

Altova MapForce 2021 Professional Edition ユーザーマニュアル

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

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

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

公開日: 2015-2021

(C) 2015-2021 Altova GmbH

目次

1 はじめに 17 1.1 新規機能 18 1.2 サポートメモ 25 1.3 MapForce とは? 28 1.4 基本概念 33 1.5 ユーザーインターフェイスの概要 35 1.6 規約 44

2 チュートリアル

2.1	XML を新しいスキーマに変換	46
2.2	複数のソースから1つのターゲットにマップ	56
2.3	複数のターゲットスキーマとの作業	62
2.4	ファイルを動的に処理と生成する	70

3 一般的なタスク

79

3.1	マッピン	グとの作業	80
	3.1.1	マッピングにコンポーネントを追加する	80
	3.1.2	URL からコンポーネントを追加する	81
	3.1.3	データストリーミングについて	84
	3.1.4	変換言語の選択	85
	3.1.5	マッピングの検証	86
	3.1.6	マッピング出力の検証	87
	3.1.7	出力のプレビュー	88
	3.1.8	テキストビューの機能	89
	3.1.9	テキストビュー内の検索	93
	3.1.10	XSLT コードのプレビュー	97
	3.1.11	XSLT コードの生成	97
	3.1.12	XQuery ⊐ードのプレビュー	98

	3.1.13	複数のマッピングウィンドウとの作業	
	3.1.14	マッピング設定の変更	
3.2	コンポー	-ネントとの作業	103
	3.2.1	コンポーネント内の検索	104
	3.2.2	コンポーネントの整列	105
	3.2.3	コンポーネント設定を変更する	106
	3.2.4	入力の複製	107
3.3	接続との	の作業	108
	3.3.1	必須の入力について	109
	3.3.2	優先する接続線の表示の変更	111
	3.3.3	接続の注釈	111
	3.3.4	接続設定	112
	3.3.5	一致する子要素の接続	114
	3.3.6	コンテキストメニューの接続	117
	3.3.7	見つからない親接続の通知	118
	3.3.8	接続と子接続の移動	119
	3.3.9	コンポーネント削除後の接続の保持	122
	3.3.10	見つからないアイテムの扱い方	124
3.4	マッピン	・グプロジェクトとの作業	128
	3.4.1	プロジェクトを開く、検索する、閉じる	129
	3.4.2	新規プロジェクトの作成	129
	3.4.3	コード生成設定の設定	131
	3.4.4	プロジェクトフォルダーの管理	132

4 マッピングのデザイン

4.1	相対と維	対パスの使用	136
	4.1.1	コンポーネント上で相対パスを使用する	136
	4.1.2	ファイルベースのデータベースへパスを設定する	138
	4.1.3	破損したパスの参照を修正する	140
	4.1.4	多種の実行環境内のパス	141
	4.1.5	コピーと貼り付けと相対パス	142
4.2	マッピング	ゲ接続の種類	143
	4.2.1	ターゲット優先マッピング	143
	4.2.2	ソース優先マッピング	143

	4.2.3	全てコピー接続	151
4.3	チェーン	ママッピング	153
	4.3.1	例: パススルーが有効な場合	154
	4.3.2	例: パススルーが無効な場合	158
4.4	複数の	入力または出力ファイルを動的に処理	161
	4.4.1	複数の入力ファイルを単一の出力ファイルにマップする	163
	4.4.2	複数の入力ファイルを複数の出力ファイルにマッピングする	165
	4.4.3	ファイル名をマッピングパラメーターとして提供する	165
	4.4.4	複数の出力ファイルをプレビューする	166
	4.4.5	例: 1つの XML ファイルを複数のファイルに分割	166
	4.4.6	例: データベーステーブルに対して複数の XML を生成	168
4.5	マッピン	・グにパラメーターを与える	170
	4.5.1	単純型入力コンポーネントの追加	171
	4.5.2	単純型入力コンポーネント設定	172
	4.5.3	デフォルトの入力値を作成する	174
	4.5.4	例: ファイル名をマッピングパラメーターとして使用する	175
4.6	マッピン	・グから文字列の値を返す	179
	4.6.1	単純型出力コンポーネントの追加	180
	4.6.2	例: 関数出力のプレビュー	181
4.7	変数の	使用	183
	4.7.1	変数の追加	184
	4.7.2	変数のコンテキストとスコープの変更	188
	4.7.3	例: データベースのテーブルの行の計算	190
	4.7.4	例: ノードのフィルターと番号付け	191
	4.7.5	例:記録のグループ化、および、サブグループ化	192
4.8	データの	D並べ替え	194
	4.8.1	複数のキーを使用した並べ替え	196
	4.8.2	変数を使用して並べ替える	197
4.9	データの	Dグループ分け	199
	4.9.1	例: キー別にレコードをグループ分けする方法	202
4.10	フィルタ	一と条件	205
	4.10.1	例: ノードのフィルター	206
	4.10.2	例: 条件付で値を返す	208
4.11	データの	Dジョイン	211
	4.11.1	ジョインの条件の追加	213

