

Altova XMLSpy 2017 Enterprise Edition

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.

発行日:2017

(C) 2017 Altova GmbH

目次

1	XMLSpy インターフェイス	2
1.1	表示	
1.2	ウィンドウ	
1.3	メニューとツールバー	
1.4	テキストビューの設定	
2	XML スキーマ :基本	13
2.1	新しい XML スキーマファイルの作成	
2.2	名前空間の定義	
2.3	コンテンツモデルの定義	
2.4	ドラッグアンドドロップによる要素の追加	
2.5	コンテンツモデルビューの構成	
2.6	基本的なスキーマの完成	
3	XML スキーマ∶高度な機能	27
3.1	複合型と単純型を扱う	
3.2	グローバル要素の参照	
3.3	属性と列挙型	
4	XML スキーマ :XMLSpy の機能	38
4 4.1	XML スキーマ :XMLSpy の機能 スキーマナビゲーション	38
4 4.1 5	XML スキーマ :XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション	38 39 41
4 4.1 5 6	XML スキーマ :XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント	38 39 41 45
4 4.1 5 6 6.1	XML スキーマ :XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成	38
 4 4.1 5 6 6.1 6.2 	XML スキーマ :XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定	38
 4 4.1 5 6 6.1 6.2 6.3 	XML スキーマ:XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力	38
4 4.1 5 6 6.1 6.2 6.3 6.4	XML スキーマ:XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力	38
4 4.1 5 6 6.1 6.2 6.3 6.4 6.5	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 ドキュメントの検証	38 39 41 45 46 48 50 51 51 54
4 4.1 5 6 6.1 6.2 6.3 6.4 6.5 6.6	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 ドキュメントの検証 要素と属性の追加	38 39 41 45 46 48 50 50 51 54 58
4 4.1 5 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 ドキュメントの検証 要素と属性の追加 データベース /テーブルビューでの編集	38
4 4.1 5 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 ドキュメントの検証 要素と属性の追加 データベース /テーブルビューでの編集 スキーマの修正	38
 4 4.1 5 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 7 	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマトキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 ドキュメントの検証 要素と属性の追加 データベース /テーブルビューでの編集 スキーマの修正 XSLT 変換	38 39 41 45 46 48 50 51 51 54 58 60 64 66
 4 4.1 5 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 7 7.1 	XML スキーマ: XMLSpy の機能 スキーマナビゲーション スキーマドキュメンテーション XML ドキュメント 新規 XML ファイルの作成 要素の型の指定 グリッドビューを使ったデータの入力 テキストビューを使ったデータの入力 データベース /テーブルビューでの編集 スキーマの修正 XSLT ファイルの割り当て	38 39 41 45 46 48 50 51 51 54 58 60 64 66 67

Inde	x	7	75
9	おしまい	-	74
8.2	プロジェクトの作成		72
8 8.1	プロジェクトの利点		7U 71
0	プロジェクト答理		70
7.3	XSLファイルの修正		69

XMLSpy チュートリアル

このチュートアルではXMLの概要を理解し、数々の重要なXMLのタスクを体験することができます。これにより XMLSpyの最も重要な機能を学習します。

このチュート」のルは以下のようご構成されます:

- XMLSpy <u>インターフェイス</u> アプガーショのグラフィカルユーザーインターフェイス (GUI) を詳しく理解することができます。
- XML <u>スキーマの作成</u>。XMLSpy の直感的なスキーマビューこて XML スキーマを作成する方法、ドラッグアンドトロップによ複雑なコンテンバモデルを作成する方法、そしてスキーマの設定方法を学習します。
- <u>スキーマビューの機能を使用</u>。複合型ならバビ単純型、グローバルな要素参照、そして属性の列挙の作成方法を学習します。
- スキーマビューを使た、スキーマのナビゲーション方法、スキーマのナビゲーション方法を学習します。
- XML ドキュメントの作成。スキーマをXML ドキュメントに割り当て、グリットビューやテキストビューを使ってXML ドキュ メントを編集する方法、そして XMLSpy に組み込まれているパレデーターを使用して、XML ドキュメントを検証する方法を学習します。
- XSLT スタイルシートを使用してXML ファイルを変換する XSLT ファイルを割り当て XMLSpy に組み込まれた XSLT エンジンを使用することで変換を引ます。
- XMLSpy プレジェクトにで作業することにより、XML ドキュメントを簡単に整理することができます。

イノストールと設定

このチュートアルはあれたのコンピューターにXMLSpyがインストールされており、無料評価キーコートを受け取っているか、登録されたユーザーであることを前提としています。XMLSpyの評価版の機能は製品版と変わりませんが、その使用は30日間に限定されています。XMLSpyの一般ライセンスはAltovaのセキュアーウェブサーバ、または販売店からお求めになれます。

チュートリアリファイル

チュートレアレファイルはアプケーションフォレダにあります:

<code>C:\Documents</code> and <code>Settings\<username>\My Documents\Altova\XMLSpy2017\Examples \Tutorial</code>

Examples フォレダーは抱々なこと気はすためのXML ファイルが配置されており、Tutorial フォレダーはよ このチュートアレアで使用される全てのファイルが含まれています。

アプケーションフォレダー内の Template フォレダー 通常は c:\Program Files\Altova しなします) はは ワァ イル 新規作成 」 メニューオプショカ ら選択 できる全てのXMLテンプレートファイルが収められています。 これらのファイルに は必要なデータ (名前空間とXML 宣言)が記述されており、すぐご作業を始められるようこなっています。

1 XMLSpy インターフェイス

チュートアルのこのセクションで、XMLSpyを開始してインターフェイスについて理解することができます。

XMLSpy の開始

XMLSpy を開始するは、デスクトップのXMLSpy アイコンをダブルクリックするか、または スタート すべてのプログラム」メニューを使用してXMLSpy プログラムコンクセスします。XMLSpy は ドキュメントがインターフェイス上で開かれていない、 状態で開始されます。 XMLSpy を開きます。

インターフェイスの概要

XMLSpy インターフェイスのデフォレトのビューは3つの垂直のエレアによ構成されています (下のフィギュア)の 3つのウシンドプによ構成されています: () プロジェクトと情報ウィンドブ; (i) メイン出力ウンドブ; そして (ii) 入力ヘレパーウンドウ、プロジェクトウンドウを確認してなさい。 XMLSpy を最初に開始した時にデフォレトで開かれる サンプレプロジェクトか表示されています。

以下は、インターフェイスのレイアウトと多種のエレポーネントの機能を理解するために役に立つ主要なポイントです。チュートリアルのこの最初のパートのサブセグションはインターフェイスに貫れ親したために役に立ちます。

メニューバー		
ツールバー		
プロジェクト ウィンドウ	メインウィンドウ	入力 ヘルパー
ステータスバ	ー:現在選択されているノードへの	D XPath

メインウィンドンカのドキュメントパー:複数のトキュメントが開かれて、は場合、各トキュメントはメインウィンドンのトキュメントパーのタブに表示されて、ます、図を参照)。タブをクリックすることでトキュメントをアケティブ化することができます。 ンドバーの右側の矢印をクリックすることでトキュメントタブをスクロールすることができます。(例えば、Examples プロジェクトから)2つ以上のファイルを開き、タブの動作について確認してくたさい。

ドキュシント編集ビュー:アケティブなドキュメントは適用することのできる複数の編集ビューの1つにより表示することができます。例えば:

- XML (.xml) ドキュメントはテキストビュー、グリッドビュー、Authentic ビューで表示されますが、スキーマビューなどの他のビューでは表示することができません。
- XML スキーマ (.xsd) ドキュメント、他方、テキストビュー、グリッドビュー、スキーマビュー、プランザービューで表示されますが、Authentic ビューでは表示することができません。

以下のビューを使用することができます:テキストビュー、グリッドビュー、スキーマビュー、Authentic View- ビュー、お よびブランザービュー。 人力へルパー:入力へルパーウシンドウはアケティブなドキュメント(例えばXML おけはXSD おけはCSS おけは WSDL) および、現在アケティブなドキュメントビュー(例えば、テキストビューおけはスキーマビュー)に従い 変更されます。入 カへルパーによりアケティブなドキュメントをコンテキストに敏感な編集サポートにより素早 のつ正確に編集することができま す。

1.1 表示

チュートアルのコのパートでは以下を学ぶことができます:() ドキュメントの編集ビューの切り替え および (i) 特定のドキュ メントの型のデフォレトの編集ビューの変更。

ドキュメントビューの切り替え

トキュメントを開くとそのトキュメントの型のために設定されたデフォルトのビューで開かれます。トキュメントを以下の方法で開きます:

- 1. 「ファイル」開くコマンドをクリックします。
- 2. C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2017 \Examples\Tutorial フォレダーにあるファイルAddress.xsd を参照し選択します、開く をクリックする とファイルがスキーマビューで開かれます。
- メインウィンドカフトにあるビュータブをクリックして多種のビュー (Fキストビュー、グリッドビュー、など)を引着えることができます。
 XML スキーマドキュメントはデキストビュー、グリッドビュー、スキーマビューおよびブランザービューで表示されます。
- 4. 表示」メニューのオデンョからビューのオデンョンを選択してビューを変更することができます。表示」メニュースマントを使用してAddressFirst.xsd ドキュメントのビューを引着えて代えい。
- 5. ファイル 閉じる」を使用してドキュメントを閉じてたさい

ドキュメントの型のデフォルトのビューの変更

.xsd 拡張子を持つすべてのドキュメントは、デフォレトでスキーマビューで開かれます。全ての型のドキュメントのデフォレトで開かれるビューをオプションダイアログで変更することができます。ここでは、xsd ドキュメントを使用して行います。

- 1. 「ソール」オプション」コマンドをクリックレ、ファイルタイプタブに移動します(下のスクリーンショント参照。)
- 2. 「ファイルタイプ」ペインで .xsd にスケロールして選択します (スクリーンショントで) イライトされて ほす)。
- 3. デフォルトのビュー」ペインでテキストビューを選択します。

オプション	
色 Iンコード XSL XSL スト ファイル ファイルの型 編集 ビュー !	2-ド オプティマイザー XQuery スクリプト タクソノミパッケージ ソース管理 ブリッド フォント スキーマ フォント WSDL フォント XBRL フォント テキスト フォント
ファイルの型 txt vml vxml vxml Uxml	ラ−設定 IC XML Schema tt/xml
wml wsdl xbrl xdr xdr xhtml wsw xhtml wbw	デフォルト ビュー ◎ グリッド ビュー ◎ スキーマ ◎ テキスト ビュー ■ テーブルを自動的に構築 テキスト ビュー
Xisx ② XML エンテイ xml xq ③ XQuery 準拠 xq ③ XQuery 準拠 xq ③ XQuery Upp xqr ④ XQuery Upp	^{r1} ◎ ブラウザービュー ◎ WSDL ◎ XBRL タクソノミ
xquery xsd 単 xsig xsig xsi xsit zip ▼	 ■ 自動検証を無効化 ✓ 空の要素を省略して "<e></e>> " で保存する ■ RaptorXML Server を使用して [県<!--保存] を検証す</li-->
新たなファイル拡張	子を追加 選択されたファイル拡張子を削除
	OK キャンセル 適用(A)

- 4. **OK」**をクリックします。
- 5. 「ファイル」開くコマンドをフリックして、AddressFirst.xsd ファイルを開きます。ファイルはテキストビューで開かれます。
- 6. スキーマビューは小替えて、ファイルをコンビューで表示し、ファイルと、ファイルと閉じる」を使用して閉じます。
- 7. 「シール」オプション」を使用して、オプションダイアログに戻り、ファイルタイプタブで、xsd ファイルのデフォルトのビュ ーをスキーマビューは戻します。
- 大モ: オプションダイアログのファイルタイプのタブで(上のスクリーンショント) 「ストされるファイルの拡張子のデフォレトのビューを変更することができます。ファイルの拡張子は、新たな拡張子を追加」ボタンを使用して「ストは追加することができます。

1.2 ウィンドウ

デフォルトでは、複数のウィンドウがメインウィンドウカ周リに表示されて(下のスクリーンションイ参照)次のウィンドウカブループに整理されています:

- プジェクトウィンドウ
- 情報ウシドウ
- 入力へリパー 現在アケティブなトキュメントの型により複数存在します)
- 出力ウインドウ:メルセージ、XPath、XSL アウトライン、ファイルの中を検索、スキーマの中を検索、



このセケノョンでは、ウインドンのグループの表示のオンとオス、ウインドンのスクレーン上の移動方法の説明を引ます。ウインドンの表示の管理は、インターフェイス上のスペーズ管理が必要な場合役に立ちます。

ウインドウグループの表示のオンオフの切り替え

ウシンドウガレープ(プロジェクトウシンドス、情報ウシンドス、入力ヘリパー、出力ウシンドウ)は ウシンドウ メニューのエマンド でオンとオフを加加着えることにより、表示ませる非表示にすることができます。表示されたウシンドウガレープはタイトリバーを右 クリックルてニマンド 非表示 を選択することで非表示にすることができます。非表示のウシンドウは ウシンドウ メニューを使用 してのみ表示することができます。

C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2017\Examples \Tutorial フォレダー内のXML ファイルを開き、基本のコマントの使用を練習し、コマントの作動方法に貫れ親しんでく ださい。ウインドングループ表示と非表示に関しての詳細は、XMLSpy インターフェイスのセグションを参照してください。

スクリーン上でウィンドウを移動する

個別のウンドガは インターフェイス上を自由にスロートませはインターフェイス内でドッキングさせることができます。 ウンドガは ウンドウガループ内のタブとしてドッキングすることができます (ウィンドウグループは上で説明されてします)。 例えば 下の スクリーンショントでは 320のタブ (グローバルウンド文 名前空間ウンド文 ID 制約)付きのウンンドンを持つ、ユポーネ ントの入力ヘルパーがスキーマビューで表示されています。

コンポーネント エ	x
⊞element (12) ⊞complexType (2) ⊞simpleType (46) ⊞attribute (2) ⊞attributeGroup (2)	
グローバル 名前空間 同一性制約	

ウシンドウは、以下の方法を用いてフロートまたはドッキングすることができます:

- ウシドカワイトリバーをダブルクリックします。ドッキングすると、ウシンドガはアロートすることができます。アロートして いると、ウシンドガは最後にドッキングしていた箇所にドッキングします。
- ウインドウのタイトリバーを右クリックして、必要なコアンド(フロートまたはドッキング)を選択します。
- ・ (タイトルバーを) いちいして、)ウシンドンをドゥナングされて、は易所からドラッグしてフロートさせます。フロートして、い をうシンドンを(タイトルバーを) いちいして、)ドラッグしてドゥキングする場所にドラッグします。2つのセットの青い 矢印が表示されます。外側の4つの矢印のセットにより、アプケーションウシンドウ (GUIの上、右、下、および左 端)は目対してウシンドンをドゥキングすることができます。内側の矢印のセットは、カーソルが現在置かれて、ほうシン ドブン目対してウシンドンをドゥキングすることができます。ドラッグされたウシンドンを内側の矢印のセット (おは、ウィ ンドンのタイトルバーの上)の中心の下にトロップすることで、ドラッグされたウシンドンをトロップされたウンドン内のタブ されたウシンドンとしてドゥキングすることができます。

タブされたウンレドプをフロートするはは、そのタブをダブルクリックします。。タブされたウィンドプをタブされたグループからドラッグする はは、タブをドラッグします。

スキーマビュー内で C:\Documents and Settings\<username>\My Documents\Altova \XMLSpy2017\Examples\Tutorial フォルダーからXML スキーマファイルを移動させる練習をすること、上記の方法でインターフェイス内でのウィンドンの移動に買り親しんで作さい

