エンストラクターが呼び出された場合(java.util.Date.new())、java.util.Date を持ったオブジェイトが返されます。返された オブジェイトのレキシカルフォーマナトは、XPath データ型のレキシカルフォーマナトにマッチしたは、場合もあり、目的の XPath データ型に対する レキシカルフォーマナーへ値の変換を行い、その後目的の XPath データ型へ変換を行う必要があります。

コンストラクターによれ作成された 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(<u>インスタンスメソッドならびドインスタンスフィールド</u>を参照ください):
 <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

静的メノバは Java 名ならびにメノバの引数により直接呼び出すことができます。E や PI といた定数の静的フィールド(引数を持たな しメノバ)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

静的メノバならびコノーール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ご注意してくたさい。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、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)を含むようご拡張されています。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 な静的メッチが引数ともに続くか(例: cos(3.14))、public な静的フィールが続きます(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みます:

```
<xsl:value-of xmlns:java="java:"
    select="java:java.lang.Math.cos(3.14)" />
```

XQuery の例 XQuery におち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いースされる Java オブジェクトが、インスタンスメンドには与えられています。このような Java オブジェクト 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・パラメーター/変数によい作成されます。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 クラスマオオるオブシェクトがコンストラクター(date:new() コンストラクター)ともに作成されます。
- 2. jlang.Object.getClass メンドの引数とて Java オジェクトが いースされます。
- 3. getClass メノバにお得られたオブジェクトがjlang.Object.toString メノバの引数とていースされます。

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 オブジェクト ではおひと ら点で、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れに引数として渡されますが、パラメーター/変数そのたのに Java オブジェクト から返された値が含まれてい る場合もあります。例えば、CurrentDate パラメーター ゴは java.util.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。 す。この値は引数として、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います。

Java では、以下のルールが適用されます。

- 同名のJava メノバカをつ以上あり、それそれが違う数の引数を受け取る場合、呼び出しに使用されている引数の数に一番マッチするメノバが選択されます。
- XPath/XQuery の文字列、数値、boolean データ型は、黙示的に対応する 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 型がつ以上の Java 型へ変換できる場合(例: xs:integer)、選択されたメノバに 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 メノバが 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。

以下のテーブルに、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 (プ ミティブ型) , java.lang.Boolean
xs:integer	int, long, short, byte, float, double, な らびにjava.lang.Integer のようなこれらのラッドークラス
xs:float	float (プ ミティブ型) , java.lang.Float, double (プ ミティブ型)
xs:double	double (プ ミティブ型) , java.lang.Double

xs:decimal	float (プ ミティブ型) ,	java.lang.Float,
	double(プミティブ型),	java.lang.Double

上のJストにある XML スキーマデータ型(ならび)こ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 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 メンド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ょう:

- 与えられた引数が10とう値を持すこxs:untypedAtomic型で、mymethod(float)メンドへ渡されるのを意図している。
- しかし、そのプラスココお別のデータ型を取る mymethod (double) とうメンバも存在する。
- メノデ名か同じで、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ため、 xs:untypedAtomicがfloatではなくdoubleに変換される可能性もある。
- 結果として、意図したメソンドは選択されず、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 ソンドを使用するユーザー定義のメソンドを別の名前で新たコン作成する必要があります。

上のノストでカントされていない型(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いうことも留意してくがさい。

12.2.2.1.7 データ型: Java から XPath/XQuery へ

Java メハボには、値のデータ型が文字列、数値、おけま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ため、ラップして使用するべきでしょう。

Java オジェイトや文字列、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 ンドを使用して Java オブジェイトを文字列へ変換することで、目的のXPath/XQuery 型への変換を行います。XPath/XQuery で は、文字列を目的となる型のレキンカルフォーマナーへ変換し、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 プラオ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 おうよインスタンス変数)プロン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おり.NET クラスのプロン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と、構成されています

- <u>.NET エレストラクター</u>
- <u>.NET:静的メノメ ならび 静的フィール</u>
- <u>.NET: インスタンスメンドとインスタンスフィールド</u>
- <u>データ型:XPath/XQuery から.NET へ</u>
- <u>データ型:.NET からXPath/XQuery へ</u>