	4.11.2	3つ以上の構造をジョインする	. 216
	4.11.3	例: XML 構造のジョイン	. 217
4.12	Value-M	lap の使用	. 223
	4.12.1	例: 曜日の置換	. 226
	4.12.2	例: 職位の置換	. 228
4.13	例外のi	追加	. 232
	4.13.1	例: "より大きい" 条件の例外	. 232
	4.13.2	例: ノードが存在しない場合の例外	. 233
4.14	文字列(の解析とシリアル化	. 236
	4.14.1	コンポーネントの解析/シリアル化について	. 236
	4.14.2	例: 文字列にシリアル化 (XML からデータベースへ)	. 238
4.15	ノード名	のマッピング	. 243
	4.15.1	ノード名へのアクセスを取得する	. 244
	4.15.2	特定の型のノードへのアクセス	. 251
	4.15.3	例: 要素名を属性値にマップする	. 255
	4.15.4	例:名前別にノードをグループ化またはフィルターする	. 259
4.16	StyleVis	ion を使用してマッピングの出力のスタイルを指定する方法	. 264
	4.16.1	StyleVision スタイルシートを使用したマッピングの例	. 265
4.17	マッピン	グドキュメンテーションの生成	. 267
	4.17.1	定義済みの StyleVision パワースタイルシート	. 268
	4.17.2	カスタムスタイルシート	. 273
4.18	マッピン	グのルールと戦略	. 275
	4.18.1	シーケンス	. 276
	4.18.2	マッピングコンテキスト	. 277
	4.18.3	優先コンテキスト	. 286
	4.18.4	複数のターゲットコンポーネント	. 291

5	マッピングのデバッグ	294
5.1	デバッガーの準備	297
5.2	デバッガーコマンド	298
5.3	デバッグモードについて	
5.4	ブレークポイントの追加と削除	
5.5	値ウィンドウの使用	
5.6	コンテキストウィンドウの使用	

5.7	ブレークポイントウィンドウの使用	309
5.8	部分的に生成された出力のプレビュー	311
5.9	コネクタの現在の値を表示する	312
5.10	最近の過去にステップバックする	313
5.11	コネクタにより処理された値の履歴を確認する	314
5.12	コンテキストを値に設定する	315
5.13	デバッガー設定	316

6 データソースとターゲット

6.1	XML と	XML スキーマ	318
	6.1.1	XML スキーマの生成	318
	6.1.2	XML コンポーネント設定	319
	6.1.3	"スキーマ" コンポーネントとしてを DTD 使用する	323
	6.1.4	派生した XML スキーマの型	323
	6.1.5	QNames	326
	6.1.6	Nil の値 / Nillable	327
	6.1.7	コメントと処理命令	329
	6.1.8	CDATA セクション	330
	6.1.9	ワイルドカード - xs:any / xs:anyAttribute	332
	6.1.10	複数のスキーマからのデータのマージ	337
	6.1.11	カスタム名前空間の宣言	338
6.2	MapFor	ceとデータベース	341
	6.2.1	データソースへの接続	343
	6.2.2	データベースマッピングの概要	427
	6.2.3	データをデータベースにマップする	458
	6.2.4	データベースのデータのジョイン	508
	6.2.5	データベースデータのフィルターと並べ替え (SQL WHERE/ORDER)	524
	6.2.6	仮想テーブルとしての SQL SELECT ステートメント	530
	6.2.7	XML データから / データベースフィールドへのマッピング	538
	6.2.8	データベースの参照とクエリ	549
	6.2.9	ストアドプロシージャ	566
	6.2.10	トランザクションロールバックの処理	591
	6.2.11	データベースのトレースとエラーのログ	601
6.3	ح csv	テキストファイル	610