1.3 メニューとツールバー

チュートアルのこのセクションでは XMLSpy のメニューとンールバーの機能を学習します。

メニュー

2 つのメニューバーがあます:() ドキュメントが開かれていない時に表示されるデフォルトメニュー、および(i) ドキュメントが開かれると表示されるXMLSpy アプリケーションメニュー全体。以下を行ってください:

- 1. メニューニアンド ファイル 閉じる」 を使用してすべての開かれてい るトキュメントを閉じます。 デフォレトのメニュー が表示されます。
- 2. AddressFirst.xsd ファイルを「ファイル」メニューの下にある最近開かれたファイルの「ストからクリック」で開き ます。ファイルがスキーマビューで開かれて、容易合、メニューはXMLSpy アプケーションメニューに変更されます。

メニューは主に機能には整理されており、メニュー内のエアンドは、カーソルポイントおけよアケイブトキュメントの現在のビューの選択の場合のみ実行することができます。メニューニアントが有効化おけは無効化されているかを決定する要素を理解することが口は以下を引ます:

1. ワキーマデザイン」メニューをフリックします。 図の保存」、 表示設定」、および ワーム」コマンドが無効化 されてい ることに注意して 代さい (下のスクリーンショント)。

ス=	⊧マ設計(<u>M</u>)	XSL/XQuery(Q)
	スキーマ設定(S)
6	図の保存(D)	
C	ドキュメンテー	-ションの生成(U)
	表示設定(C)	
	ズーム(Z)	

- 2. メニューの外側の空白をクリクして、メニューを消します。要素コポーネントの左横にある 図の表示 アイコン をクリクします。これによりスキーマビューのコンポーネントモデルビューは移動します(スキーマビューの 2番目のビュ ー;1番目のビューはスキーマ概要です)、スキーマデザインメニューを超認すると、図の保存」、表示設定」、 および「ズーム」コマントが有効化されていることが確認できます。これらのコマンドはスキーマビューのコンテンノモデ ルビュー内のみで有効化されますが、スキーマビューのスキーマ概要ませまその他のビューでは有効化されません。 XML スキーマファイルはスキマービューのみで開かれることに注意してください。
- XML スキーマファイルもまた、XML ファイルですから、このファイルもましてキストビューおよびグリッドビューでXML ファ イルとして表示されます。 XML ファイルですから、このファイルもましたできるメニューニアンドすべてしまこれらのビューで有効 化されます。 スキーマビューとテキストビュー内の 編集」メニューのニアンドを、有効化ませは無効化されているかを) 比較してなさい。
- 4. 次に、テキストビューとグリッドビューの XML | 挿入」 メニュー内のコマントを有効化ませま無効化されているか を比較して作さい コのメニュー内のコマンドはグリッドビューのみで有効化されています。

全てのメニューコマンドの詳細に関しては、ユーザーマニュアルの「ユーザーレファレンス」のセクショク参照してください

ツールバー

ッシールバーの表示は現在のビューには異ないます。アプケーションのデフォルトの設定は各ビューの正し、シールバーを表示 します、ツールバーは各ビューによって異ないます。ですが、カスタマイズダイアログ内のシールバー(『シール』カスタマイズ』 ッシールバー」、下のスクリーンショントにはリカスタム化することができます。

カスタマイズ	×
コマンド ツールバー ツール キーボード メニュー	マクロ プラグイン オプション
ツールバー(T):	
Attr. & Elem.	 リセット(R)
Authentic	
▼CALS/HTML テーブル	=
CDATA	新規作成(N)
DOCTYPE	
DTD/スキーマ	名前の変更(M)
JSON	削除(D)
Oracle XML DB	
Portable XML Form (PXF)	■ テキフト ラベルを表示(S)
Soap デバッガー	
WSDL 2.0 デザイン	-
🔲 全てのビューに変更を適用	

ットルバーをGUIの周辺で移動させる練習をしてくたさい。シールバーのハンドルをクリックして、GUI内の希望する場所に ドラッグします。(シールバーハンドルは各ツールバーの左横にある垂直の点線で表示されています。下のスクリーンショント 参照)。

| 🎬 🗟 🛱 🐮 📧 🖡 🖬 🖬 🖌 🖇 🚽

以下の場所にソールバーをドラッグします:() ツールバーエリア内の他のライン())他のソールバーの左側ませば右側、 (iii)メインウィンドンの中央、(v)アプリーションウィンドンの左側ませば右側(の操作には、グラップする)ンドルがアプリ ケーションウィンドンの左側ませば右側の上に置かれる必要があります。

終了するとAddressFirst.xsd ファイルを閉します。

1.4 テキストビューの設定

このセクションでは、再フォーマットされたトキュタントの設定および編集中のブックマークの使用方法を学習します。トキュタントが再フォーマットされると、テキストビューで表示され、XML 階層の各下のレベリは前のレベリルさ比較してインデントされて 表示されます(下のスクリーンショント参照)。ブックマークにより、トキュメントの位置をマークし、戻る場所を素早く検索する ことができます。

1	xml version="1.0" encoding="UTF-8"?
2	<company></company>
3	<address xsi:type="US-Address"></address>
4	<name>US dependency</name>
5	<street>Noble Ave.</street>
6	<city>Dallas</city>
7	<zip>04812</zip>
8	<state>Texas</state>
9	
10	<pre></pre>
11	<pre><first>Fred</first></pre>
12	<last>Smith</last>
13	<phoneext>22</phoneext>
14	<email>Smith@work.com</email>
15	
16	
	1

再フォーマット

再フォーマットコは 2 つのステップがあります: () 再フォーマット をオンにして、 インデントの数 慈設定します、 そして (i) 再 フォーマット を適用します。

- 1. C:\Documents and Settings\<username>\My Documents\Altova\XMLSpy2017 \Examples\Tutorial フォレダー内にあるファイルCompanyFirst.xml を開きます。(XML ドキュメント を表示するデフォレトの設定がテキストビューでは、場合、テキストビューは加替えます)
- オブションダイアログ 「シール」オブション ビュー」下のスクリーンション ビュータブで 「インデントを使用」チェックボックスをチェックします。これによりインデントのある再フォーマットが デフォルトの設定として)オスにはます。終了すると OK」をクリックします。この設定はデキストビューで開かれるすべてのファイルに適用されます。

拡張グリッド ビュー 「 属性のプレビューを表示 一 最適幅を自動的に提供	プログラム ロゴ ウィンドウ タイトル 図 起動時に表示 回 印刷時に表示 回 印刷時に表示 回 つル パスの名前
最適幅の制限 ピクセル セルの高さ制限 行	Authentic ビュー ▼ SPS ファイルに関連付けられている場合、常にファイ ルを Authentic ビューで開く
- 整形 ▼ インデントを使用 ニナコト パー、 (ニス教工) ギャンボ 押工 たわえね、 (ゆっパー、 わ	ブラウザー ビュー
テキストビューにし登形がみクルサードされるが、他のビューから切り替えや保存が行われたときに整形が適用されます。	→スキーマ ビュー 基底型を変更することで、影響を受けたコンテンツが派生し た型には含まれる事があります。
	 新たな基底型でも使用することができる場合、コンテン ツを保持する 基底型の修正時ごとに確認する

3. テキストビュー設定ダイアログを **表示 | テキストビューの設定 」**コマントを使用して (下のスクリーンショント) 開きます。 タブペインでタブサイズを3に減らします。 タブの挿入の率ボタのデフォレトの選択をそのままこしてお、 にく ださい これによ、再フォーマットのインデントが(スペースでしばなく) タブンなるようご設定されます、 各タブは 320空 白の幅しています。 終了すると **じ** K 」をグリングします。

- 4. メニューニアンド **編集 | 再フォーマット」**をクリックします。 トキュメントは新しい タブ値を持つように再フォーマットされます。
- 5. テキストビュー設定 ダイアログ (表示 | テキストビューの設定」) を開いて、視覚補助ペイン内の行末マーカーを 解除します。
- 6. テキストビュー内で、行末に移動して、行末マーカー、胡除することにより、次のラインが上に移動します。
- 7. グリッドビュー は川 潜えて、再度 テキストビューは川 潜えます。 ドキュメントは、再フォーマットされ、削除された行 末マーカーが回復されます。
- 天: オプションダイアログのビュータブ内で(「シール」オプション」 ビュー、上のスクリーンションり、インデントの使用のチェングボックスのチェンクを解除すると、すべての行はインデント無しこ再フォーマットされます。

ブックマーク

マークする行の左側のブックマークマージンにブックマークを表示することができます。トキュメントの行をブックマークして素早い参照やアクセスを行うことができます。

1. テキストビューの設定 (表示 | テキストビューの設定 、のスクリーンショント) で、ブリアージのオプショがマージパインで選択されてい ることを確認します。完了すると OK」をクリックします。

ଃ テキスト ビュー	の設定		×
余白 ✓ 行番号の余白 ✓ ブックマークの余 ● 折りたたみ余白	タブ (L) タブサイズ:3 白(B) ・ タブを挿入(T) (F) ・ スペースを挿入(S)	視覚補助 ✓ インデント ガイド(I) □ 行末マーカー(E) □ 空白マーカー(W)	OK キャンセル 適用(A)
─▼自動ハイライトの	D有効化		
◎ 選択された 単語(2	二一致する		
◎次の選択に一致す	する: 2 🔹 文字		
☑ 大文字と小文号	字を区別		
キーマップ			
Up	1行上へ		
Down	1行下へ		=
Left	1列左へ		
Right	1列右へ		
Ctrl + Up	1行スクロール		
Ctrl + Down	1行下へスクロール		
Ctrl + Left	1単語左へ		
Ctrl + Right	1単語右へ		
Alt+Up	×ML 要素を上に移動		v

- 2. ファイル Company First.xml のテキストビューで、ブリアイノオるライノムファーソルをポイントし、メニューコアンド 編集 | ブリアマークの挿入削除」を選択します。ライノはブリアイクされ、ブリアマーケアージン内に青い ブリアア ーケア 表示されます (下のスクリーンションナ参照)。
- 3. ステップ22同様に、他のラインのブックマークを作成します。

1	<pre><?xml version="1.0" encoding="UTF-8"?></pre>
2	<company></company>
3	<address xsi:type="US-Address"></address>
4	<name>US dependency</name>
5 😁	<pre><street>Noble Ave.</street></pre>
6 -	<city>Dallas</city>
7	<zip>04812</zip>
8	<state>Texas</state>
9	
10 🗎	<person degree="BA" manager="true" programmer="false"></person>
11 ~	<pre><first>Fred</first></pre>
12	<last>Smith</last>
13	<phoneext>22</phoneext>
14	<email>Smith@work.com</email>
15	
16	
17	

- 4. **F2」**(おはエマンド **編集 | 次のブックマークへ移動」**) を押して、ドキュメント内の次のブックマークへ移動し ます。 Shift+F2」(おはエマンド **編集 | 次のブックマークへ移動」**) を押して、ドキュメント内の前のブック マークに移動します。この2つのエマント・特別限なく繰り返すことができます。
- 5. ブリアークされたライノトコーソルをポイントして、メニューコマンド **編集 | ブックマークの挿入 削除」**を選択する と ブックマークガ削除されます。
- 6. 保存してファイルを閉します。ブックマークの精報はファイルに保存されないか、ファイルを再度開き確認して代さい。

2 XML スキーマ :基本

XML スキーマロは XML ドキュメントの構造が記述されます。 XML ドキュメントがスキーマ内で指定された要件に合致する か容検証することに XML スキーマは使用されます。 合致する場合、 その XML ドキュメントは**授当**てあると言われ、 そうで は、場合は不正であると言われます。 XML スキーマを使って XML ドキュメントで許される構造とエンテングをデザインするこ ができ、 XML ドキュメントが妥当かどう検証するこができます。

XML スキーマドキュシントの構造と構文は複雑で、それ自体がXML ドキュシントになっています。XML スキーマそのものも XML スキーマの壮様に従った妥当なものではかいばなりません。XMLSpy では、スキーマビューで使えるドラゾアンドド ロップにより、妥当な XML スキーマを簡単に作成することができます。作成された XML スキーマドキュシントはテキストビュ ーやグリッドビューでも編集可能ですが、スキーマビューを使ったほうが作成や編集をより簡単に行うことができます。

目的

このセクションではスキーマビューを使ってのXMLスキーマの編集方法を学習します。特に、以下のことが引こし、で学習します:

- スキーマファイルを新たっ作成する
- スキーマの名前空間を定義する
- 基本的なコンテンソモデルを定義する
- コテキストメニュームラッグアンドドロップを使ってコテンソモデルに要素を追加する

基本的なスキーマが完成したら %MLSPY% によるXML スキーマのさらに高度な機能について次のセクションで学習します。その次には %MLSPY% のスキーマナビゲーションとトキュメンテーションに関するセケショか続きます。

このセクションで使用されるコマンド

このセクションではスキーマビューだけを使用します。以下のコマントが使われます:

ダイアゲラム表示(おはコンテンバモデ)ビュー表示)、 このアイコンはスキーマ概要にある 全てのグローバリレコンポーネントの左側に配置されています。 このアイコンをクリックすること で対応するグローバリレコンポーネントのコンテンバモデリを表示します。

2.1 新しい XML スキーマファイルの作成

新しいXML スキーマファイルの作成:

1. メニューオプショから ファイル 新規作成」を選択します。新規ドキュメント作成」ダイアログが開きます。

作成した	いドキュメントの種類な選択してください。	
Vml	Septet 23 Deployment Descriptor	
xa	XML Quent Language v1.0	
Ya Xa	XML Query Language v31	
xal	XML Query Language VI 0	
xal	XML Query Language v3.1	
xar	XML Query Language for DB2	
xqu	XQuery Update Facility v1.0	
xqu	XQuery Update Facility v3.0	
xquery	XML Query Language v1.0	
xquery	XML Query Language v3.1	
xsd	XBRL Taxonomy Schema	=
xsd	XML Schema VI.0	
xsd	XML Schema v1.1	
veia	XML Signature	Ψ.
Style	Vision スタイルシートを選択 OK キャンセル	,

- 2. ダイアログから xsd (XML Schema v1.0) エントリ (ウインドウオのトキュタントの詳細としたトはスクレーン ショットと異なる場合があります。) を選択し **ひ**K」で確定します。空のスキーマファイルがウインドウのスキーマビュ ーに表示されます。
- 3. スキー る時 ツールバーから **XSD 1.0** モートボタン (下のスクリーンショント参照) をクリックして、スキーマビュー がXSD 1.0 編集モード になるよう 設定します。ます。



4. ルート要素を入力するよう求められます。

element	ENTER_NAME_OF_ROOT_ELEMENT ann:Cor

5. ハイライトされたフィール・をクリックして、Companyと入力します。 Enter」キーを押下することで確定します。 Companyがのスキーマのルート要素にない、グローバリレ要素として作成されます。メインウントドンで表示されるビュ ー(下のスクリーンショット)はスキーマ概要は呼ばれます。メインウントドカの上部に全てのグローバリレンポーネントのリ ストを表示することでスキーマの概要を提供し、下部には選択されたグローバリレンポーネントの属性と同一性制約 が表示されます。(個々のグローバリンポーネントのエレテンプモデルの表示と編集は、グローバリレンポーネントの 左にあるダイアグラム表示アイエンかどううこができます。)

				×
element	Company	ann:Comme	ent describing	your root element
	1			
禹性 同一性制約				
				×
Name	Туре	Use	Default	Fixed