拡張関数のフォーム

XPath/XQuery 条件式にある拡張関数は、prefix:fname()の形式を取る必要かあります。

- 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 クラス内にあるコンストラクター、プロ/ ティ、おさは(静的おささんスタンス) メノバガ特定され、必要な場合にお|数が与えられます。
- URI は clitype: で開始する必要があり、これにより関数が.NET 拡張関数であることか認識されます。
- 拡張関数の prefix:fname() 形式は、システムクラスならびコードされたアセンブルともコ使用することもできます。しかし ク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、ラメーターが必要しています。

パラメーター

アセンブをロードするコお以下のウメーターを使用してくたさい

- asm ロードするアセンブの名前。
- ver バージョン番号(ピリボによ)分離された最大4桁の整数)。

sn アセンブ厳密名のキートークン(16新数の数値)。

from ロードするアセンブリ(DLL)の場所を特定するURI。URIか相対パマの場合、XSLT やXQueryドキュメントに対して相対的にないます。この、デメーターか指定された場合、 その他の、デメーターか無視されます。

partialname アセブ児名の一部。Assembly.LoadWith.PartialName() へ渡され、アセブ リークロードか試みられます。 partialname が指定された場合、その他の デメータ ーが無視されます。

loc 例えばen-US とらロケール。デフォルはneutral です。

アセンブがDLLからロードされる場合、from ノラメータが使用して、snノラメーターは使用しないでくたさい。アセンブがジローノシルアセンブドマンユ(GAC)からロードされる場合、snノラメータを使用してfromノ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セミコロンで分離する必要があります。パラメーター名へ値を受け渡すには、統合符号(=)を使用します(以下の例を参照ください)。

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、ロードするクラスを Trade.Forward.Scrip と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、システムクラス 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:

```
<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">
<xsl:nutput method="xml" omit-xml-declaration="yes" />
<xsl:output method="xml" omit-xml-declaration="yes" />
<xsl:template match="/">
<math xmlns:math="clitype:System.Math">
<sqrt><xsl:value-of select="math:Sqrt(9)"/></sqrt>
<pi><sqrt><xsl:value-of select="math:PI()"/></pi>
<pw><xsl:value-of select="math:PI()"/></e>
<pw><xsl:value-of select="math:Pi(), math:E())"/></pow>
</math>
</xsl:template>
```

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 か関連付けられます。URI の最初に ある clitype:により、それ以降の記述がシステムクラスませまロードされたク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(そしてクラス)System.Math に関連付けられます。拡張関数により、 System.Math クラス内のメンドか特定され、必要な場所に引数か与えられます。

XQueryの例 上のXSLT(マオする例と同様のXQuery例を以下に示します:

```
<math <pre>xmlns:math="clitype:System.Math">
    {math:Sqrt(9)}
```

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 ソドが特定され、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。クラス内に2つ以上のコンストラクターかある場合、与えられた引数の数か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か見つかびよ場合、、No constructor found ' エラーか返されます。

582 付録

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 コンストラフ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ップした.NET オブ ジェクト がコンストラフ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フターが、System.DateTime.new() により、呼ばれた場合 System.DateTime 型を持オー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かあります。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 データ型ヘキャストする必要かあります。

コンストラクターにより作成された.NETオブジェクトーズ北て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)"
 xmlns:date="clitype:System.DateTime" />
- 拡張関数へ渡すことができます(<u>インスタンスメソッドとインスタンスフィールド</u>を参照ください):
 <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
 xmlns:date="clitype:System.DateTime" />
- 文字列、数値、おはboolean ~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)))" xmlns:date="clitype:System.DateTime" />

12.2.2.2.2 .NET:静的メソッドと静的フィールド

メンボ名と引数を与えることで、静的メンボを直接呼び出すことができます。呼び出しに使用される名前は、クラス内にある public static メンボと完全に一致する必要があります。関数の呼び出しに使用されたメンボ名と引数の数にマッチするものかう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。

メモ: .NET クラス内にあるフィールドは引数を持たない ベンバとしてみなされます。プロイライは get_PropertyName()構文によ リーダび出されます。