	6.3.1	例: CSV ファイルから XML へのマッピング	610
	6.3.2	例: アイテムの反復	612
	6.3.3	例: CSV と FLF ファイルから階層を作成する	614
	6.3.4	CSV オプションの設定	617
	6.3.5	例: 固定長テキストファイルを(データベースへ)マッピング	621
	6.3.6	FLF オプションの設定	627
6.4	バイナリ	ファイル	633
	6.4.1	例: バイナリファイルの読み取り	634
	6.4.2	例: バイナリファイルの書き込み	636

7 関数

7.1	使用方法	៩	639
	7.1.1	マッピングにビルトイン関数を追加する	639
	7.1.2	マッピングに定数を追加する	641
	7.1.3	関数の検索	642
	7.1.4	関数の型と詳細を確認する	643
	7.1.5	関数引数の追加、または、削除	644
7.2	デフォル	トとノード関数	645
	7.2.1	デフォルトとノード関数の作成方法	645
	7.2.2	既存のルールの編集と削除	649
	7.2.3	デフォルトとノード関数のしくみ	650
	7.2.4	ノード関数とデフォルトを条件付きで適用する	656
	7.2.5	ノードメタデータをノード関数に提供する	661
	7.2.6	例: 空の CSV フィールドの置換	664
7.3	ューザー	-定義関数	669
	7.3.1	ユーザー定義関数の作成	671
	7.3.2	ユーザー定義関数内のパラメーター	674
	7.3.3	インラインと正規ユーザー定義関数	679
	7.3.4	ユーザー定義関数のナビゲート	681
	7.3.5	ユーザー定義関数の編集	681
	7.3.6	ユーザー定義関数の削除	682
	7.3.7	ユーザー定義関数の呼び出しとインポート	683
	7.3.8	マッピング間で UDF をコピーし張り付ける用法	684
	7.3.9	例: ルックアップと連結	685

	7.3.10	例:再帰的な検索	. 689
7.4	カスタム	、XSLT 1.0 または 2.0 関数のインポート	. 693
	7.4.1	例: カスタム XSLT 関数の追加	. 694
	7.4.2	例: ノードの値の集計	. 697
7.5	カスタム	、XQuery 1.0 関数のインポート	. 700
	7.5.1	例: カスタム XQuery 関数のインポート	. 701
7.6	カスタム	、Java と .NET ライブラリのインポート	. 705
	7.6.1	例: カスタム Java クラスのインポート	. 708
	7.6.2	例: カスタム .NET DLL アセンブリのインポート	. 710
7.7	Java、C	# および C++ ライブラリを手動で参照する	. 712
	7.7.1	.mff ファイルの構成	. 712
	7.7.2	.mff ファイルを MapForce にインポートする	. 717
	7.7.3	データ型のマッピング	. 718
	7.7.4	例: C# を .mff に適用する方法	. 720
	7.7.5	例: C++ を.mff に適用する方法	. 721
	7.7.6	例: Java を .mff に適用する方法	. 723
7.8	関数ライ	イブラリの管理	. 726
	7.8.1	ローカルとグローバルライブラリ	. 728
	7.8.2	相対的なライブラリパス	. 729
7.9	正規表理	見	. 731
7.10	関数ライ	(ブラリレファレンス	. 734
	7.10.1	core aggregate functions (集計関数)	. 751
	7.10.2	core conversion functions (変換関数)	. 757
	7.10.3	core file path functions (ファイルパス関数)	. 772
	7.10.4	core generator functions (ジェネレーター関数)	. 776
	7.10.5	core logical functions (論理関数)	. 778
	7.10.6	core math functions (数学関数)	. 784
	7.10.7	core node functions (ノード関数)	. 789
	7.10.8	core QName functions (QName 関数)	. 794
	7.10.9	core sequence functions (シーケンス関数)	. 796
	7.10.10	core string functions (文字列関数)	. 820
	7.10.11	db	. 833
	7.10.12	lang datetime functions (日付時刻関数)	. 836
	7.10.13	lang file functions (ファイル関数)	. 855
	7.10.14	lang generator functions (ジェネレーター関数)	. 857