- 6. 要素の注釈フィールド(ann)にて Company 要素の説明を入力します。 ここでは Root element 記述します。
- 7. メニューオプション プァイル 保存」をクリックして、好きな名前(例えばAddressFirst.xsd)でXMLスキーマ を保存します。

2.2 名前空間の定義

XML 名前空間はXML スキーマとXML ドキュメントにとて重要な課題です。XML スキーマドキュメントはXML スキ ーマ名前空間を参照する必要があり、XML ドキュメントインスタンスのためにターゲット名前空間を定義することができま す。スキーマデザイナーとしては、基本的には、どのプレフィックスを使用して、これらの名前空間をどのように定義するか決定 する必要があります。

作成中のXML スキーマ内で XML ドキュメントインスタンスのためにターゲット名前空間を定義します。 (XML スキーマ ドキュメントを新規作成すると XML スキーマ名前空間に対する必要なレフォンスは自動的に XMLSpy により作成さ れます。)

ターゲット名前空間の作成方法:

1. **ワキーマ設計 | スキーマ設定 」**をメニューオプションから選択します。これによりスキーマ設定ダイアログが開かれます。

スキーマ設定	×
デフォルトの要素 form :	O Qualified O Unqualified
デフォルトの属性 form :	🔘 Qualified 💿 Unqualified
デフォルトの Block :	
デフォルトの Final :	
Version :	
xml:lang	ID :
○ ターゲット名前空間なし	
◉ ターゲット名前空間	http://my-company.com/namespace
	Nemeeree
Frenz	http://mv-company.com/namespace
xs	http://www.w3.org/2001/XMLSchema
I	
	0K 4+>>tzu

- 2. ターゲットの名前空間のラジオボタンをクリックして、http://my-company.com/namespace を入力します。XMLSpy では、ターゲットの名前空間として入力された名前空間が XML スキ ーマトキュメントのデフォルトの名前空間として作成され、ダイアログの下のペインの名前空間 のリストに表示されます。
- 3. **OK」**ボタンを押して、確認します。

注意点:

- XML スキーマの名前空間は、XMLSpy により自動的に作成され、xs:のプレフィックスが与えられます。
- XML ドキュメントインスタンスが作成されると、ドキュメントが妥当であるために、スキーマ内で 定義されたターゲット名前空間が必要になります。

2.3 コンテンツモデルの定義

スキーマ概要には、既に作成されたCompanyとう名前のプローバリ要素が表示されます。この要素は1つのAddress 要素と出現数に制限の無いPerson 要素(コテンソモデル)を持つことになります。クローバリレコパーネントの中の要素、複合型、そして要素グリープはコンテンソモデルを持つことができます。

	×
📲 element	Company ann:Root element

このセクションでは Company 要素のコンテンソモデルを作成します。

基本的なコンテンツモデルの作成

Company 要素のコンテンソモデルを作成するは:

1. スキーマ概要にて、要素のダイアグラム表示アイコン・・・・ をクリックします。今のところは空の company 要素のコン テンソモデルが表示されます(以下のスクリーンションを参照)。その地に、コンポーネント入力へリパーにある Company エントをダブルクリックしてコンテンソモデルを表示させる方法もあります。



2. コテンチデルはコボジターとコポーネントによ構成されます。コボジターは2つのコポーネントの関係を指定します。Company コテンケデルでは、子要素を加えなかに子コボジターを追加する必要があります。コン ボジターを追加するはは、Company 要素を右クリクレ、コテキストメニューから子を追加|シーケンス」を選択 します。(シーケンス、選択」、そして全て」は、コテンケデルで使用される3つのコボジターです)



この操作によりシーケンスが指定され、そこから表示されるコポーネントを定義するシーケンスコポジッターが挿入されます。



- 3. シーケンスコンポジッターを右クリックして、**子を追加 | 要素」**を選択します。名前の無い要素コンポーネントが追加されます。
- 4. 要素の名前にAddress と入力して Enter」で確定します。

Root element	Address

5. シーケンスコンボジターを再度右クリックして、 **子を追加 | 要素」**を選択します。新たご作成された要素コンポジットの名前をPerson にします。



これで1つの会社あさ2、1つの注所とし、の人物の記述を許可するスキーマを定義しました。これからきに Person要素の数を増やす必要があます。

6. Person 要素を右クリクレ コテキストメニューかど無制限を選択します。ダイアグラムのPerson 要素に許された出現回数として1かど無限が表示されます。



その他に、詳細入力へリパー内からminOccとmaxOccフィーリルを編集して、許可された出現回数を指定することができます。この場合ならそれそれ1とunboundedにないます。

コンテンツモデル構造に新たな階層を追加

これまでで作成した基本的なコンテンソモデルは、1つのレベル(Address とPerson 要素とらcompany 要素の子レベル)しか持ちませんでした。ここではAddress 要素の内容を定義します。Address 要素以下にName、Street、 とcity 要素を配置します。これは第200レベルです。再度子コンポジッターをAddress 要素に加え、要素コンポーネントを加えることにはます。

や方:

- 1. Address 要素を右クリクレコテキストメニューを開き 子を追加 | シーケンス」を選択します。これでノーケンス コノポジッターが追加されます。
- 2. シーケンスコンポジターを右クリックし、 子を追加 | 要素」を選択します。 新たご作成された要素コンポーネントの 名前をName に変更します。



複合型、単純型、スキーマデータ型

こまで、要素の型を明示的に定義はしませんでした。テキストタルクリンクス、スキーマをテキストビューに表示させてたさい(下記参照)。シーケンスコンポジッターが挿入された場所にはxs:complexType 内にxs:sequence 要素が挿入されているのが分かると思います。簡単に説明すると、Company そしてAddress 要素は子要素を持っているため複合型にないます。複合型要素は属性おけて要素を持ちます。

一方で、単純型要素は属性を持たずテキストだけを含みます。テキストは、文字列や日付、数値などです。Addressの子であるNameにはデキストしか入って欲しくありません。つま文字列だけ、制限され、単純型とりことによります。 XML スキーマデーク型をxs:stringにすることで、このよう設定を行うことができます。

Name 要素のデータ型を定義するは:

- 1. スキーマタブをクリックし、スキーマビューに戻ります。
- 2. Name 要素をクリックして選択します。
- 3. type 詳細入力へリパーこて、Name コルボボックスのトロップダウンメニューからxs:string を選択します。



minOcc とmaxOcc は1の植を持っていて、この要素は1度しか現れないという意して代える

Name 要素のテキスト表記は以下のようこなじます:

<xs:element name="Name" type="xs:string"/>

注意:単純型要素は幾つかあるXMLスキーマデータ型のとれても取ることができます。全ての場合において、要素ボックスにはテキストコンテンツであることを示すアイコンが表示されます。

2.4 ドラッグアンドドロップによる要素の追加

こまでは要素やコンボジターを右クリックし、コンテキストメニューを使って要素を追加してきました。ドラッグアンドドロップを 使うニコンテキストメニューより光素早、要素を追加することができます。このセクションでは、ドラッグアンドドロップを使って Address 要素にされて子要素を追加して定義を完成させます。

ドラグアンドトロップを使ってAddress 要素の定義を完成させるは:

1. Address 要素のName 要素をクリックして Ctrl」キーを押下したま要素ボックスをマウスでドラッグします。 要素ボックスに表示される小さな「+」アイコンはコピーが行われようしてい ほう。要素のコピーが接続線と一緒に表示され、要素がどこご作成されるのかを示しています。



2. マウズボタク離して、Address シーケンスに新し、要素を作成します。新し、要素が間違った場所に作成された場合、Name 要素以下にドラゾレてたさい。



- 3. 要素ボックスをダブルクリックして、要素の名前をStreet に変更します。
- 4. 同様に City と と名前でもうつの要素を作成します。 コテンソモデルは以下のようことはます。





2.5 コンテンツモデルビューの構成

そろそろコテンバモデルビューの設定を行っても良いでしょう。コテンバモデルビューを要素ボックスに、その要素が持つ型が表示されるようよ設定します。コテンバモデルビューの設定を行うには:

- 1. コポーネントのコテンチェデルビューを選択します(ダイアログ表示アイコン・1 をクリックします)。
- 2. メニューオプション 「スキーマ設計 | 表示設定」を選択します。スキーマ表示設定ダイアログが表示されます。

スキーマ表示構成			×
要素属性			ОК
		×	キャンセル
type	type	•]
fixed	fixed	-	1
length	length	<u> </u>	すべてクリア
min/maxexci			
pattern		•	
type		-	定義済み
I			開、/保存
単一行設定		一共通行設定一	
) 🙆 営け行を表示(1)	「一行の」詳明を	表示(1)
) ● 市に口を払力(に) ◎ 値の無い(示を照す(山)	■ 110/06-312	10/1(0)
(D) // // // (D)			
幅		距離	
最低(M): 🚽		親/子(P):]
最高(X):		子/子(C):	0
図に表示			描画方向
☑ 説明、幅の上限(A)	:		◎水平(Z)
☑ 代替グループチェイン	ν(U)		◎ 垂直(\)
■ 属性(B)			○王區(*)
▼アサーションの			スマート制約
□□□─性制約00			I 有効(E)

- 3. (Element タブニある)追加アイコン 目 をクリックして、要素ボックスはするプロ ティク表示レコードを追加します。
- 4. ドロップダウンメニューから type を選択します (またはフィールドをダブルクリックして "type" と入力します)。 これで 各要素のデータ型がエンテンパモデルビューて表示されるようにないます。
- 5. 記述行設定では、 値の無い行を隠す」を選択します。これで要素の値が無い場合、データ型の記述子が要素 ボックスに表示されなないます (例えば、要素が複合型の場合などです)。



xs:string とうご述子がName、Street、そしてCity 要素に表示されているのに対して、複合型の要素には無いとは注目してください。これは値の無い行を隠すオブションが有効になっているためです。

- 6. 記述行設定グレープで、常に行を表示ランオボタンを選択します。
- 7. じK」をクリックして変更を確定します。



データ型の記述子が常に(植を持たな)、複合型の要素ボックスでも)表示されていることご注目してくたさい。

ڮ:

- プロパテの表示レコードは編集することができ そこで入力された値が要素定義の一部にないます。
- スキーマ表示設定ダイアログで行った設定はスキーマドキュメテーション出力や印刷時の出力にも適用されます。

基本的なスキーマの完成 2.6

こまでAddress 要素の定義を引ました。次にPerson 要素の内容を定義する必要があます。Person 要素は 以下の子要素(全て単純型)を含む様にします First、Last、Title、PhoneExtt、 そして Email。 Title なら Ch い要素は全て必須で、設定された順に現れなければなりません。PhoneExt のデータ型は2桁までのxs:integer とな り、それ以外は全てxs:string といます。

Person のコンテンソモデルを作成するは:

- 1. Person 要素を右クルクレ、コンテキストメニューを表示します。 子を追加 | シーケンス」を選択します。 これに よシーケンスコンポジッターが挿入されます。
- 2. シーケンスコボジターを右クリクレ、 **子を追加 | 要素 」** を選択します。 3. First を要素名として入力し、 **タブ」キーを押します。これでカーソルが**type フィールドご移動します。



- 4. xs:string エノトをドロップダウノストから選択するか、その値をtype フィールドに入力します。
- 5. ドラゾアンドドロップにより、その他4つの要素を作成します。それぞれLast、Title、PhoneExt そして Email 名前を引ます。



モ: Ctrl」キーを押しながら必要な要素をケルクすることで複数の要素を選択することができます。この方法で、例えば 複数の要素を一度にことーすることができます。

要素を省略可能にする

Title 要素を右クリックして、コンテキストメニューかど省略可能を選択します。要素ボックスの境界線が実線から、要素が省略可能であることを示す点線に変わります。

	[≡] First	言羊糸田		τ×
	type xs:string	name	Title	•
	≣ Laat	isRef		
		minOcc	0	
	type xsistring	maxOcc	1	•
	Title	type	xsistring	•
╢╼╍╌╘╴┤		content	simple	▼
		derivedBy	/	▼
		default		▼
		fixed		▼
	type xsanteger	nillable		•
	Email	block		_
	tupo veretring	form		
	Type [xs.string]	lid		
		単間部	純型	

詳細入力へリパーにて、minOcc が Q maxOcc が 1 と表示され、要素が省略可能であることを示します。 コンテキスト メニューを使う方法以外にも、minOcc に 0をセットして要素を省略可能にすることができます。

要素の内容を制限する

PhoneExt 要素がxs:integer 型で 最高 2桁の数値だけを持てるようこするこむ:

1. PhoneExt 要素のType フィールドをダブルクリックして、ドロップダナンストからxs:integer を選択 (ませは入力)します。

Title	min ma:
	typi cor
type xs:integer 💌	der def
xs:int Emaxs:integer	
type xs:language	

この時点でファセット入力へリパーのアイテムが変化します。

2. ファセット入力へリパーにて、maxIncl フィールトをダブレクリックして、99と入力し、Enter」で確定します。

ĒFirst	言羊糸田	ά×
type xsistring	name PhoneEx	t 🔳
ELast	minOcc 1	
	type xs:intege	r 💌
type xs:string	content simple derivedBy restriction	n 💌
[■] PhoneExt	default Lived	
type xs:integer	詳細 単純型	
Email	ファセット	τ×
Lype (xs.suring)	minIncl maxIncl 99	6
	minExcl	6

この定義により 99までの内線番号 (Phone Extension)が妥当といます。

3. メニューオプション プァイル 保存」 を選択し、スキーママンする変更を保存します。

天:

- 単純型のXMLスキーマデータ型(例えばxs:string やxs:date)を選択することで、詳細入力へリパーのコンテン ツモデリが自動的に単純型 (content = simple) ご変更されます。
- 要素にコオジッター(シーケンス 選択、ませますべて)を追加することで、詳細入力へルパーのコンテンソモデルが自動的 に単純型 (content = complex
- 上記で説明されたスキーマはXMLSpy アプリケーションフォレダーの C:\Documents and Settings \<username>\My Documents\Altova\XMLSpy2017\Examples\Tutorial フォレダーにある AddressFirst.xsd とと名前のファイル名で入手可能です。

3 XML スキーマ :高度な機能

今までの作業で基本的なスキーマが作成されました。これからスキーマの開発における高度な機能に進んでしきます。

目的

このセクションでは、以下のことを学習します:

- スキーマの要素としても使用できる複合型ならびに単純型の扱い方
- <u>グローバル要素</u>を作成し他の要素からの参照を行う
- <u>属性</u>との列挙値を含むプロピティーを作成する

このセクションでは、直前のパートで作成したAddressFirst.xsd スキーマからスタートします。

このセグションで使用されるコマンド このセグションではスキーマビューだけを使用します。以下のコマンドが使われます:

- ダイアゲラム表示(おはコンテンパモデ)ビュー表示)。 このアイコンはスキーマ概要にある 全てのグローバリレコポーネントの左側に配置されています。 このアイコンをクリンクすること で対応するグローバリレコポーネントのコンテンパモデルを表示します。
- 全てのグローバルを表示。このアイコンはコンテングモデルビューの左上に配置されて、ます。このアイコンをクリックすることで、全てのグローバルコンポーネントを表示するスキーマ概要に移動します。
- 追加アイコノはスキーマ概要の左上に配置されています。このアイコンをクリックすることでグ ローバリレコパーネントの追加を引います。

3.1 複合型と単純型を扱う

要素のコンテンバーデルの定義を行ったので、それをスキーマのどうか別の場所で再利用することにします。そのためはは要素 の定義をグローバリ複合型、またはグローバリ要素として作成する必要があります。このセクションではグローバリ複合型を 扱います。まずは複合型をグローバリなレベルで作成し、それをコンテンバーデル内で使用できるように拡張します。 グローバ ル要素については後ほど学習します。