例

1つの引数ともレントボ (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"/>

(引数なしのメノボとてみなされる)フィールド (System. Double. Max Value () 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/ディ(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つの引数ともにベンド((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

インスタンスメンドはよ メンド呼び出しの第一引数として、NET オブジェクトが渡されます。通常この、NET オブジェクトは、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・ウメーター/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 コントラクター(new(2008, 4, 29))が、System.DateTime 型の.NET オブシェクトの作成 「使用されます。このオブシェクトは、最初にreleasedate 変数の値とて、次にSystem.DateTime.ToString() メンドの引 数とて作成されます。System.DateTime.ToString() インスタンスメンドは、System.DateTime コンストラクターの (new(2008, 4, 29))におお引数とて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 オブジェクトを 取得するのに使用されます。

インスタンスメソッドとインスタンスフィールド

インスタンスペッドとインスタンスフィールドの違いお理論的なものです。インスタンスペッドでは、NET オブジェクトか直接引数に渡され、イン スタンスフィールドでは、デメーターや変数が、NET オブジェク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 オブジェクトが含まれており、この変数が2番目のdate 要素エンストラクターにて ToString() の引数と して渡されます。そのよめ、最初のdate 要素にある ToString() インスタンスがインスタンスペットドであるのに対し、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。

12.2.2.2.4 データ型: XPath/XQuery から .NET へ

.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 メッドのうち、どれか呼び出されたのか決定するのに関数の引数に使用されるデータ型か重要になります。

.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が2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決められます。
- 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)ストを参照)。与えられた XPath/XQuery 型が2つ以上の.NET 型へ変換できる場合(例: xs:integer)、選択されたメノ・ド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 メノ・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	IntegerValue, decimal, long, integer, short, byte, double, float
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いNET メン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ょう:

- 与えられた引数が10あるxs:untypedAtomic 値で、mymethod(float) メンボへ渡されるのを意図している。
- しかし、そのケラスコお別のデータ型をとる mymethod (double) とうペンドも存在する。
- メノボ名が同じで、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なさめ xs:untypedAtomicがfloatではよくdoubleに変換される可能性もある。
- 結果として、意図したメソンドは選択されず、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 ソンドを使用するユーザー定義のメソンドを別の名前で新たコン作成する必要があります。

上のパトでかいーされていた型(例:xs:date)は変換されずエラーとないます。

12.2.2.2.5 データ型: NET から XPath/XQuery へ

.NET メンボには、植が返される際に植のデータ型が文字列、数値、おけま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、.NET のdecimal データ型はxsd:decimal へ変換されます。

.NET オブンエクトや文字列、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() といた)、NET メンドを使用して、NET オブジェク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 フォーマナーへ変換し、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 条件式内部から呼び出し を行うことができます。<msxsl:script> はップレベル要素で、<xsl:stylesheet> おは<xsl:transform> の子要素で ある必要があります。

<msxsl:script> 要素はurn:schemas-microsoft-com:xslt 名前空間内に存在する必要があます(以下を参照くたさい)。

スクリプト言語と名前空間

ブロック内で使用されるスクリプト言語は<msxsl:script>要素のlanguage 属性にて指定され、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 属性に以特定されます(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W indows Scripting Runtime を使すたやとを行うかれたしいのコンピューターにクンネールされた言語だけ、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 以上のプラットフォ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 がデンオルトとして想定されます。

implements-prefix 属性はは名前空間スコープ内で宣言されたプレフィ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のために 予約されたユーザーの名前空間となります。<msxsl:script>要素内で定義された全ての関数ならびに変数は、implementsprefix 属性にて指定されたプレフィ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 修飾関数名か同じ名前空間内に関数として定義されていなければなりません。

サンプル

<msxsl:script> 要素内で定義された関数を使用するXSLT スタイルシートの例を以下に示します:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:msxsl="urn:schemas-microsoft-com:xslt"
    xmlns:user="http://mycompany.com/mynamespace">
```

```
<msxsl:script language="VBScript" implements-prefix="user">
```

<! [CDATA[

- ' Input: A currency value: the wholesale price
- ' Returns: The retail price: the input value plus 20% margin,