7.10.15	lang logical functions (論理関数) 8	357
7.10.16	lang math functions (数学関数)	359
7.10.17	lang QName functions (QName 関数)	368
7.10.18	lang string functions (文字列関数) 8	369
7.10.19	xpath2 accessors	383
7.10.20	xpath2 anyURI functions (anyURI 関数) 8	385
7.10.21	xpath2 boolean functions (ブール値関数) 8	386
7.10.22	xpath2 constructors	386
7.10.23	xpath2 context functions (コンテキスト関数) 8	388
7.10.24	xpath2 durations, date and time functions (期間、日付、および時刻関数)	390
7.10.25	xpath2 node functions (ノード関数)	905
7.10.26	xpath2 numeric functions (数值関数)	912
7.10.27	xpath2 string functions (文字列関数)	ə13
7.10.28	xpath3 external information functions	9 23
7.10.29	xpath3 formatting functions) 25
7.10.30	xpath3 math functions	929
7.10.31	xpath3 URI functions	935
7.10.32	xslt xpath functions (xpath 関数)	3 37
7.10.33	xslt xslt functions (xslt 関数)	9 40

8 マッピングの自動化と MapForce

8.1	RaptorXML Server を使用して自動化する	945
8.2	MapForce Server を使用して自動化する	946
8.3	サーバーでの実行のためにマッピングを準備する	947
8.4	MapForce Server 実行可能ファイルにマッピングをコンパイルする	952
8.5	マッピングを FlowForce Server にデプロイする	955
8.6	MapForce コマンドラインインターフェイス	959

9 Altova グローバルリソース

9.1	グローバルリソースの作成	965
9.2	グローバルリソースとしてのデータベース	966
9.3	グローバルリソースとしての MapForce と StyleVision 変換結果	970
9.4	グローバルリソース XML ファイル	971

10

944

9.5 9.5.1 9.5.2 9.5.3 9.5.4 9.5.5 9.6 9.7 9.8 9.9

10 MapForce のカスタマイズ

10.1	MapFor	ce オプションの変更	989
	10.1.1	Java 設定	990
	10.1.2	ネットワークプロキシ設定	991
10.2	キーボー	ードのショートカット	994
	10.2.1	ショートカットのカスタム化	995
10.3	メニュー	をカスタム化する方法	997
10.4	カタログ	゙゚ファイル	999

11 Visual Studio のための MapForce プラグイン 1003

11.1プラグインの有効化	1004
11.2マッピングとプロジェクトと作業	1006
11.3 …共通メニューと関数へのアクセス	1007

12 Eclipse のための MapForce プラグイン 1009

T2.1Eclipse 072400 MapForce 7 9912012XF-70	
12.2MapForce パースペクティブ 10	014
12.3共通メニューと関数へのアクセス 10	017
12.4マッピングおよびプロジェクトと作業1(020
12.4.1 MapForce/Eclipse プロジェクトの作成10	020
12.4.2 新しいマッピングの作成10	022
12.4.3 Eclipse プロジェクトに既存のマッピングをインポートする1(024

989

	12.4.4	自動ビルドの構成と MapForce コードの生成	1027
12.5 .	Eclipse	のための MapForce プラグインの有効化	1030

13 メニューレファレンス

1031

13.1ファイル	1032
13.2編集	1035
13.3挿入	1036
13.4プロジェクト	1038
13.5 …コンポーネント	1039
13.6接続	1041
13.7関数	1042
13.8出力	1043
13.9デバッグ	1045
13.10表示	1046
13.11ツール	1048
13.12ウィンドウ	1049
13.13ヘルプメニュー	1050

14 Code Generator

14.1	4.1What's new		
14.2Generating C++ code			. 1059
	14.2.1	Generating code from a mapping	. 1060
	14.2.2	Generating code from a mapping project	. 1061
	14.2.3	Building the project	. 1061
	14.2.4	Running the application	. 1062
14.3Generating C# code			. 1064
	14.3.1	Selecting the target platform	. 1065
	14.3.2	Generating code from a mapping	. 1066
	14.3.3	Generating code from a mapping project	. 1067
	14.3.4	Building .NET Framework projects	. 1068
	14.3.5	Building .NET 5 and .NET Core projects	. 1069
14.4	Generati	ing Java code	. 1071
	14.4.1	Generating code from a mapping	. 1072