グローバル複合型の作成

既に定義した(Name、Street、City要素を含む)基本的なAddress要素は注所の書式とて色々な場所で利用できます。そこで、これと要素の再利用をすることができ、複合型を定義することにます。

グローバル複合型を作成する」は:

- 1. コテンアデルビューCT Address 要素を右クリックする。
- 2. 現れたコンテキストメニューで グローバリ化 | 複合型」を選択します。AddressType とう名前のグローバリ複 合型が作成され、Company コンテングモデルのAddress 要素がの型に割り当てられます。Address 要素 の内容は、黄色のボッグスで表示されて、GAddressType のコンテンノモデルになります。Address 要素のデー タ型がAddressType になっていることに注目して作さい。



- 3. 全てのグローバルを表示アイコン目をクリックします。メインウンドウの表示がスキーマ内の全てのグローバルコンポーネントを表示するスキーマ概要に移動します。
- 4. コポーネント入力へリパーにて(+ て示される)要素と複合型エント」の展開アイコンをクリックして、対応するスキ ーマが表示されることを確認します。

スキーマ概要は2つのガローバリンポーネント(Company 要素と複合型のAddressType)を表示しています。 コパーネント入力へリバーにもAddressType 複合型は表示されます。

		×	コンポーネント	ąΧ
📲 element	Company	ann:	⊟element (1)	
complexType	AddressType	ann:	Company	
			© xs:anyType	
			⊞simpleType (45)	

5. AddressType のコンテンバテアバコン・・・ をアレックして、(下のスクレーンショントにあるような)コンテンバテアノに表示します。複合型コンテナーの形に注目して付き。



6. 全てのガローバルを表示アイコン国をクリックして、スキーマ概要に戻います。

複合型の拡張

それでは 2つの国は特化した注所を表す為に、グローバルAddressType コパーネントを使用したいと思います。このために、AddressType をベースこした新たな複合型を作成し、その定義を拡張します。

以下を引ます:

- 1. スキーマ概要に移動します(コンテンソモデ)ビューには、は場合、全てのグローバリル表示アイコン・目をクリックします)。
- 2. コパーネントウィンドカケェトにある追加アイコン目をクリックします。以下のメニューが開きます:

	×
インポートΦ インクルード(L) 再定義(B)	iompany ann: iddressType ann:
 要素(E) グループ(<u>G</u>)	
単純型(S)	
複合型(<u>C</u>)	

- 3. メニューから複合型を選択します。コンポーネントレストに新しいレコートが現れ、コンポーネントの名前を入力するようにカーソルがセットされます。
- 4. US-Address と入力し、Enter」で確定します。(ハイフン()を入力する代わりに空白(スペース)を入力した場合、要素名が赤々なり不正な文字であることか示されます)

		×	コンポーネント	ņ	×
element	Company	ann:Root Element	⊟element (1)		
📢 complexType	AddressType	ann:	Company		
complexType	US-Address	ann:	⊜complexType (3)		
			Co AddressType		
			OS-Address		
			🛛 🚾 xs:anyType		
			⊞simpleType (45)		

- 5. US-Address のコンテンソモデルビューアイコン 2 をクリックし、新し、複合型のコンテンソモデルを表示します。コンテンソモデルは、以下のスクリーンショントはないことです。
- 6. 詳細入力へいいーて、base コンボボックスをクリックし、AddressType エント 塩選択します。



コテンモデルビューはUS-AddressのコテンモデルとてAddressType が表示されます。(下のスク リーンショント)。

	コンポーネント ユ ×
	Eelement (1) Ecompany ComplexType (3) C AddressType US-Address C xs:anyType SimpleType (45)
Address Type (extension)	グローバル 名前空間 同一性
City	Imame US-Address base AddressType derivedBy extension mixed ▼ abstract ▼ block ▼ final ▼

7. これでUS-Address 複合型のコンテンソモデルを郵便番号 (ZIP Code)を持つよう拡張することができます。 US-Address コパーネントを右クリゾルズ、現れたコテキストメニューから **子を追加 | シーケンス**」を選択しま す。シーケンスコルドベッターがAddressType ボックスの外に表示されます(下のスクリーンショントを参照)。これ は要素につける拡張であることを表わしています。



- 8. 新し、シーケンスコポジッターを右クリックして、 子を追加 | 要素」を選択します。
- 9. 新たったのされた要素にZip とり名前を付け、「タブ」キーを押下し、カーソルをtype フィールドで移動させます。
- 10. ドロップダウンメニューからxs:positiveInteger を選択し Enter」キーによ確定します。



これで、複合型 AddressType をもとした郵便番号 CIP Code の拡張を持つ複合型 US-Address が作成されました。

グローバル単純型

複合型のus-Address が複合型のAddressType をベースこしたように、単純型をベースにすることもできます。その 利点はクローバリ複合型の場合と同じで、単純型が再利用されることにあります。単純型を再利用するためには、その単純 型をグローバリルご定義しなければなりません。このチュートレアルではアメリカの州のコンテンソモデルを単純型として定義するこ とこします。この単純型は他の要素のベースとして使用されることになります。

グローバル単純型の作成

クローバル単純型の作成は、クローバルコンポーネントのレストに新し、単純型を追加し、名前を与え、データ型を定義する ことから構成されています。

グローバル単純型を作成するこさ以下をおここと、ます:

1. スキーマ概要は別、替えます。(コンテンソモデリビュー内で作業している場合は、すべてのガローバリアイコンの表

示団 をりりれます。)

- 2. アイコンの追加をクリックします。表示されるコンテキストメニューからSimpleTypeを選択します。
- 3. 新し作成された simple Type の名前 とて US-State を入力します。
- 4. Enter」を押して確認します。単純型 US-State が作成され、コポーネント入力へリパー内の単純型のリストに表示されます。 \$imple Type エントリのアイコの展開をクリックして確認してたさい。

		×	コンポーネント ユ ×
📲 element	Company	ann:Root Element	📮 element (1)
📲 complexType	AddressType	ann:	Company
📲 complexType	US-Address	ann:	□ □ complexType (3)
simpleType	US-State	ann:	- Coll Address I ype
			- O US-Address
			🔲 🗤 💽 xs:anyType
			⊟simpleType (46)
			State
			s:anySimpleType
			-s xs:anyURI
			💽 xs:base64Binary 🖉
			グローバル 名前空間 同一性制約

5. 詳細入力へいや (下のスクリーンション)内では restr の値フィールドローンリンをポイントして、 xs:string を入力、ませはrestr 値フィールドのトロップダウンメニューからxs:string を選択します。



xs:string のデータ型である US-State とり名前の単純型が作成されます。このグローバルコポーネントは US-Address のコンテンソモデル内で使用することができます。

コンテンソモデル内でグローバル単純型を使用する

グローバル単純型はコテンソモデル内でコパーネントの型を定義するために使用することができます。State とら名前の要素を定義するためこのUS-Address コテンソモデル内のUS-State を使用します。

次を引ます:

- 1. スキーマ概要内のUS-Address のコポーネントモデルビューアイコン・ セラレクします。
- 2. 下のシーケンスコンポンターを右クリックして、 子の追加 | 要素」を選択します。
- 3. 要素名のために State を入力します。
- 4. ダブ」キーを押して、カーソルを型詳細ラインの値フィールドコポイントします。
- 5. Jutity JACONTO JUS-State BERLET.


State 要素は US-State 単純型をベースことす。

第2の複合型をAddressTypeをベースにして作成する。

UKの注所が入力できるようにグローバル複合型を作成します。 複合型は AddressType をベースこして、UKアドレスのフォーマットに一致するようご拡張されています。

次を引ます:

- 1. スキーマ概要内にAddressType (base=AddressType) をベースにする UK-Address と 治的のつ ーバル 複合型を作成します。
- 2. UK-Addressのコテンルデルビー内に Postcode 要素を作成し xs:string の型を与えます。

K-Address コノテンソモデルは以下のようことはす:



天: このセグションでは、コンテンノモデル定義で使用されるガローバル単純型および複合型を作成しました。グローバル型の利点は複数の定義で再利用できることです。

3.2 グローバル要素の参照

このセクションでは既に定義した Person 要素をグローバリ要素に変換し、その要素をCompany 要素が参照します。

- 1. 全てのガローバルを表示アイコン国をクリックして、スキーマ概要に移動します。
- 2. Company 要素のダイアグラム表示アイコン をクリックします。
- 3. Person 要素を右クルクレ、ゲローバリ化 | 要素」を選択します。要素に小さい矢印アイエル表示され、この要素がプローバルに宣言された Person 要素を参照していることを表します。詳細入カヘルパーには isRef チェングボックスにチェングが入ります。



4. 全てのガローバルを表示アイコン国 をケリックレてスキーマ概要に戻ります。Person 要素がガローバル要素として表示されます。コポーネント入力へリバーにも Person が表示されます。

		×	
📲 element	Company	ann:Root element	⊟element (2)
📲 complexType	AddressType	ann:	- E Company
📲 complexType	US-Address	ann:	Person
simpleType	US-State	ann:	⊟complexType (4)
• complexType	UK-Address	ann:	AddressType
element	Person	ann:	🛛 🕞 UK-Address
			G US-Address
			xs:anyType
			⊞simpleType (46)
			グローバル 名前空間 同一性制約

5. コポーネント入力ヘルパーこて、Person 要素をダブルクリックレーバル要素 Person のコンテンソモデ ルを開きます。



グローバリ要素ボックスは、た印か無してとは注目してくたさい。これはその要素が参照されているものであい、参照を行っている要素では無いからです。参照を行っている要素のボックスにた印は表示されます。

注意:

- グローバリ要素を参照する要素は、参照するグローバリ要素と同じ名前でなければなりません。
- グローバリ定義はXML ドキュメントのとこでユポーネントが使用されるかび記述はおうわず、ユテンツモデルの記述だけ をうします。グローバリ定義が他のユポーネントから参照された時に、XML ドキュメント内での場所が指定されます。

グローバリル電話された要素は異なる場所で再利用することが可能です。グローバリ複合型の場合とは違い、グローバリ要素そのものを変更することなくコテッソモデルの変更を行うことはできません。グローバリ要素を参照する要素のコテンソモデルを変更した場合、グローバリ要素のコテッソモデルも同様に変更され、そのグローバリ要素を 参照しているその他の要素のコテッソモデルを変更されます。

属性と列挙型 3.3

このセクションでは、属性と別挙型の属性の作成方法について学習します

要素属性の定義

- 1. スキーマ概要にて、Person 要素をクリックしてアケティブにします。
- 2. 属性 /同一性制約タブブループ(スキーマ概要ウィンドウ下部)の左上にある追加アイコン 目 をクリックし属性エン トを選択します。

		×
📲 element	Company	ann:Root element
complexType	AddressType	ann:
complexType	US-Address	ann:
simpleType	US-State	ann:
complexType	UK-Address	ann:
element	Person	ann:
属性同一性制約		
		×
属性(<u>A</u>)	Use [Default Fixed
属性グループ(<u>G</u>)		
任意属性(Y)		

- 3. Name フィールドこManager とう属性の名前を入力します。
- Type ゴボボリスでxs:boolean を選択します。
 ゴボボリスでrequired を選択します。

							\sim	
■ 1 1 <u></u> 1	Company				ann:Root element		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
📲 complexType	AddressType			ann:	ann:			
📲 complexType	US-Address				ann:	ann:		
simpleType	US-State				ann:			
📲 complexType	UK-Address				ann:			
📲 element	Person				ann:			
属性同一性制約								
							X	
Name	Туре		Use		Default	Fixed		
Manager xs:boolean 💌		•	required	•				
Programmer xs:boolean 💌		•	optional	•				
Degree	xsistring	•		•	-		_	

6. 同様の方法でProgrammer と 溶前の Type=xs:boolean で Use=optional の属性を作成しま す。

属性の列挙型の定義

列挙型とはある属性で許される値の集合のことです。XML インスタンスドキュメント内の属性の値がXML スキーマで指定された列挙型の中にては、場合、ドキュメントは不正とないます。ここでは Person 要素の Degree 属性に対して列挙型 を作成します。

次を引ます:

- 1. スキーマ概要にて、Person 要素をクリックしてアケティブにします。
- 2. 属性ウィンドカ左上にある追加アイコン目をフリックして、属性」エンドを選択します。
- 3. 属性の名前に Degree と入力し Type にxs:string を選択します。
- 4. Degree 属性を選択した状態で、入力へリパー内の列挙」タブをクリックします(スクリーンションを参照)。

属性 同一性制約						ファセット ユ x
				×	1 6	
Name Ty	ype	Use	Default	Fixed		
Manager xs	s:boolean 📃 💌	required 💌				BA
Programmer xs	s:boolean 📃 💌	optional 💌				MA
Degree xs	sistring 📃 💌	-	•			Ph.D
					11	
					-	ファセット パターン 列挙 サンプル

- 5. 列挙」タズにて追加アイコン 思をかりかします。
- 6. BA と入力して Enter」で確定します。
- 7. 同様にMA と PhD と う 学を追加します。
- 8. Person のコンテンソモデルビューアイコン 4 をクリックします。



先程定義した属性がコンテンソモデルビューに表示されます。展開アイコンでその要素で定義された全ての属性が表示されます。この表示モードはメニューオプションの「スキーで設計 | 表示設定」の図に表示」ペインにある 属性」チェックボックスのチェックを入れることで有効になります。

9. 全てのガローバルを表示アイコン国をクリックして、スキーマ概要に戻います。

完成したスキーマの保存

注意 XMLSpy による スキーマファイルを保存する前に 既にある AddressLast.xsd ファイルの名前を変更し例えば AddressLast_original.xsd) 上書きしな ようことす。

完成したスキーマファイルを引きる名前で保存します(ファイル)名前を付けて保存」)。チュートアの次のパートで使用されるスキーマファイルの名前はAddressLast.xsdであるため、ここでもAddressLast.xsd と名前をつけることを指わります。

4 XML スキーマ :XMLSpy の機能

XMLスキーマを完成させた後は、ナビゲーションヨートカットとXMLSpy から生成することができるスキーマトキュメンテーションの学習をしましょう

このセクションで使用されるコマンド

このセクションではスキーマビューだけを使用します。以下のコマントが使われます:

ダイアグラム表示(おおはコンテンソモデルビュー表示)、コのアイコンはスキーマ概要にある全てのグローバルコンポーネントの左側に配置されています。コのアイコンをクリックすることで対応するグローバルコンポーネントのコンテンソモデルを表示します。

4.1 スキーマナビゲーション

このセクションでは効果的にスキーマビューをナビゲートする方法を学習します。

グローバルコンポーネントのコンテンソモデルを表示する

コテンソモデルを持つことのできるグローバルコポーネントは複合型、要素、そして要素グループです。これらのコポーネントのコケテンパモデルビューは以下のようこして開くことができます。

- スキーマ概要にて、コポーネント名の左にあるダイアグラム表示アイコン・ をりいりします。
- スキーマ概要かコテンソモデルビューにてコポーネント入力ヘルパーの複合型、要素、おは要素グループをダブル クリックします(以下のスクリーンショントを参照)。そのコンポーネントのコンテンソモデルが表示されます。



コンポーネント入力へルパーこあるその他のグローバルコンポーネント(単純型、属性、属性グループ)をダブルクリック すると、そのコンポーネントはスキーマ概要でリイライトされます(そのようにコンポーネントはコンテンソモデルを持たない ため)。