```
' 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>
          <b>Total Retail Price =
            $<xsl:value-of select="user:AddMargin(50)"/>
          </b>
          < br/>>
          <b>Total Wholesale Price =
           $<xsl:value-of select="50"/>
          </b>
        </p>
      </body>
    </html>
  </mmsl:template>
</xsl:stylesheet>
```

データ型

スクリプトブロックとのやりとりに使用される、ラメーターの値は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りとら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、アセンブをスクレプト内部へんポートすることができます。アセンブは名前やURI により特定されます。アセンブのんポートは、コントイル時に行われます。以下に msxsl:assembly 要素の簡単な使用例を示します:

```
<msxsl:script>
        <msxsl:assembly name="myAssembly.assemblyName" />
        <msxsl:assembly href="pathToAssembly" />
```

• • •

</msxsl:script>

アセンブ 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"

"myAssembly.Draw" のような短い名前でも指定できます。

名前空間

msxsl:using 要素によれ前空間の宣言を行うことができます。これにより、スクレプト内において名前空間無しでアセンブル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:

<msxsl:script>

<msxsl:using namespace="myAssemblyNS.NamespaceName" />

. . .

</msxsl:script>

namespace 属性の値は名前空間の名前とおます。

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、下に挙げたれるDB で使用されます。下のリストから使用するDB を選択してください。

- <u>ADO</u>
- MS Access
- MS SQL Server
- MySQL
- <u>ODBC</u>
- Oracle
- <u>Sybase</u>

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
smallmoney datetime	xs:decimal xs:dateTime
smallmoney datetime smalldatetime	<pre>xs:decimal xs:dateTime xs:dateTime</pre>
smallmoney datetime smalldatetime binary	<pre>xs:decimal xs:dateTime xs:dateTime xs:base64Binary</pre>
<pre>smallmoney datetime smalldatetime binary varbinary</pre>	<pre>xs:decimal xs:dateTime xs:dateTime xs:base64Binary xs:base64Binary</pre>
<pre>smallmoney datetime smalldatetime binary varbinary image</pre>	<pre>xs:decimal xs:dateTime xs:dateTime xs:base64Binary xs:base64Binary xs:base64Binary</pre>
<pre>smallmoney datetime smalldatetime binary varbinary image integer</pre>	<pre>xs:decimal xs:dateTime xs:dateTime xs:base64Binary xs:base64Binary xs:base64Binary xs:integer</pre>
<pre>smallmoney datetime smalldatetime binary varbinary image integer smallint</pre>	xs:decimal xs:dateTime xs:dateTime xs:base64Binary xs:base64Binary xs:base64Binary xs:integer xs:integer
<pre>smallmoney datetime smalldatetime binary varbinary image integer smallint bigint</pre>	<pre>xs:decimal xs:dateTime xs:dateTime xs:base64Binary xs:base64Binary xs:base64Binary xs:integer xs:short xs:long</pre>

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 の別にチェンク制約が適用されている場合、オニチェンク制約が0 おけま1 の値をチェンクする場合、この別の 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

12.4 技術データ

このセクションは、ソス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。以下のように整理されています。 OS ととモノ要件Altova XML バノデーターAltova XSLT とXQuery エンジンUnicode のサポートインターネトの使用

12.4.1 OS とメモリ要件

オペレーティングシステム

Altova ソストウェアアプケーションは、以下のプラナ・フォームでご使用レサオミオます: プラナ・フォーム更新済みのWindows 7 SP1、Windows 8、Windows 10プラナ・フォーム更新済みのWindows Server 2008 R2 SP1 おとは以降 メモリ

ソフトウェアがC++で書かれているすめ、Java Runtime Environmentをダウンロ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、通常少ないメモルを必要とします。しかしながら、各ドキュメントは完全に解析するすめ、また、ビューと編集の速度を向上するすめにメモ リニダウンロードされます。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クションの切り取り、貼り付け操作を繰り返し行うと、使用できるメモルがすくに消費されます。

12.4.2 Altova XML バリデーター

XMLドキュメートを開くとアプリケーションは、内蔵のXML パリデーターを使用して、指定されている場合、スキーマイン対して整形式をチェック、ツレーとインフォセナを作成します。XML パリデーターは、ドキュメントを編集する際にインデルジェントな編集へルプを提供し、発生する検証エラーを表示するオーカニで使用されます。

内蔵のXML バデーターは 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より勧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。各エンジンのオーダントと実装に固有の振る舞いて関しては、製品で使用されるエンジンの各ドキュメントの付属書(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 が必要です。