14.4.2	Generating code from a mapping project	. 1073
14.4.3	Building the project with Ant	. 1073
14.4.4	Handling JDBC references	. 1075
14.4.5	Example: Generate and run Java code	. 1076
14.5Integrati	ng MapForce-Generated Code	. 1085
14.5.1	Java example	. 1086
14.5.2	C# example	. 1088
14.5.3	C++ example	. 1090
14.5.4	Changing Input and Output Programmatically	. 1091
14.6Generat	ing Code from XML Schemas or DTDs	. 1097
14.6.1	About Schema Wrapper Libraries (C++)	. 1099
14.6.2	About Schema Wrapper Libraries (C#)	. 1102
14.6.3	About Schema Wrapper Libraries (Java)	. 1104
14.6.4	Integrating Schema Wrapper Libraries	. 1106
14.6.5	Example: Book Library	. 1109
14.6.6	Example: Purchase Order	. 1131
14.7Referend	ce to Generated Classes (C++)	. 1138
14.7.1	altova::DateTime	. 1138
14.7.2	altova::Duration	. 1141
14.7.3	altova::DayTimeDuration	. 1143
14.7.4	altova::YearMonthDuration	. 1144
14.7.5	altova::meta::Attribute	. 1144
14.7.6	altova::meta::ComplexType	. 1145
14.7.7	altova::meta::Element	. 1146
14.7.8	altova::meta::SimpleType	. 1146
14.7.9	[YourSchema]::[CDoc]	. 1147
14.7.10	[YourSchema]::[ElementType]	. 1150
14.7.11	[YourSchema]::MemberAttribute	. 1151
14.7.12	[YourSchema]::MemberElement	. 1151
14.8Referend	ce to Generated Classes (C#)	. 1153
14.8.1	Altova.Types.DateTime	. 1153
14.8.2	Altova.Types.DateTimeFormat	. 1156
14.8.3	Altova.Types.Duration	. 1158
14.8.4	Altova.Xml.Meta.Attribute	. 1160
14.8.5	Altova.Xml.Meta.ComplexType	. 1161

14

15

The MapForce API

14.8.6	Altova.Xml.Meta.Element	
14.8.7	Altova.Xml.Meta.SimpleType	
14.8.8	[YourSchema].[Doc]	
14.8.9	 [YourSchema].[ElementType]	1165
14.8.10	[YourSchemaType].MemberAttribute	1166
14.8.11	[YourSchemaType].MemberElement	1166
14.9Referen	ce to Generated Classes (Java)	1168
14.9.1	com.altova.types.DateTime	1168
14.9.2	com.altova.types.Duration	1172
14.9.3	com.altova.xml.meta.Attribute	1175
14.9.4	com.altova.xml.meta.ComplexType	1176
14.9.5	com.altova.xml.meta.Element	1176
14.9.6	com.altova.xml.meta.SimpleType	1177
14.9.7	com.[YourSchema].[Doc]	1178
14.9.8	com.[YourSchema].[ElementType]	1179
14.9.9	com.[YourSchema].[YourSchemaType].MemberAttribute	1181
14.9.10	com.[YourSchema].[YourSchemaType].MemberElement	1181
14.10Code Ge	eneration Tips	1183
14.11Code Ge	enerator Options	1184
14.12SPL Ref	erence	1186
14.12.1	Basic SPL structure	1186
14.12.2	Declarations	1187
14.12.3	Variables	1189
14.12.4	Predefined variables	1190
14.12.5	Creating output files	1191
14.12.6	Operators	1193
14.12.7	Conditions	1193
14.12.8	Collections and foreach	1194
14.12.9	Subroutines	1195
14.12.10	Built in Types	

15.1 ...Accessing the API...... 1204

15.3Error Ha	ndling	1208
15.4Example	C# Project	1210
15.5Example	Java Project	1214
15.6JScript	Examples	1217
15.6.1	Start Application	1217
15.6.2	Simple Document Access	1218
15.6.3	Generate Code	1219
15.6.4	Generate Code (Alternative)	1221
15.6.5	Run a Mapping	1222
15.6.6	Project Tasks	1226
15.7Object F	Reference	1230
15.7.1	Interfaces	1230
15.7.2	Enumerations	1370