コポーネント入力へルパーにある、グローバルや名前空間タブでもダブルクリックによるメサエズムが使用できます。

参照している要素からグローバリ要素に移動する

グローバリ要素を参照する要素がエンテンパモデルに含まれている場合、「Ctrl」キーを押したから必要な要素をダブルクトックすることで、そのグローバリ要素おはその要素のエンポーネントに移動することができます。

例えば Company コテンモデル参照してい 国際、 Ctrl」キーを押下したがら、Last をダブルクリンクすることで Person コテンソモデル開かれ その中にあるLast 要素が、イライトされます。



Last 要素がバイライトされたとき、関連する情報が入力へいいーならいに情報ウィンドウに表示されます。

複合型の定義に移動する

複合型はコケンモデル内の要素の型とてよく使用されます。コケンノモデル内が直接複合型の定義に移動した場合は、黄色のボックスにある複合型の名前をダブルクリックします(以下のスクリーンショントのマウスポインターに注目)。



この操作で複合型のコンテンソモデルビューに移動することができます。



★: 参照されているガローバリ要素の場合、参照を行っているエレテンプモデル内の要素を Ctrl」キーを押下しながら ダブルクリックすることで、対応する複合型定義内の要素に移動することができます。

5 スキーマドキュメンテーション

XMLSpy を使ってXML スキーマの詳細な文章をHTML やMicrosoft Word (MS Word)の形式で生成すること ができます。トキュメントは収める ロンポーネントと希望する詳細度を選択します。 関連するコンポーネントはHTML や MS Word トキュメントなどで \イライトされます。 MS Word トキュメントを生成するためはよ お使い のコンピューター (ま たはネットワーク)に MS Word がインストールされてい ら必要があります。

このセクションでは AddressLast.xsd XML スキーマのドキュタンテーションを生成します。

以下を引ます:

1. メニューオプショの ワキーマ設計 | トキュメンテーションの生成」 を選択し スキーマトキュメントダイフログを表示します。

 HTML、Word、または RTF フォーマットにおける スキーマト HTML、Word、RTF、または PDF フォーマットに対してユー SPS デザインを選択: %AltovaSchemaDoc%¥OverallDoc 	キュメントに対して固定デザインを使用 ザー定義デザインを使用(StyleVision が必 umentation.sps	要)] 参照 編集
出力書式 HTML 図の作成: Microsoft Word ● PNG ● EMF RTF 回図を埋込む PDF(上記参照) 回図のフォルダーを作成 二 出力を複数のファイルに分割 HTML (こ CSS を埋め込む	 ✓ 生成後に結果ファイルを表示 ✓ スキーマドキュメンテーション設定 ローカルファイルにリンクを生成 ● 絶対パス ○ 結果ファイルに対する相対パス 	き保存
含める対象	詳細	
 ✓ インデックス ✓ 単純型 ✓ グローバル属性 ✓ 視合型 ✓ グローバル要素 ✓ グリレープ ✓ ローカル属性 ✓ 代替グループ ✓ ローカル要素 ✓ 再定義 ✓ NOTATION ✓ 参照されたスキーマ 	 ✓ 図 ✓ 単 ✓ 名前空間 ✓ パ ✓ 型 ✓ 列 ✓ イ替グループ ✓ 厚 ✓ 子 ✓ 使用者 ✓ プロパティ ✓ ソ すべて選択 	〔一のファセット ターン]挙]叶性]一性制約 :釈 ースコード _ックをすべて解除

- 2. 出力書式にHTMLを選択し、OK」をクリクします。
- 3. 名前を付けて保存」ダイブログにて、ファイルが保存される場所を選択し、ファイルの名前を入力(例えば AddressLast.html)します。 保存」ボタンを押してファイルを保存します。

XMLSpy のプラナザービューにHTML ドキュメントが表示されます。表示されてい ビレンクをプリックして、対応する コンポーネントに移動します。

schema localion: C:UsersialUDocuments\Altova\XMLSpy2013 LizamplesTutorialNaddressLast.ssd attributeFormDefault: unqualified targetNamespace: http://my-company.com/namespace ElementFormDefault: gualified targetNamespace: http://my-company.com/namespace Element Company diagram diagram diagram diagram diagram diagram diagram tormpany_com/namespace properties content complex namespace http://my-company.com/namespace properties content complex children Address Person annotation Root element source <ss: element="" name="Company"> <ss: annotation<br="">Root element source <ss: element="" name="Company"> <ss: complextype=""> <ss: complextype=""> <ss: complextype=""> <ss: element="" name="Address" type="AddressType"></ss:> <ss: element="" maxoccurs="unbounded" ref="Person"></ss:> </ss:> </ss:> </ss:> </ss:> </ss:> </ss:> 	Schema AddressLast.xsd						
schema localion: <u>C:UsersialUDocuments\Altova\XMLSpy2013</u> <u>UzamplesTutorialAddressLast.ssd</u> attributeFornDefault: unqualified targetNamespace: http://my-company.com/namespace Elements Complex types Simple types <u>Company AddressType US-State</u> Person: <u>UK-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u> <u>US-Address</u>							
Usamples/Lutorial/AddressLast.xsd attributeFormDefault qualified targetNamespace: http://my-company.com/namespace Element Compaxy AddressType US-State Person UK-Address US-Address US-Address element Company Image: Company Address State Person UK-Address US-Address US-Address diagram Image: Company term diagram Image: Company term namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element source <xs:element name="Company"> <xs:ionotation> <xs:ionotation> <xs:ionotation> <xs:ionotation> <xs:ionotation> <xs:sequence> <xs:ielement maxoccurs="unbounded" ref="Person"></xs:ielement> <xs:ielement maxoccurs="unbounded" ref="Person"></xs:ielement> <xs:ielement maxoccurs="unbounded" ref="Person"></xs:ielement></xs:sequence></xs:ionotation></xs:ionotation></xs:ionotation></xs:ionotation></xs:ionotation></xs:element>	schema locatio	schema location: C:\Users\alU\Documents\Altova\XMLSpy2013					
attributeFormDefault: unqualified elementFormDefault: unqualified targetNamespace: http://my-company.com/namespace Elements Complex types Simple types Company AddressType US-State Person UK-Address US-Address		\Examples\Tutorial\AddressLast.xsd					
elementFormDefault qualified targetNamespace: http://my-company.com/namespace Elements Complex types Simple types Company AddressType US-State Person UK-Address US-Address element Company diagram diagram type_ivpe_ivpe_ivpe_ivpe_ivpe_ivpe_ivpe_iv	attributeFormE	attributeFormDefault: unqualified					
Elements Complex types Simple types Company AddressType US-State Person UK-Address US-A	elementi-ormipetauti: qualified						
Elements Complex types Simple types Company AddressType US-State Person UK-Address US-Address US-Address diagram diagram diagram diagram company diagram d	argenvarmespace: nttp://my-company.com/namespace						
Company AddressType US-State Person UK-Address US-Address US-Address US-Address element Company diagram Company difference fype AddressType Root element namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element source <xs:element name="Company"> <xs:selement name="Company"> <xs:selement name="Company"> <xs:selement <="" pre=""></xs:selement></xs:selement></xs:selement></xs:element>	Elements Cor	nplex types Simple types					
Person UK-Address US-Address element Company diagram Company type Root element iype Root element i	Company Add	dressType US-State					
US-Address element Company diagram iggram	Person UK	Address					
element Company diagram diagram Company type Address type Address properties content complex children Address Person annotation documentation Root element source <xs:element name="Company"> <xs:element name="Company"> <xs:element name="Company"> <xs:element source<br=""><xs:element name="Company"> <xs:element company"="" name="Compa</td><th>US</th><td>Address</td></tr><tr><td>element Company
diagram
diagram
Company
ype AddressType
AddressType
AddressType
Person
type AddressType
Not element
namespace
http://my-company.com/namespace
properties
content complex
children
Address Person
annotation
documentation
Root element
source
</rs: element name="> Root element </xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element>							
element Company diagram diagram Company type AddressType Root element Person type AddressType Person ype AddressType Person ype AddressType Person ype AddressType Person ype AddressType Person ype AddressType AddressType (xs element name="Company"> (xs : complexType> (xs : complexType> (xs : element name="Address" type="AddressType"/> (xs : element ref="Person" maxOccurs="unbounded"/> (xs : element>							
diagram Address image: Company image: properties namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element Noot element source <xs:element name="Company"> <xs:element< td=""> <xs:documentation> <xs:complextype> <xs:documentation> <xs:sequence> <xs:element name="Address" type="AddressType"></xs:element> <xs:element< td=""> name="Address" type="AddressType"/> <xs:element< td=""> <xs:element< td=""> name="Address" type="AddressType"/> <xs:element< td=""> <xs:element< td=""> <xs:element< td=""> <xs:element< td=""> <xs:element< td=""> <xs:element< td=""></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:element<></xs:sequence></xs:documentation></xs:complextype></xs:documentation></xs:element<></xs:element>	element Co	mpany					
Image: company type Image: company.com/namespace namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element Root element source <xs:element name="Company"> <xs:element name="Address" type="AddressType"> <xs:element maxoccurs="unbounded" ref="Person"> <th>diagram</th><th>Address</th></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element>	diagram	Address					
Image: company for element Person for element Image: company.com/namespace properties content complex children Address Person annotation documentation Root element source <xs::element name="Company"> <xs::element name="Company"> <xs::complex:< td=""> <xs::complex:< td=""></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::complex:<></xs::element></xs::element>		type AddressType					
Image: source Person namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element source source <xs:element name="Company"> <xs:element name="Company"> <xs:complextion> <tr< th=""><th></th><th></th></tr<></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:complextion></xs:element></xs:element>							
namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element Root element source <xs:element name="Company"> <xs:element name="Company"> <xs:complextype> <xs:complextype> <xs:element maxoccurs="unbounded" ref="Person"></xs:element></xs:complextype></xs:complextype></xs:element></xs:element>		Root element					
namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element Root element source <xs:element name="Company"> <xs:element name="Company"> <xs:complextype> <xs:complextype> <xs:element maxoccurs="unbounded" ref="Person"></xs:element> <th></th><th>type</th></xs:complextype></xs:complextype></xs:element></xs:element>		type					
namespace http://my-company.com/namespace properties content complex children Address Person annotation documentation Root element Root element source <xs:element name="Company"> <xs:element name="Address" type="AddressType"></xs:element> <xs:element maxoccurs="unbounded" ref="Person"></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element>		1					
properties content complex children Address Person annotation documentation Root element Root element source <xs:element name="Company"> <xs:element name="Address" type="AddressType"></xs:element> <xs:element maxoccurs="unbounded" ref="Person"></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element></xs:element>	namespace	http://my-company.com/namespace					
children Address Person annotation documentation Root element source <xs:element name="Company"> <xs:element name="Company"> <xs:annotation> <xs:documentation>Root element <th>properties</th><td>content complex</td></xs:documentation></xs:annotation></xs:element></xs:element>	properties	content complex					
annotation documentation Root element source <xs:element name="Company"> <xs:annotation> <xs:documentation>Root element</xs:documentation></xs:annotation></xs:element>	children	Address Person					
Root element source <xs:element name="Company"> <xs:annotation> <xs:documentation>Root element</xs:documentation> <xs:complextype> <xs:element name="Address" type="AddressType"></xs:element> <xs:element maxoccurs="unbounded" ref="Person"></xs:element> <t <="" td=""><th>annotation</th><td>documentation</td></t></xs:complextype></xs:annotation></xs:element>	annotation	documentation					
<pre>source <xs:element name="Company"></xs:element></pre>		Root element					
<pre>source <xs:element name="Company"></xs:element></pre>							
<pre> </pre>	source	<pre><xs:element hame="Company"> </xs:element></pre>					
<pre> </pre>		<pre><xs:documentation>Root_element</xs:documentation></pre> /xs:documentation>					
<pre><xs:complextype> <xs:sequence></xs:sequence></xs:complextype></pre>							
<pre><xs:sequence></xs:sequence></pre>		<xs:complextype></xs:complextype>					
<pre><xs:element name="Address" type="AddressType"></xs:element></pre>		<xs:sequence></xs:sequence>					
<pre><xs:element maxoccurs="unbounded" ref="Person"></xs:element></pre>		<pre><xs:element name="Address" type="AddressType"></xs:element></pre>					
 		<pre><xs:element maxoccurs="unbounded" ref="Person"></xs:element></pre>					

上の図はHTML フォームの最初のページを示しています。他のスキーマのコンポーネントが含まれていて場合、それらのスキーマもトキュメントの中についえられることにないます。



上の図は複合型がどのように表示されるか示しています。

lement US-Address/Zip				
diagram	Ezip type xs:positiveInteger			
namespace	http://my-company.com/namespace			
type	xs:positiveInteger			
properties	name Zip isRef 0 content simple			
source	<xs:element name="Zip" type="xs:positiveInteger"></xs:element>			
element US-Ad	ddress/State			
diagram	[≡] State type US-State			
namespace	http://my-company.com/namespace			
type	<u>US-State</u>			
properties	name State isRef 0 content simple			
source	<xs:element name="State" type="US-State"></xs:element>			
simpleType US	i-State			
namespace	http://my-company.com/namespace			
type	xs:string			
properties	name US-State			
used by	element US-Address/State			
source	<xs:simpletype name="US-State"> <xs:restriction base="xs:string"></xs:restriction> </xs:simpletype>			

上の図は要素や単純型がどのようて表示されるか示しています。

MS Word 出力オプショモ試してみてくださり Word ドキュメントがMS Word で開かれます。MS Word ドキュメントでレイパーレン機能を使いたは場合、 でたて」キーを押下したがジレンをクリックしてください

6 XML ドキュメント

このセケンヨンではXMLSpy を使用してXML ドキュメントを作成し、作業を行う方法を学習します。併せてXMLSpy のインデンデント編集機能を使用する方法も学習します。

目的

このセクションでは以下のことを学習します:

- AddressLast.xsd スキーマをベースにした新しいXML ドキュメントを作成する。
- 要素の型を指定して、その要素に対応するコンテンツモデルを検証に利用できるようにする
- 要素と属性を挿入し、グリッドビューならびにテキストビューにてインテリジェント入力へいいーを使ってコンテンンを挿入する
- XMLSpy からMicrosoft Excel にXML データを MS Excel から XMLSpy にコピーしなおす。この機能はグリッドビューにあるデータベーステープルビューで使用できます。
- データベーステーブルビューの並べ替え機能を使用してXML要素の並べ替えを行う
- XML ドキュメントを検証する。
- 3桁の内線番号に対応するためにスキーマの修正を行う

このセクションで使用されるコマンド

このセクションでは主にグリッドビューならいにテキストビュー、そして一部スキーマビューを使用します。以下のニアントが使われます:

- D ファイル 新規作成新し、種類のXMLファイルを作成します。
- 表示 | デキストビュー デキストビューゴリ 替えます。
- 表示 | グリッドビュー 拡張グリッドビュー さい 替えます。
- エメルト・テーブル(テーブルとして表示 1つの要素型の複数の出現をファットなテーブルで表示します。このビューはデータベーステーブルビュー(ませば単純にテーブルビュー)とします。アイコンはグリットビューで一般的なビューとテーブルビューを引い替えるために使用されます。
- **F7**. 整形式のチェックを引ます。
- ▶ F8. 割・当てられたDTD おはスキーママンけしてXML ドキュメントの検証を引ます。
- 副当てられたDTD まちはスキーマファイルを開きます。

6.1 新規 XML ファイルの作成

XMLSpyにて新しいXML ファイルを作成する際、スキーマのTD おはXML スキーマをベースにしたものにするかどうか 決定することができます。このセクションでは、既に作成したAddressLast.xsd スキーマをベースにした新しいファイルを 作成します。