フォントの多くは、Unicode 範囲全体の特定のサブセナを含む場合があり、このため、通常は対応する表記システムをターゲオとします。 テキストの一部が、文字化しれて表示された場合、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れ、場合があたれます。です から、特に、異なる言語、ませま、異なる言語システムのXMLドキュメントを編集する場合、範囲全体をカバーするフォントを使用することが 役はたちます。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。

アプリ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でき、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8つのフィールドが通常のhttp(ポート80)接続を使用し、サーバーは転送され、無料の評価キーか顧客に通常の 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、イクーネト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、 ドキュメトは、次のプロトコレダンドと接続の「つを使用して取得されます: HTTP(通常、ポト 80)、FTP(通常、ポト 20/21)、HTTPS(通常、ポト 443)。HTTP サードをポト 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トを指定します)。
- XML スキーマ、おは、DTD を参照する XML ドキュメト、と、URL によ 指定されてい、ド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(ポト 80) おけま URL によ 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 抽出されます。 XML ファイル お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。
- WSDLとSOAPを使用するAltovaアプケーションでは、Webサービスを使用する接続は、WSDLドキュメントに比定 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、ユーザーのマンノニインストールされているMAPI コンプライアント電子メールプログラムにお送信されます。
- ソトウェアの荒レセス認証とLiveUpdateの一部として、Altovaソト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。

12.5 ライセンス情報

このセク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に関する情報
- 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
- ソフト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 ライセノ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たさい。

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、以下のユニーグよりトカあります。

- 購入を決定する前に、無料でノ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、<u>Altova Web サイト</u>にて注文を行います。すく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。
- オンライノニで注文を行うと、常に最新のノストウェアをご利用いただけます。
- 製品パケージョンのユーザーマニュアルは <u>https://www.altova.com/ja/</u>上にあり、(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 印刷に対応しておます。

30日間の評価期間

この製品をダンロードした後は、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4項から、製品からイセス登録 されていないことがノスト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びに表示され、30日間を超えてプログラムを 使用するコよ、キーコードを含むライセノスファイルから提供される製品のライセノスを購入します。ライセンスファイルを製品のノストウェアアクティ ベーションダイアログにアップロードして、製品をアノロックします。

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、おけよクターネトに接続されていないロ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、セントアッププログラムけっか配布を行うことが可能です。ソス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のノストウェアアウティベーションの一部として、ソストウェアにより内部やトワークまけはインターやホーへの接続を行し、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バ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防ぎ、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

日付/時刻、ソフトウェアの、ージョン、コ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・ユ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てとが保証されます。

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、別々のエン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リケーションが起動すると、ソフトウェアにお短いゲータグラムがブロードキャストにお送信され、同一の ネッ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ずみかのチェックが行われます。応答が無い場合は、アプリケーション の他インスタンスから送信される信号に広えるさっな、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、スタートアップ時に、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どとが行われ、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たよい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ユーザーはノーズナブルな価格にて同時使用マルチュ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、ゲットを送信することで、ネトワーク、マ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(<u>IANA Web サイト</u> (<u>http://www.iana.org/)</u>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エレビ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ラ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スペクをブロックすることは勿論可能です。

証明書に関するとモ

Altova アプケーションはHTTPS を介して Altova ライセスサーイ・(link.altova.com) に通信します。この通信のすめに Altova は登録済みのSSL 証明書を使用します。(例えば、社内 IT 部署おけおト部エージェンシー(こり)この証明書が置き換えられて いる場合、使用中の Altova アプリケーションは接続が安全では、こを警告します。 Altova アプリケーションを開始するけっけに仕替の証明 書を使用することができますが、自己責任で行ってくけさい。 安全でけない 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おけ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けさい。