16 ActiveX Integration

16.1必要	要条件		
16.2Acti	iveX コント	ロールをツールボックスに追加する…	
16.3 …アフ	゚リケーショ	シレベルの統合	
16.4 …ドキ	・ユメントレ	ベルの統合	
16.5Act	iveX 統合	のサンプル	
16.5	5.1 C#		
16.5	5.2 HTM	L	
16.5	5.3 Java		
16.5	5.4 VB.N	ET	
16.6コマ	ンド レファ	·レンス	
16.6	3.1 「ファ	イル」メニュー	
16.6	6.2 「編∮	€」メニュー	
16.6	3.3 「挿)	し」メニュー	
16.6	6.4 「プロ	↓ジェクト」メニュー	
16.6	8.5 「コン	[,] ポーネント」メニュー	
16.6	6.6 「接絲	売」メニュー	
16.6	6.7 「関数	友」メニュ―	
16.6	5.8 「出っ	<u></u>	
16.6	8.9 「ディ	、ッグ」メニュー	1419

16.7.1 16.7.2 16.7.3 16.7.4 16.7.5 16.7.6 列举......1441

17 付録

1443

17.1エン:	ジン情報	1444
17.1.	1 XSLT および XQuery エンジンに関する情報	1444
17.1.	2 XSLTとXPath/XQuery 関数	1448
17.2技術	データ	1538
17.2.	1 OS とメモリ要件	1538
17.2.	2 Altova XML バリデーター	1538
17.2.	3 Altova XSLTとXQuery エンジン	1538
17.2.4	4 Unicode のサポート	1538
17.2.	5 インターネットの使用	
17.3ライト	2ンス情報	1540
17.3.	1 電子的なソフトウェアの配布	1540
17.3.	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	1540
17.3.	3 エンドユーザー使用許諾契約書	

18 用語

1542

インデックス

1 はじめに

MapForce® 2021 Professional Edition は高度なデーダ統合プロジェクトにで決定的な役割を果たす視覚的なデータマピングソ ールです。MapForce® は、プラナフォーム更新済みのW indows 7 SP1、W indows 8、W indows 10 とプラオフォーム更新済 みのW indows Server 2008 R2 SP1 おけお以降上で作動する32/64-bit W indows アプレケーションです。64 ビホサポート は、Enterprise とProfessional エディンヨンでご利用にサイズことができます。MapForce は Visual Studio とEclipse、および Microsoft Office 製品と統合することもできます。サポート大を参照してくたさい。



最終更新日: 2021 年 02 月 24 日

1.1 新規機能

MapForce 2021 リース2 の新規機能

- C#コードを生成する際.NET Core 3.1 と.NET 5.0 をコード生成オプションからテーゲットフレームワークとして選択することができます(.NET Framework プロジェクトに既存のサポートを追加します)。詳細に関してはC#コードの生成を参照してくたさい。
- XSLT 3.0 はマピング言語とてサポートされるようごなりました。XSLT コードの生成 を参照してくたさい。また、MapForce にはマピング言語がXSLT 3.0 である場合サポートされる新規のビルトイン関数が搭載されています。

 <u>関数ライブラルファレン</u>
 <u>ス</u>を参照してくたさい。
- 内部のアップデートと最適化。

MapForce 2021 の新規機能:

- MapForce マピングはBLOB (バイナリラージオブシェクト)データをバイナリファイルから読み取り、バイナリファイルをディスクに書き込むことができます。これは、例えば、BLOB フィールドをデータベースから読み取り、ディスクトでイメージファイルとして保存すること、封まは、PDFなどの、イナリファイルをディスクから読み取りXML ファイル内のxs:base64Binary フィールドとして保存することが可能にないます。詳細に関してはシィナリファイルを参照してくたさい。
- 新規のデータベース デンション かりポート されています: Maria DB 10.4、10.5
- 新規のEclipse バージョンがサポートされて、ます: 2019.09, 2019.12, 2020.03, 2020.06
- マピング内でSQL ジョインエンポーネトを使用して複数のデータベーステーブルをジョインおけまビューする場合、LEFT OUTER JOIN おけまINNER JOIN としてジョインモードを設定できます。ジョインモードの変更を参照してくたさい。
- 内部のアップデートと最適化