新しいXML ファイルを作成するこは:

1. メニューオプショから ファイル 新規作成」を選択します。新規トキュメントを作成ダイアログが開かれます。

新規ドキ	ュメントを作成	×
作成した	いドキュメントの種類を選択してください。	
wml	Wireless Markup Language	
wsdl	WSDL Web Service Description v1.1	
wsdl	WSDL Web Service Description v2.0	
×brl	XBRL Instance	
xdr	XML-Data Reduced Schema	
xhtml	XHTML 1.0 Strict	
xhtml	XHTML 1.0 Transitional	
xhtml	XHTML 1.1	=
xlsx	MS Office Excel OpenXML	
xml	DocBook 4.5 Article	
xml	DocBook 4.5 Book	
xml	EJB 2.0 Deployment Descriptor	
xml	Extensible Markup Language	
vml	. PEE 1.3 Application	•
Style	Wision スタイルシートを選択 OK キャンセ	ll

2. ダイアログからExtensible Markup 言語 Extensible Markup 言語 エント 超選択し じK」で確定しま す。ポップアップが表示され、XML ドキュメントのベース CDTD おさはスキーマを使用するか尋ねられます。

新規ファイル	? ×
特定のスキーマや DTD に従った新規の XML 文章を作成しようとしている のであれば、 XMLSpy は自動的に適切な値を加えることができます。 -そ うであれば、 ドキュメントの作成元を DTD かスキーマから選んでください:	OK キャンセル
 ○ DTD ● スキーマ 	
それ以外の場合、キャンセルボタンをクリックして空のドキュメントから 始めてください。	

- 3. スキーマランオボタンをクリックし、 **OK」**で確定します。 次のダイアログが表示され、 XML トキュメントのベースとなるスキーマファイルを選ぶようにはされます。
- 4. 参照もしくおうシドラボタを使用して、スキーマファイルを見つけます。ウシドラボタを使うことでXMLSpy にあ る全てのファイルとプロジェクトを閲覧することができます。AddressLast.xsd を選択し(チュート)がのイントロ ダケシュを参照(たさい)、 **OK」**により確定します。スキーマで定義された要素を含むXML ドキュメントがメイン ウンドンで開かれます。
- 5. ゲリッド」タブをクリックし、グリッドビューを選択します。
- 6. グリッドビューに表示されるドキュメントの構造に注目してください要素をクリックして選択を決めてみてくださいドキュメントは以下のように見えているはずです。

	version	1.0
	encoding	UTF-8
Co	mpany	
	= xmins	http://my-company.com/namespace
	= xmlns:xsi	http://www.w3.org/2001/XMLSchem
	xsi:schemaLoca	http://my-company.com/namespace AddressLast.xsd
	Address	
	Person Manager=	

7. Address の横にある アイコンをクリックして、Address の子要素を表示します。ドキュメントは以下のようてな るはずです。

▲ Co	mpany	
	= xmins	http://my-company.com/namespace
	= xmins:xsi	http://www.w3.org/2001/XMLSchema-insta
	= xsi:schema	http://my-company.com/namespace AddressLast.xsd
	Address	
		Name
		() Street
		() City
	🔳 Person Manag	er=

6.2 要素の型の指定

Address の子要素はプローバル複合型 AddressType とて定義されています (XML スキーマ AddressLast.xsd 内で定義されているスキーマビューのコンテンソモデルが以下のスクノーンショルでは示されます)。



しかしてでは一般的な住所の型よれ、US おはUK の住所を使用します。AddressLast.xsd スキーマでUS-Address とUK-Address とりクローバリ複合型を作成したのを覚えているでしょうか?US-Address のエレテンチ デレを以下に示します。



XML ドキュメントこて Address 要素が(一般的な Address Type 型ではなく拡張された Address 型 (us-Address かuk-Address)の要件を満たすためには 必要な複合型を Address 要素の属性 として指定しなければ なりません。

次を引ます:

1. XML ドキュメントこて Address 要素を右クリックして、追加 | 属性」をメニューから選択します。

() Name	挿入①		属性(<u>A</u>)	Ctrl+Shift+I
() City	追加(<u>A</u>)	, io	要素(<u>E</u>)	Ctrl+Shift+E
	子を追加(<u>H</u>)	•	テキスト①	Ctrl+Shift+T
			CData(<u>C</u>)	Ctrl+Shift+D
			- 11 (()	0.1 01 W. M

Address 要素に属性フィールトが追加されます。

2. xsitype か属性の名前として入力されていることを確認します(以下のスクリーンショントを参照)

3. 「タブ」キーを押して隣の他)フィールド」を動します。

Company						
	= xmins	http://my-company.	com/namespace			
	= xmins:xsi	http://www.w3.org	http://www.w3.org/2001/XMLSchema-insta			
	≡ xsi:schema	. http://my-company.com/namespace AddressLast.xsd				
	Address					
		= xsi:type	US-Address			
		() Name				
		() Street				
		() City				
	🗷 Person Manag	jer=				

- 4. 属性の値にUS-Address と入力します。
- ・
 KSi プレフィクスを使うとでXML ドキュシトインスタノスで類連した特別なXML スキーでを使えるようないます。スキーでをXML ファイルに関連付けたときにxSi プレフィクスの名前空間が、自動的にドキュメントに追加されたことに注目してできょう。上の場合はAddress要素の型を指定しました。さらに詳し、情報についてはXML スキーマの仕様(英文)を参照できょう。

6.3 グリッドビューを使ったデータの入力

XML ドキュメントにデータを入力することができます。データの入力を行うこは:

1. Name の値フィールトをダブルクリックし(おは矢印キーを使い), US dependency と入力し Enter」により 確定します。

	= xmins	http://my-company	.com/namespace	
	= xmins:xsi	http://www.w3.org/2001/XMLSchema-instar		
	= xsi:schema	http://my-company.com/namespace AddressLast.xsd		
	Address			
		= xsi:type	US-Address	
		() Name	US dependency	
		Street		
		() City		

- 2. 同様の方法で Street とCity にも値を入力します 例えば Noble Ave とDallas)。
- 3. Person 要素をクリックし、 Delete」キーを押下して Person 要素 結果にます 次のセクションで追加し直します)、この作業が終わると Address 要素全てが バライトされます。
- 4. Address 要素の子要素をクリックして、それ以外のAddress の子要素が選択されていない状態にします。 XML ドキュメントは以下のようこなっているはずです:

🔳 XI	NL				
A Co	mpany				
	= xmins	http://my-company	http://my-company.com/namespace		
	= xmins:xsi	http://www.w3.org/2001/XMLSchema-insta			
	= xsi:schema.	 http://my-company.com/namespace AddressLast.xsd 			
	Address				
		= xsi:type	US-Address		
		() Name	US dependency		
		() Street	Noble Ave		
		City	Dallas		

6.4 テキストビューを使ったデータの入力

上記に加えて、テキストビューはおキストの編集作業を楽にする多の構造的な編集機能があります。これらの機能はおキストビュー設定ダイアログにて有効無効の変更を行うことができます(表示 | テキストビュー設定」、下のスクレーンショットを参照)。

3 テキスト ビューの設定			×
余白 ✓ 行番号の余白(L) ブックマークの余白(B) ✓ 折りたたみ余白(F)	タブ タブサイズ: 4 ● タブを挿入(T) ◎ スペースを挿入(S)	視覚補助 ✓ インデント ガイド(I) □ 行末マーカー(E) □ 空白マーカー(W)	OK キャンセル 適用(A)
─▼ 自動ハイライトの有効化			
◎ 選択された単語に一致する			
◎ 次の選択に一致する:	1 🔻 文字		
📝 大文字と小文字を区別			

下のスクリーンショットは、上のダイアログの設定に合わせたテキストビュー内の現在のファイルを表示しています。



左側に3つのマージンがあります:()行番号のマージン、(i)ブックマークマージン(2つの青いブックマークを含みます), (iii)(XML要素の表示を展開折りまたみすることのできる)ノース折りまたみマージン。

更に インデントガイド 行の終わりアーカー、空白マーカーなどの視覚補助は チェンガおよびチェングの解除によりオンオフす ることができます。テキストビュー設定ダイアログの視覚補助ペインでオンオフすることができます。(上のスグリーンショントを参 照)、上のスグリーンショントは、インデントガイトが切り替えられており、1つのインデントガイトをAddress 要素で表示して ります。

天: テキストビューは関連した「刷およびブックマークは機能はコのチュート」のLの <u>テキストビューの設定</u>のセクションで 説明されています。

テキストビューでの編集

テキストビューに表示される以下の余白 (マージン)の有効 無効を設定することができます。

以下を引ます:

- 1. メニューオプションの 表示 | テキストビュー」を選択するか、テキストダをクリックします。これで XML ドキュメント がテキストの形で構文カラー強調表示と共に表示されます。
- 2. テキストカーソルを Address 要素の終了タブの直後に移動させ、 Enter」を押して新しい行を追加します。
- 3. この位置で小なりかこくを入力します。(スキーマの定義により)この位置で許されている全ての要素がポップアップ て表示されます。この位置では Person 要素だけが許されているので、それだけが表示されます。

- 4. Person エント 超選択します。Person 要素が Manager 属性と共に 挿入され Manager 属性の 植フール ドの 内部 に カーソリ が移動します。
- 5. Manager 属性のポップアップからtrue を選択します。



6. カーソルを行来に移動し(例えば End」キーを使うこができます)、スペースバーを押します。新し、ポップアップが 開かれ、この位置で許されている属性の「人」が表示されます。同様に、属性入力へリパーにて使用することのでき る属性が赤色で表示されます。Manager 属性はグレー」なっていますが、それはその属性が既に使われたため です。



7. 矢印キーを使って Degree を選択し Enter」を押します。これておおりのポップアップが表示され、既に定義された列挙(BA、MA、おけまPhD から選択を行うこができます。(列挙はXML スキーマによ、許可されている値です。)



8. 矢印キーを使って BA を選択し、 Enter」で確定します。 さらこカーソルを(End」キーを使うなどして)行末に 移動し、ペースバーを押します。 今度はManager とDegree が属性入力へリパーでグレーナないます。.



9. 矢印キーで Programmer を選択し Enter」を押します。

ſ	<pre>- </pre>	fľ
		talse
		true

- 10."f" キーを押して Enter」を押します。
- 11.カーソルをう末まで移動し、大な「記号 > を入力します。 XMLSpy がPerson に必要な子要素を自動的に 挿入します (省略可能なTitle 要素が挿入されて、ないによご注目してください)。 全ての要素には開始と終了 タブがありますが、内容は記述されていません。

<pre>Gompany xmlns="http://my-company.com/namespace" xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" http://wy-company.com/namespace Addresslast.ysd">http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" http://wy-company.com/namespace</pre>
Address weight and "UC Address"
Address Xs1:type= US-Address >
<name>US dependency</name>
Street Noble Ave Street
<city>Dallas</city>
-
<pre></pre>
<pre></pre>
diasts (liasts
<pre><phoneext></phoneext></pre>
<pre><email></email></pre>
/Person
- NPEI SUI A
└

これてテキストビューにてPerson にデータを入力することができますが、グリッドビューに移動して、トキュメントの編集時にどれば、簡単にビューを変更できるか見てみましょう

グリッドビューへ変更

グリッドビューへ変更するは、メニューオプショの 表示 | 拡張グリッドビュー」 を選択するか、 ゲリッド」 タブをクリックします。 新たし加えられた Person の子要素が いっています。



それではドキュメントの検証を行い見つかたエラーを修正していきましょう

6.5 ドキュメントの検証

XMLSpy ではXML ドキュメントを2つの方法で評価できます:

- 整形式のチェック
- 検証

上のうちのどちらかが失敗した場合、ドキュメントを正しく修正しなければないません。

整形式のチェック

開始ダブと終了ダブがマッチし、要素が正しくネストされ、間違った文字があった」必要な文字(例えばセミエレの無しエン ティティなど)がなかったししなければ、そのXMLトキュメントは整形式であると言します。

どのビューでも整形式のチェックを行うことができます。テキストビューでやってみましょう、整形式のチェックを行うには、メニュー オプションの XML | 整形式のチェック」 る選択するか、「77」キーを押すか、「27」をフリックします。メインウィンドウの下部にあるメッセージウィンドウィンドは整形式であると、らメッセージが表示されます。

ッセージウンドカ出力には9つのタガあることは自して代は、評価された結果の出力にカケティブなタブはすして行われます。従って、タブ1て整形式のチェックをしておさいから、その結果を残したまま、検証をすう前に、タブ2に移ることができます(タブ11で残ったままだと検証の結果で上書きされてしまします)。

メッセージ	×
💦 🥑 ファイル C¥Altova¥XMLSpy¥Examples¥Tutorial¥CompanyFirst.xml は整形式です。	
<u>></u>	

注意:このチェックではスキーマとの整合性を考慮したXMLの構造はチェックされません。スキーマとの整合性の評価は検証にて行われます。

検証

XML構造が指定されたスキーマの記述に従ったものであれば XML ドキュメントは妥当とないます。

XML ドキュシントの妥当性をチェンクするこは、まずグリットビューを選択し、メニューオプショから XML | 検証」を選択す るか、「F8」キーを押すか、または「」をクリックします。ファイルが妥当でないというエラーシャセージがシャセージウントドに表示されるはずです。これはAddress内のCity要素の後に必須要素が必要だからです。スキーマをチェックしてみると、 (xsi:type 属性を使って Address 要素から指定した)US- Address 複合型の中にはていま 要素の後には Zp 要素と State 要素が続かなければならないと記述されているのが分かるでしょう

不正なトキュメントの修正

ドキュメントが不正しておる原因箇所(この場合はCity要素)がバライトされます。

🗷 XML				要素
🔺 Company				
	= xmins	http://my-compar	iy.com/namespace	Name
	= xmlns: <i>xsi</i>	http://www.w3.org/2001/XMLSchema-instanc e		() Street () City
	z <i>xsi:</i> schemaLoc	. http://my-company.com/namespace C:¥PROGRA~1¥Altova¥XMLSpy2010¥Example s¥Tutorial¥AddressLast.xsd		() State
	Address			
		= xsi:type	US-Address	
		() Name	US dependency	· 追加 插入 子 存追加
		() Street	Noble Ave.	
		() City	Dallas	属性
	🗷 Person Manager=	false Degree=BA P	rogrammer=false	z xsi:type

それでは画面右上にある要素入力へいいーを見てたさい。Zip 要素が感嘆符(!)と共に表示されています。これはその要素が現在のエンテキストで必須であることを示しています。

検証エラーを修正するしよ:

- 2. 「タブ」キーを押し州の郵便番号(04812)を入力し、Enter」で確定します。これで要素入力へいパーが
- 2. 「タブ」キーを押し州の郵便番号 (04812)を入力し Enter」で確定します。これで要素入力へりパーが State 要素が必須であることを示すようことはます(感嘆符と一緒に表示されます)。 以下のスクリーンショルを参照して 代ごさし

	ИL		A	Elements
A Co	 Company 			() [!] State
	= xmins	http://my-company	.com/namespace	() Name
	= xmins:xsi	http://www.w3.or	p://www.w3.org/2001/XMLSchema-ir	
	xsi:sche http://my-company.co AddressLast.xsd		.com/namespace	() City () Zip
	Address			
		= xsi:type	US-Address	Append In
		() Name	US dependenc	
		() Street	Noble Ave	Attributes
		() City	Dallas	= xsi:type
		() Zip	04812	
	Person			