(例えば、クライアトマシンへの、おけば、クライアトマシンへの通信を監視するけっかに)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 <u>Altova LicenseServer</u> を使用中のホットワークにインストールすることが奨励されます。このセットアップでは、 Altova LicenseServer は Altova との通信のけっかに Altova 証明書の使用を許可しつつクライア・トマシ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 エイューザー使用許諾契約書: http://www.altova.com/ja/legal/eula
- Altova プライバシーポルシー: <u>http://www.altova.com/ja/privacy</u>

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

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

Η

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

I

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 と 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

Μ

Microsoft Office 2007 (Enterprise Edition のみ), 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 のみ), 15, 29 OOXML (Enterprise Edition のみ), 15, 29 OS,

Altova 製品のための, 594

Otherwise 条件 分岐, 233

Ρ

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 Authentic View (Enterprise と Professional Edition), 19 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 StyleVision, ユーザー マニュアル, 13 製品の機能,15

Style Vision Power Stylesheet, SPS を参照, 13 Style Vision について, 489 Style Vision に関する良くある質問, 489 Style Vision のセットアップ, 21 Style Vision のためのサポート, 489 Style Vision のための技術サポート, 489 Style Vision Batch, 21, 366

Г

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

Х

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

Ζ

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 StyleVision, 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 StyleVision 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

テーブル. 行の追加.464 行の展開/折りたたみ,137 作成,444 条件付きの処理,130 静的な テーブルの挿入.463 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 編集 プロパティの.467 列の挿入,464 列の追加,464 列を非表示にするボタン,137 テーブル (SPS). プロパティの編集,397 編集のためのツールバー ボタン, 397 テーブル セルコンテンツの配置. SPS 内. 468 テーブル メニュー, 463 テーブルセルの分割, 左右,466 上下,466 テキスト参照,282 デザイン コマンドの保存, 415 デザイン ツリー. サイドバー ウィンドウ, 39 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, スイッチオンとオフ,435 デザイン フィルター, スイッチオンとオフ,435 デザイン フラグメント. 挿入,453 デザイン 概要. サイドバー ウィンドウ, 33 デザイン 構造, 175 デザインツリー, デザイン入力ヘルパーも参照,31 デザインビュー. SPS デザイン内のシンボル, 376 と JavaScript エディター, 29 マークアップの表示,29 詳細.29 デザインビュー内のシンボル. XML ドキュメントコンテンツの, 376 XML ドキュメントノードの, 376 イメージの.376 データ-エントリデバイスの, 376

ハイパーリンクの、376 ブックマーク (アンカー)の, 376 自動計算の.376 条件付きのテンプレートの,376 定義済みフォーマットの,376 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 表示.131 デザインビュー内のマークアップタグ、29 デザインビュー内の入力ヘルパー, 表示のオンとオフの切り替え,435 デザインフラグメント,212 デザイン入力ヘルパー, 自動的に非表示.31 詳細.31 非表示,31 表示のオンとオフの切り替え,435 デザイン入力ヘルパー ウィンドウ. ドッキング,31 フロート, 31 デザイン要素,399 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 テンプレート. タグで囲まれた、456 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 と一致するノードの変更,172 のツリー, 39 ビューのオンとオフの切り替え,435 插入,450 テンプレート XML ファイル (Enterprise と Professional Edition), 92 定義.22 テンプレート フィルター, 401 テンプレートー致 コマンドの編集, 117 ドキュメント. 開くと閉じる,28 概要.25 ドキュメント スタイル, 225 ドキュメント プロパティ, 225 ドキュメント 要素 (ルート 要素参照), 177 ドキュメントノード. 定義.22 ドキュメントビュー, GUI内,28 ドキュメント要素, 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 と 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 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 プロパティ入力ヘルパー. 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 插入 | ハイパーリンク、447 挿入 | ブックマーク, 446 插入 自動計算, 439 挿入 | 条件, 449 插入 | 水平線, 443 挿入 | 段落, 440

插入 特殊な 段落 440 メモリ要件,594 モジュール. 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 (静的と動的),444

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. StyleVisionBatch 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 出力-エスケープの無効化,293 出力ビュー, 詳細.29 出力ファイル,

(C) 2015-2021 Altova GmbH

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 出カ-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

(C) 2015-2021 Altova GmbH