2020 Release 2 の新規機能

- 新規のライブラル管理ウイボウニンドキュメトおけコプログラムレベルでインポートされたすべての関数ライブラを表示管理できるようプロジョルた(これ)はMapForceユーザー定義関数と他のライブラル含まれています)。例えば1つのアメピングから他のアメピングに簡単にユーザー定義関数をコピーして貼い灯けることができます。マメピング間でUDFをコピーし張い灯ける用法を参照してください。
- マッピングファイルがライブラ たんポートする際に、インポートされたライブラ ファイルの やはデフォルトで相対パ やです。相対的な <u>ライブラル や</u>を参照してくたさい。以前のリース同様アプリケーションレベルでマッピングファイルをインポートすることは可能ですが、この場合ライブラルの やは絶対パ やです。
- マッピングファイルがXSLT おはXQuery ライブラをインポートする場合、相対的ないなを使用してインポートされたライブラを 参照するXSLT おはXQuery コードを生成することができます。新規のオプションはマッピング設定 ダイアログボックス内で使 用することができます。
- MapForce API はインポートされたライブラレビアング的に管理できる機能には拡張されました(例えば、ライブラルの追加ませば削除機能など)。
- XML スキーマラッパーライブラルのオーダに生成されたコードは要素名前空間とプレフィックスを管理する機能を提供します。例 Purchase Order を参照してください。
- 新規のデータベースバージョンがサポートされます: PostgreSQL 12.1 とInformix 14.10.
- 内部のアップデートと最適化。

MapForce リース2020 の新規機能

- <u>Visual Studio のためのMap Force プラグインとコード生成</u>内のVisual Studio 2019 へのサポト。
- Eclipse 4.9 4.12のためのサポト。EclipseのためのMapForceプラグインを参照してくたさい
- Oracle パッケージン ジー・クスト アドプロシージャ、おけま関数が含まれる場合、これらはマッピングでも使用することができます。 マッピングへストアドプロシージャを追加する方法を参照してくけざい。
- データベースオブジェクト名が特定のスキーマではよくデフォルトのスキーマニ対して相対的に扱われるようにデータベースエレポーネントを構成することができます。この機能により異なるデータベース間を切り替える時間を節約することができます。データベースとスキー <u>オ間の切り替え</u>を参照してくたさい。

- はじめに
 - MapForce内で作成されたグロー・シリノースをFlowForce Serverにデプロイすることができます。FlowForce Serverに リノースをデプロイを参照してください。
 - ルッグアップテーブルを使用して値を置き換える場合、テーブルデータ(キー値ペア)をCSV おけまExcel などの外部ソースからマメピングに張り付けることができます。おこ、代替の使用を必要としたい値や不足する関数など定義済みのルッグアップテーブル内で値が見つからない場合、このような値の処理がsubstitute-missing 関数の使用を必要としたいたが扱いが簡単になります。
 - 内部のアップデートと最適化

MapForce 2019 リース3 の新規機能

- <u>データベーストレースとエラーのログ</u>のためのサポート
- 生成されたプログラムコード内およびサーバー実行内でのデータベーストランザグションの処理のオンダのサポート。マメピングがデータベースを更新する際にエラーが発生すると、操作の一部が失敗しても、マメピングの実行を継続することができます。
 データベース例 外の扱いを参照してください。
- 新規の<u>charset-encode</u> と<u>charset-decode</u> 関数を使用して、マピングはBase64 データをエンコード、おさまデコード することができます。
- 新規の出エータベース・ディンがサポートされます: PostgreSQL 11、Db2 for i 7.2 および7.3
- Eclipse 4.7-4.10 へのサポト。Eclipse のかのMapForce プラグインを参照してたさい。
- MapForce から Java 仮想マシンを明示的に設定するためのサポート。Java 設定を参照してくたさい。
- 内部のアップデートと最適化

MapForce リース2019 の新規機能

- 新規のデータベースがサポートされるよう」なりました Firebird 3.0、Informix 12.10、MariaDB 10.2、MySQL 8.0、 PostgreSQL 10、SQL Server 2017、Linux のよののSQL Server(前のデンタンにサポートが追加されました)。
- カスタム定義の条件を満たす固有のノードレード関数まけはデフォルトを選択して適用できます。例えば正規表現に一致する名前を持つ特定の型のノードの升・ノード関数を適用することができます。を参照してくたさい。ノード関数とデフォルトを条件付きで適用する。
- ノード関数を作成する際、ノード名、おゴはノード注釈などをベースコンてデータを条件付きて処理することができます。ノードメタ データをノード関数に提供を参照してください。
- グーループ化関数を含むマピングは生成されたC#、C++、おけよJavaコード内で更に素早く実行することができます。
- 内部の更新と最適化。