3. 要素入力へルパーにて、State要素をダブルクリックします。その後「タブ」キーを押し州の名前(Texas)を入力し、Enter」で確定します。これで要素入力へルパーコングレーの要素しか表示されなくなります。これは Address にはましジ須の子要素が無いことを表しています。

	XML			A	Elements
()	Co	edited v alex pilz	with XML Spy v4.0 U (: (altova)	http://www.xmlspy.com)	 Name Street
	Company				City
		= xml	ns http://my-company	http://my-company.com/namespace	
		= xml	ns:: http://www.w3.o	rg/2001/XMLSchema-inst	() State
		= xsi :e	s http://my-company AddressLast.xsd	http://iny-company.com/namespace AddressLast.xsd	
		🔺 Add	ress		[[[[[[[[[[[[[[[[[[[
			= xsi:type	US-Address	Attributes
			() Name	US dependency	
			() Street	Noble Ave	a vsi.type
			() City	Dallas	
			🜔 Zip	04812	
			() State	Texas	
		A Pers	son Manager=true De	gree=BA Programmer=fal	

ドキュメントを完成させ、再度検証する

それでは再度検証を行う前にPerson要素の情報を入力しドキュメントを完成させましょう。

以下を引ます:

1. First 要素をクリックして、名前(例えばFred)を入力し、Enter」で確定します。

A Perso	n	
	= Manager	true
	= Degree	BA
	Programmer	false
	() First	Fred
	() Last	
	PhoneExt	
	() Email	

2. 同様の方法で、Personの子要素全て(Last、PhoneExt、そしてEmail)のデータを入力してたさい。PhoneExt の値は(スキーマで指定既に指定したように)99以下の整数でなければならないとは注意してたさい。XMLド キュメントはグリッドビューで以下のようこなっているはずです。

Company		
= xmins	http://my-company.co	om/namespace
= xmlns:xsi	http://www.w3.org/2	2001/XMLSchema-instance
= xsi:schemaLoca.	 http://my-company.com C:\PROGRA~1\Altova utorial\AddressLast.> 	om/namespace \\XMLSpy2006\Examples\T \sd
Address		
	= xsi:type	US-Address
	() Name	US Dependency
	() Street	Noble Ave.
	() City	Dallas
	() Zip	04812
	() State	Texas
Person		
	= Manager	true
	= Degree	BA
	Programmer	false
	() First	Fred
	() Last	Smith
	() PhoneExt	22
	() Email	Smith@work.com

3. 2 を再度クリックストキュメントが受当かどうかのチェックを行って作さい、メルセージウィントプにファイルが受当であると言うシャセージが表示されます。



4. メニューオプショから「ファイル」上書を保存」を選択し、XML ドキュメントに適切な名前、例えば CompanyFirst.xml)を与えます。Tutorial フォレダー内に世況にCompanyFirst.xml ファイルかあることご注意してく ださい。作成したファイルを保存する前に、そちらのファイルの名前を変更する必要があるかもいれません。

大モ:保存する際にXML ドキュメントが受当てある必要はありません。不正なドキュメントを保存しようすると、その旨の警告が表示されますが、不正のままの状態で保存した」のであれば、保存することは可能です。

6.6 要素と属性の追加

Person この段階ではドキュメント内にPerson 要素は1つしかありません。

新たCPerson要素を追加すること:

- 1. Address 要素の左にあるグレーのサイドバーをクリックして Address 要素を折りたまます。これによどユー内 で使える空間を増やせます。
- 2. グリッドビューの Person 要素をクリックして Person 要素全体を選択します。要素入力へいしつ 追加」タ ブーン アーマン 要素が使えるようことうていることご注目して代きい

🔳 XM	L					
(! C	edited with alex pilz (a	n XML Spy v4.0 U (http Itova)	://www.xmlspy.com) b			
Cor	прапу					
	= xmins	http://my-company.com	m/namespace			
	= xmin http://www.w3.org/2001/XMLSchema-instar					
	= xsi:s	http://my-company.com/namespace AddressLast.xsd				
	🔳 Addre	ss xsi:type=US-Addre	e=US-Address			
	A Perso	1				
		= Manager	true			
		= Degree	BA			
		= Programmer	false			
		🔇 First	Fred			
		() Last	Smith			
		PhoneExt	22			
		() Email	Smith@work.com			

3. 要素入力へいーのPerson 要素をダブルクルクします。新し、Person 要素が必須子要素と一緒に追加されます(下のスクレーンショルを参照)。Person 要素で省略可能なTitle 子要素が追加されていた。ことに注目してでは、

Company					
	= xmins	http://my-company.com	/namespace		
	= xmins:xsi	http://www.w3.org/20	01/XMLSchema-instance		
	xsi:schemaLoca	http://my-company.com C:\PROGRA~1\Altova\X	/namespace MLSpy2006\Examples\Tu	torial\AddressLast.xsd	
	🖬 Address xsi:type=	US-Address			
	Person				
		Manager	true		
		Degree	BA		
		Programmer	false		
		() First	Fred		
		() Last	Smith		
		PhoneExt	22		
		Email	Smith@work.com		
	Person (1)				-
		= Manager	() First	() Last	PhoneExt
	1				

- 4. Person 開始タズの大なかっこ> の直前にカーソレを移動させ、属性入力ヘリレーの 追加」タズこて、 Programmer エントをダブルクリックします。これによりManager 属性の後にProgrammer 属性が追加され、属性入力ヘリレーではProgrammer がブレーごとます。
- 5. 新し IPerson 要素のManager 属性をクリクします。属性入力へいにを見てたさい、既に入力された Manager エントガヴレーになっています。Manager 属性に関する情報が表示されている情報ウシンドンにも注 目してたさい Manager 属性は必須の属性であるこか自動的に追加されています。Programmer 属性は追 加されていません。
- 6. 属性入力へいーの 追加」タズこて、Programmer エントをダブルクトクルます。これで空の Programmer 属性がManager 属性の後に挿入されます。

A Perse	on		Append Insert A
	= Manager	true	Attributes
	= Degree	BA	
	= Programmer	false	= xsi:type
	First	Fred	■ Manager
	() Last	Smith	Programmer
	PhoneExt	22	
	() Email	Smith@work.com	
A Perse	DN		
	= Manager		
	= Programmer		Append Insert
	First		
	() Last		_ Entities

これで属性入力へりいーのProgrammer 属性がりレーゴンます。

このビューでも要素の情報を入力することは可能ですが、Personの様な複数回出現する構造の入力に適したグリッドビューのデータベースデーブレビューは小替えましょう

6.7 データベース /テーブルビューでの編集

グリッドビューの中には複数回出現する要素の編集に適したデータベーステーブルビューとで特別なビューが存在します(これ以降はテーブルビューと表記します)。個々の要素の型がテーブルとて表示されます。要素の型がテーブルとて表示されたとき その子供 (要素ならびに属性)は別として表示され、複数回出現する情報が行(レコード)とて表示されます。

要素の型をテーブルとて表示するこよ(複数回出現する)任意の要素を選択し、ツールバーのテーブルとて表示アイコン目をクリックします(メニューオプションの「XML | テーブル | テーブルとて表示」を選択しても同様の操作が可能です。)、この操作で要素の型がテーブルとて表示されます。複数回出現する子要素もテーブルとて表示されます。テーブルビューは拡張グリットビューで提供され、XML形式のファイル全て、XML、XSD、XSLなどの編集で使用できます。

テーブビューの利点

テーブルゴーコま以下のようは利点があます:

- ドラ・グアンドドロップで列のヘッダーを変更することができます。 つまし テーブル され応する全ての出現に対して、XMLドキュメント内での子要素ませよ子属性の位置を修正することができるということです。
- テーブルは KML | テーブル 昇順に並べ替え」まは KML | テーブル 降順に並べ替え」を使うとはおン ートを行うことができます。
- 新し行 例えば新し 要素 を XML | テーブル 行を挿入」 によ 追加することができます。
- サードパーテ、製品からませまサードパーテ、製品へエピーといけだううことができます。
- テーブルでもインテレジェント編集機能が使用できます。

テーブルで要素の型を表示する Person 要素をテーブルとて表示するには:

1. グリッドビューにて、Person テキストの1つの近後クリックして Person 要素を選択します。

🔳 Addres	ss xsi:type=US-Addres	s
Persor	1	
	= Manager	true
	= Degree	BA
	= Programmer	false
	() First	Fred
	() Last	Smith
	() PhoneExt	22
	() Email	Smith@work.com
A Persor	۱	
	= Manager	
	= Programmer	
	🔇 First	
	🔇 Last	
	PhoneExt	

2. メニューオプショから XML | テーブル テーブルとて表示」を選択するか、テーブルとて表示アイエン 開 をクリックします。 2つの Person 要素が1つのテーブルに統合されます。 子要素と子属性の名前が列のへがってより、それらの値がテーブルが行 (レニード)にないます。

A Co	mp	bar	ıy								
	=	хп	nins	http://my-comp	http://my-company.com/namespace						
	=	хп	nin	http://www.w3	3.org/2001/XI	MLSchema-instance					
	xsi:s http://my-company.com/namespace AddressLast.xsd										
		Ac	idres	ss xsi:type=US-	Address						
		Pe	rsor	1 (2)							
				Manager	Degree	Programmer					
			1	true	BA	false					
			2								

3. メニューオプショから 表示 | 最適な幅」を選択するか、最適な幅アイコン 一 をクリックし、 テーブルの列幅を最適化します。

注意:テーブルを選択(テーブルの要素名をクリック)し、テーブルとして表示アイコン 間 をクリックすることで、テーブルビュー を学业し、ドキュメント内にある個々の要素を表示することができます。

テープルビューでコンテンツを入力する

2つ目のPerson 要素のエテンソ人力を行うすよ 2行目にあるれぞれのテーブルセレをダブルクリクレ 何られのデータ を入力します(下のスクレーンショルを参照)。しかしファイルを妥当なものにするためけまPhoneExt は 99までの整数で なければなられ にを忘れれてでたき)。テーブルのセル内でもヘステジェント編集機能は使用できます。従ってオプション がある箇所 (Boolean エレテンツとDegree 属性の列挙)ではドロップダブノストががら値を選択できます。

Person (3)										
П		= Manager	= Degree	= Programmer	() First	() Last	PhoneExt	() Email		
П	1	true	BA	false	Fred	Smith	22	Smith@work.com		
	2	false	MA	true	Alfred	Aldrich	33	Aldrich@work.com		

注意:テーブルとて表示されている要素や属性に対しても入力へルパーは使用できます。例えば要素入力へルパー内の Person エティをダブルクトクすることで、テーブルに新しい行いのの要素」を追加することができます。

サードパーティ製品とのXMLデータのコピー

スプレッドシートで表示されたデータをXMLSpyのXMLドキュメントとサードパーティの製品間でコピーすることができます。このデータはXMLSpy内でXMLデータとして扱うことできますし、サードパーティ製品にてネイティブフォーマットとして使用することをできます。このセグションではExcelデータントトとデータのや「取」の方法を学習します。

以下を引ます:

1. 行ラベル1をクリックします。 でたけ」キーを押下しなが空ラベル2をクリックします。 これてテーブル内の両方の行が選択されます。

Address xsi:type=US-Address									
Person (2)									
		= Manager	= Degree	= Programmer	() First	() L			
	1	true	BA	false	Fred	Smith			
	2	false	MA	true	Alfred	Aldric			

- 2. メニューオプショから **編集 | 構造化テキストとしてコピー」**を選択します。このコマンドで要素が画面に表示されたますの形式でクリップボードニピーされます。
- 3. Excel に小替え XML デーをExcel ワーケートはいけます (Ctrl+V)。

A	В	С	D	E	F	G	Н
TRUE	BA	FALSE	Fred	Smith	22	Smith@wo	rk.com
FALSE	MA	TRUE	Alfred	Aldrich	- 33	Aldrich@w	ork.con

4. Excel にて新し、行にデータを入力します。PhoneExt 要素に3桁の数字(例えば 444)を入力して代さい。

A	В	С	D	E	F	G	Н
TRUE	BA	FALSE	Fred	Smith	22	Smith@wo	rk.com
FALSE	MA	TRUE	Alfred	Aldrich	- 33	Aldrich@w	ork.com
TRUE	PhD	FALSE	Colin	Coletti	444	Coletti@w	ork.com

- 5. Excel でテーブルのデータをマークレ、 Ctrl」+C によデータをクリップボードにコピーします。
- 6. XMLSpy に戻ります。
- 7. XMLSpy のテーブル内の左上にあるデータセルをクリックして、 編集 | 貼付け」を選択します。

🔳 A	Address xsi:type=US-Address										
A P	Person (3)										
		= Manager	= Degree	= Programmer	First	() Last	Pho				
	1	TRUE	BA	FALSE	Fred	Smith	22				
	2	FALSE	MA	TRUE	Alfred	Aldrich	33				
	3	TRUE	PhD	FALSE	Colin	Coletti	444				

- 8. テーブルゴム データの表示されます。
- 9. メニーオプショの 編集 | 置換」を使用して、大文字で表示された TRUE とFALSE をそれぞれ小文字の true とfalse に変更します。

列のコンテンソによりテープルをノートする

テーブビュー内のテーブルありに対して昇順ませお解順でノートをテランができます。ここでは描字(Last)に従い Person テーブルをノートしてみましょう

列の内容に対してテーブルのノートを行うさま

1. Last 列のへッダをクリックして列を選択します。

Address xsi:type=US-Address										
Person (3)										
		= Manager	= Degree	= Programmer	() First	() Last	() Phone			
	1	true	BA	false	Fred	Smith	22			
	2	false	MA	true	Alfred	Aldrich	33			
	3	true	PhD	false	Colin	Coletti	444			

2. メニューオプショから XML | テーブレ 昇順に並べ替え」を選択するか、昇順ソートアイコン 30 をクリックする ことで、テーブ 1全てがソートされます。列はノイライトされまましてはます。

Address xsi:type=US-Address										
Person (3)										
		= Manager	= Degree	= Programmer	() First	() Last	Phone			
	1	false	MA	true	Alfred	Aldrich	33			
	2	true	PhD	false	Colin	Coletti	444			
	3	true	BA	false	Fred	Smith	22			

)ソート操作によテーブルの表示が変わっただけではなくもとものXMLトキュメントも変更されています。つまり Person 要素の順序がLast 要素の内容に従ってソートされたのです(テキストダンをクリックしてテキストビューで 変更を確認してみてください)。

3. メニューオプショから XML | 検証」を選択するか、 F8」キーを押します。 444 'と じ値は PhoneExt 要素では許されな と じか容のエラーメセージが表示されます (スクリーンショルを参照)。 不正な PhoneExt 要素が パイライトされます。



詳細」を展開して、値が99以下ではためPhoneExtが不正であることを確認してくださいエラーメウレージのレクをクリンクすることでXML ファイル内のエラーが発生した場所にジャンプすることができます。内線番号の為に設定した値のレンジではこの値はカバーされません。この値を使って妥当性を確保するためにスキーマを修正する必要があります。次のセグションではスキーマの修正を引ます。

6.8 スキーマの修正

2桁の内線番号では全ての番号をカバーできょうにからかました。それでは有効な番号を新まで増やすことにしましょうそのためははスキーマの修正が必要にないます。XMLスキーマの修正はXMLドキュタントを閉じることなく行うことができます。

以下を引ます:

1. メニューオプショから **DTD/スキーマ| 定義へ移動」** な選択するか、定義へ移動アイコン¹⁹⁹¹ をクリックします。 割・当てられたスキーマ この場合は AddressLast.xsd が開かれます(以下のスクリーンショルを参照), 必 要に応じてスキーマビューへ移動します。オプションダイアログ(「シール| オプション」)のファイルタイプタブでは、各 ファイルタイプ え対応したデフォレトのビューを変更することができます。

Ē			×
	element	Company	ann:Root element
	complexType	AddressType	ann:
	complexType	US-Address	ann:
	simpleType	US-State	ann:
	complexType	UK-Address	ann:
-	element	Person	ann:

2. スキーマビューに表示されたグローバルPerson 要素のダイアグラム表示アイコン・・・ をケルクル、Person 要素のエンテンソモデルにて PhoneExt 要素を選択します。ファセットダン内にファセットデータが表示されます。



3. ファセットタブ内にあるmaxInclの値フィールトをダブルクリックし、値を 99から 999に変更します、 Enter」で 確定します。

[■] PhoneExt	詳細「単純型
type xs:integer	ファセット
Email	minIncl
type xs:string	maxIncl 999
	minExcl

- 4. スキーマトキュメントを保存します。
- 5. Ctrl」+ タブ」キーを押してXML ドキュメントに移動します。

6. 2 をクリクして XML ドキュメントを再度検証します。



メッセージウィンドウェファイルが妥当てあると、ウベッセージが表示されます。XML ドキュメントは修正されたスキーマに 適合することが分かました。

7. メニーオプショから ファイル 名前を引けて保存」を選択し CompanyLast.xml と 名前でファイルを保存 します (XMLSpy に付随して るオジナルの CompanyLast.xml ファイルを列えば CompanyLast_orig.xml の様な名前に変更するのを忘れな でください。

た:XMLSpy に付随しているCompanyLast.xml ファイルはTutorial フォルダーの中は収められて、ます。

7 XSLT 変換

目的

XML ファイルを変換するXSL スタイルシートを使ってXML ファイルからHTML ファイルを生成します。変換と言いますが XML ファイルを変更するのではなく新しいファイルが作成されます。 変換」といず語は慣例的な表現です。

方法

変換を行うことであってあるのです:

- 既に定義されたXSL ファイル Company.xsl をXML ドキュメントに割り当てます。
- XMLSpy のインターフェースにある2つのAltova XSLT内蔵エンジンのち1つを使用して変換を実行します(下記参照)。

このセクションで使用されるコマンド

以下の XMLSpy コマンドがこのセクションでは使用されます:

- 図 **XSL/XQuery | XSLの割り当て」:**アケイズなXML ドキュメンHこXSL ファイルを 割)当てます。
- KSL/XQuery | XSLへ移動」: XML ドキュメントには参照されている XSL ファイル
 胡きます。
- KSL/XQuery | XSL変換 F10)」、おけは対応するシール(ーアイコン:アケイブ なXML ドキュシントをそのXML ファイルに割り当てられたXSL スタイルシートを使って変換します。XSL ファイルがお注割り当てられていない場合、割り当ておううプロンプトが表示されます。
- 注意: XMLSpy は 1.0 2.0 および 3.0XSLT エシンを搭載しています。正しい エシンは xsl:stylesheet かxsl:transform 要素のバージョノ情報をもとに、XMLSpy により自動的に選択されます。 このチュートリアレ の変換では XSLT 1.0 スタイルシートが使用されます。 XSL 変換 コマンドが起動されたときては、変換のための XSLT 1.0 エシンが自動的に選択されます。

7.1 XSLT ファイルの割り当て

CompanyLast.xml アイルにXSLT アイルを割当てるは::

- 1. メインウンドウのCompanyLast.xml タブをクリックして CompanyLast.xml をプライブにし、テキストビュー は川 替えます。
- 2. メニューオプションの XSL/XQuery | XSLの割り当て」 を選択します。
- 3. 参照ボタをクリクレス Tutorial ブガレダーからCompany.xsl ファイルを選択します。 (ML ドキュメント内に 記述される) (SL ファイルへの) なを目対 いなこした 場合は CompanyLast.xml からの相対/ひにする」 にチェンクを入れます。

XML Spy	? <mark>×</mark>
ファイルを選択: Companyset ● 参照… ウインドウ… ✓ CompanyLastomi に対する相対パスにする	ОК 4+>>セル
ハードディスク、リモートサーバ、またはグローバル リソースからファイルを選択するか、現在 XMLSpy で開かれ ているウィンドウを選択してください。	

- 4. OK」をクリクしてXSL ファイルをXML ドキュタントに割り当てます。
- 5. グリッドビューはい替えて割当てを確認します(下のスクリーンショルを参照)。

🔳 XML					
(Comment	edited with XML Spy v4.0 U (http://www.xmlspy.com) by alex pilz (attova)				
xml-stylesheet type="text/xsl" href="C:\Program Files\Altova\XML Spy Suite\Examples\Tutorial\Company.xsl"					
Company					
	zmins http://my-company.com/namespace				
	= xmins:x http://www.w3.org/2001/XMLSche				
	xsi:sc http://my-company.com/namespace AddressLast.xsd				
	Address				
		= xsi:type US-Address			

XSL ファイルを参照するXML-stylesheet 処理命令がXML ドキュシントファイルに挿入されます。 EompanyLast.xml からの相対バスにする」、チェングボックスを有効にした場合、表示されるバンが相対パスにない。 ない、そうでは、場合は絶対バンが使用されます。

7.2 XMLファイルの変換

割じてをテナXSLファイルを使用してXMLドキュメントを変換するしは:

- 1. XML ファイルがアケティブになっていることを確認します。
- 2. メニーオプショから **XSL/XQuery | XSL変換 F10)」**
 超期するかアイコン
 「センクレントで参照されて、 GXSL スタイルシートを使用した変換が開始されます (Company.xsl ファイルはXSLT 1.0 ドキュペントなので、変換しはAltova XSLT 1.0 エンジンが自動的に 選択されます)。 出力されたドキュペントがXSL output.html と 「ファイル名でプランザービューに表示されま す (HTML 出力ファイルが生成されない場合、オプションダイプログ (ツール) オプションのXSL タブにて、 出力 ファイルのデフォルト拡張子」をhtml に設定してください。 HTML ドキュペントはた社 (Company)の 報と、人物 (Person)の精報が表示されます。

	Your Company							
Name: US dependency Street: Noble Ave. City: Dallas State: Texas Zip: 4812								
First	Last	Ext.	E-Mail	Manager	Degree	Programmer		
Alfred	Aldrich	33	Aldrich@work	false	MA	true		
Colin	Coletti	444	Coletti@work.com	true	Ph.D	false		
Fred	Smith	22	Smith@work.com	true	BA	false		
							-	
Text	Browse	r 🕶						

天:出力ファイルの内容にテーブルのヘッダだけが表示され、その内容が表示されない場合、スキーマのターゲット 名前空間を このチュートアルのはじめこある<u>名前空間の定義</u>で記述された通りこなっているか確認してくたさい。名前空間はすべてのファイル(スキーマ、XML、そしてXSL)で一致していなければなりません。
7.3 XSLファイルの修正

XSL ドキュメントを修正することで、出力ファイルを変更することができます。例えば、HTML 出力のテーブルの背景色をライムが黄色に変更してみましょう

以下を引ます:

- 1. CompanyLast.xml タクをクリックしてドキュメントをアケティブにします。 グリッドビュー し ることを確認して 代さ い
- 2. メニーオプショから XSL/XQuery | XSLへ移動」を選択します。このエアンドでXML ドキュメントにお参照されている Company.xsl ファイルが開かれます。



- 3. <テーブル border="1" bgcobr="me"> と 行を見つ出し bgcobr="me" をbgcobr="yellow" に変更します。
- 4. メニューオプション プァイル 保存」を選択して、XSL ファイルの変更を保存します。
- 5. CompanyLast.xml ダをクリックして XML ファイルをアクティブにします。 その後 **XSL/XQuery | XSL変換」** を選択するか **F10」** を押します。 新し NXSL Output.html ファイルが XMLSpy のブラウザービュー に表示されます。 テーブルの背景色が黄色になっています。

Your Company						
Name: US dependency Street: Noble Ave City: Dallas State: Texas Zip: 04812						
	First	Last	Ext.	E-Mail	Manager	Degree
	Alfred	Aldrich	33	Aldrich@work.com	false	MA
	Colin	Coletti	444	Coletti@work.com	true	Ph.D
	Fred	Smith	22	Smith@work.com	true	BA

6. メニューオプショから プァイル 保存」を選択し Company.html とい名前でドキュメントを保存します。

8 プロジェクト管理

このセケンシンではXMLSpyのプレジェント管理機能を紹介します。XML ファイルをプレジェントによど管理する利点を学習した後、作成したファイルを簡単なプレジェントで管理することにします。

8.1 プロジェクトの利点

XML ファイルをプロジェクトにより管理すると以下のような利点があります:

- ファイルの拡張子やその他の条件により、複数のファイルやURLをフォルダーにまとめることができます。
- フォレダーおはプロジェクト全体に対してい、チ処理をううとができます。
- DTD おはXML スキーマを指定したフォレダーに割り当て、フォレダー内のファイルに対して検証をううこができます。
- XSLT ファイルを指定したフォレダーに割じ当て、割じ当てたXSLT を使ってそのフォレダー内にあるXML ファイルの変換を行うことができます。
- XSL 変換の出力ファイルの変換先を指定することができます。

上のプロジェクト設定は、全てメニューオプションの プロジェクト | プロバティ」 にで行うことができます。 次のセグションでは、 プロジェクトメニューを使ってプロジェクトを作成します。

- ・ メニューオプショの プロジェクト | ソーズ管理に追加」によりXML ファイルをノーズ管理に置くためで きます 詳レイオインラインマニュアルのソーズ管理セグションを参照してください。
- パーソナルネットワークならいに ウェブフォルダー もプロジェクトに追加し バッチ検証をううとができます。

8.2 プロジェクトの作成

これまでの過程で、メインウンドウカはは様々なファイルが開かれているはずです。これらのファイルをチュートアルプエジェクトの中にまとめることができます。まずは新しいプロジェクトを作成し、その後チュートアルのファイルを対応するプロジェクトのサブ フォルダーに追加していきます。

基本的なプロジェクトの作成

新しいプロジェクトを作成するには:

1. .メニューオプショから 「フロジェクト」新規プロジェクトを作成」を選択します。新しいプロジェクト とう名前の新しいプロジェクトがプロジェクトウンドガン作成されます。新しいプロジェクトはは典型的な XML ファイルのカテューリ フォルダーが作成されます。



- 2. CompanyLast.xml 5/6/19/0 ×12/2/19/0 CompanyLast.xml 7/11/6/7/5-1/12.
- 3. メニューオプショから「クロジェクト」 プロジェクトニアクティクならりに関係するファイルを追加」を選択します。 CompanyLast.xml とAddressLast.xsd の2つのファイルがプロジェクトに追加されます。処理命令で参照されるファイル(例えばXSLT ファイル)は関係するファイルとは見なされないととご注意してください。
- 4. メニューオプショから プロジェクト / プロジェクトを保存」を選択します。プロジェクトをTutorial とり名前 で保存します。

プロジェクトにファイルを追加する

プロジェクトはは他のファイルを追加することもできます:

- 1. CompanyLast.xml 以外の開かれている任意のXML ファイル(.xml の拡張子がつ、たファイル)を選択して、アケティブにします (XML ファイルが他に開かれていたい場合は、XML ファイルを閉め作成してください)。
- 2. メニューオプショから プロジェクト | プロジェクトニアケティブなファイルを追加」を選択します。その.xml ファイルの種類をもとに、XML ファイルがプロジェクトのXML ファイルフォルダーに追加されます。
- 3. 同様に HTML ファイルとXSD ファイル(列えば company.html とAddressFirst.xsd ファイル)をプロ ジェクトコビーレます。 これらのファイルはそれぞれ HTML File フォレダーとDTD/Schemas フォレダーに追加されます。
- 4. メニューオプショの プロジェクト プロジェクトを保存」を選択するか、プロジェクトウィンドンかのファイルはおは フォルダーを選択し、ツールバーの保存アイコン(おさは プァイル | 保存」)をクリックしてプロジェクトを保存します。
- ・ プロジェクトフォレダーを右クリックして「アケティブなファイルを追加」を選択することで、そのフォレダーにアケティブなファイルを追加することができます。

その他の便利なコマンド

その他に主良く使われる以下のプロジェクトコアントがあります:

• 新しいフォレダーをプロジェクトコ保存するは、「プロジェクト」プロジェクトこプロジェクトフォレダーを追加」を

選択して、プロジェクトフォレダーの名前を入力します。

プロジェクトのフォルダー、一部除するこは、フォルダーを右クルクレス、コンテキストメニューから 削除」を選択します。 プロジェクトのファイルを削除するこは、ファイルを右クルクレス Delete」キーを押下します。 •

9 おしまい

こまでわば おめでとう そしてありがとう!

このチュードアルはXMLSpyの基本を学習するのに役立ったでしょか? よ詳し、情報をいりたいときには、オンラインハルを参照するか、XMLSpy アプリケーションフォルダーにあるPDF版のチュードアル(tutorial.pdf)を印刷してく ださい。

Index

<

<%SPY-GEN% でのプロジェクト, その利点,71 作成方法,72 <%SPY-GEN% でのプロジェクト管理,70

X

XML スキーマ, 13 XML ドキュメントを編集中に修正する,64 コンポーネンツの追加,17 チュートリアル,13 デザインビューでのナビゲーション,39 ビューの設定,22 基本的なスキーマの作成,13 新しいファイルの作成,14 名前空間の定義,16 要素を追加する,21 XML スキーマ定義、 高度な機能,27 XML ドキュメント テキストビューにて修正,51 新規に作成,46 妥当性をチェックする,54 XMLドキュメント作成, チュートリアル,45 xsi:type, 使用方法,48 XSL 変換 . XSLT を参照,66 XSLT. <% SPY-GEN% にて修正する,69 XSLT へ変換, XSLT ファイルを割り当てる,67 XSLT 変換, <% PY-GEN% にて,68 チュートリアル,66

Ζ

```
グリッドビュー,58
 データの入力,50
 とテーブルビュー,60
 入力ヘルパを使用する,58
 要素と属性の追加,58
グローバル要素、
 XML スキーマを使用する,34
コンテンツモデル
 基本的なモデルの作成,17
 切り替え属性,36
コンテンツモデルビュー,14
コンポーネント定義
 再利用する,28
コンポジッター
 シーケンスコンポジッター、17
シーケンスコンポジッター
 使用する,17
スキーマ
 XML スキーマを参照,13
 ドキュメンテーション,41
スキーマビュー、
 ビューの設定,22
スキーマ概要 14
データベース /テーブルビュー、
 使用方法 .60
テーブルビュー。
 使用方法,60
テキストビュー
 編集,51
ドキュメンテーション
 スキーマ,41
ナビゲーション
 スキーマデザインでのショートカット,39
拡張グリッドビュー
 グリッドビューを参照,50
型,
 XML ドキュメントでの拡張,48
検証・
 XML ドキュメント,54
出現.
 回数,17
省略可能な要素
 作成,24
詳細入力ヘルパー,17
```

新規 XML ドキュメント, 作成,46 整形式のチェック、 XML ドキュメン Hに対して,54 **属性**,36 コンテンツモデルビューに切り替え,36 スキーマ定義にて,36 単純型, スキーマ定義にて,28 同一性制約, コンテンツモデルビューに切り替え,36 入力ヘルパー、 グリッドビューにて,58 詳細,17 **複合型**,28 スキーマ定義にて,28 定義を拡張する,28 名前空間, スキーマ内の,16 **要素**,24 省略可能にする,24 内容を制限する,24 要素の型, XML ドキュメントで指定する,48

列挙,

属性を定義する,36