MapForce 2018 リース2 の新規機能

- マピングの空のエリアをダブルクトックしてビルトイン関数、ユーザー定義関数、および、制約を簡単にママピングに追加できるよう になりまた(次を参照してください、マピングバビルトイン関数を追加するとマピングに定数を追加する)。
- 次のデーダベースかりポートされるようゴンリまた MariaDB 10、Teradata 16
- ビルトーン関数と定数をマピングの空のエアをクリックして便利に追加できるようゴンはした。
- 内部のアップデートと最適化

MapForce リース2018 の新規機能

- Visual Studio 2013、2015、と2017 のためのプログラムコード生成をサポートします。次を参照してくたさい、ユードジェネレ <u>ーター</u>
- 以下のデータベースバージョンをサポートします: Sybase ASE 16, PostgreSQL 9.6, MySQL 5.7
- 内部のアップデートと最適化

MapForce 2017 リース3 の新規機能

- 与えられた条件に基づき、2つの異なる構造からのデータをジョインするようは、使用する新規のエンポーネント型(ジョイン)が搭載されるようはいました(次を参照: データのジョイン)。マビングがデータをデータベースから読み取る時、データベーステーブルをジョインすることができ、また、SQL JOIN モードでビューすることができます。次を参照してくたさい、データベースデータのジョイン(SQL JOIN)。
- 「出力」ペイン、「XQuery」ペイン、および「XSLT」ペイン内でのテキストの検索オプション。か強化されました次を参照: <u>テキストビュー内の検索</u>)。また、テキストのイライトも上記のペインで使用することができます(次を参照:<u>テキストのイライ</u> ト)。
- データベースを更新するマンピングが、NULL対応しデータを比較するため、任意で構成できるようこなりました。NULL対応する比較は、null値を含むデータを(各特殊のデータベース)に合わせて)よりよく扱う方法を提供します(次を参照:データベーステーブルアグション内でNullを扱う)。
- MapForce ActiveX コトロール内で、MapForceCommand オブジェクの構造がコマドの一意の名前を取得するために 使用することのできる新規のName プロ・ティを含むようご強化されまた。これにより、MapForce コマドのプログラング上の 情報の取得か簡素化されまた(次を参照: コマンド情報の抽出)。
- 内部のアップデートと最適化

MapForce リース2017 の新規機能

- マソングは、ネイティブな接続を使用してPostgreSQLデータベースは接続することができます(次を参照してください) PostgreSQL 接続のセルアップ)。
- マピングは、ADO.NET プロ・イダを使用して SQL Server と他のデータベース型に接続することができます。(次を参照してく たさい、ADO.NET 接続のセルアップ)
- 新し、データベース型へのサポート: Progress OpenEdge。次を参照してけ込い Progress OpenEdge (ODBC) へ接続する。
- JDBC を使用してデータベースは接続する場合、jar ライブラルの検索/ ひをデータベース接続ダイアログで直接指定することが できます(次を参照してください) JDBC 接続のセットアップ)
- 「Update if... Insert Rest」アクションを使用したマンピングによりデータベースがアップデートされる場合、MERGE ステートメントが選択されたデータベースのために作成されます(次を参照してください) MERGE ステートメント)。
- MapForce Server 実行ファイル(.mfx)を、特定 MapForce Server の デションのためにエント ペルオることができます(次を参照してびさい、特定のMapForce Server バージョンのためにマンと グをコント ペルオる)
- ソースXML からノード名を読み取る おさよ、CSV/テキスト ファイルからフィールド名)を読み取ることは可能でターゲルトにつの 情報をマップすることができます。また、動的に新規のXML 属性、まさは要素(まさよ、CSV/Text フィールド名)をノースから与 えられた値をベースコンナターゲル内に作成することもできます。ノード名のマンピングを参照してくたさい。
- 要素レベルでカスタム化された名前空間を持つXML インスタンスファイルを作成することができます(次を参照してくたさい) カスタム
 <u>カスタム</u>
 <u>カスタム</u>
- 内部のアップデートと最適化

MapForce 2016 R2 の新規機能

- <u>XSLT ペイン</u>内の更に直感的なコードの使用:折げたまれたテキストが、省略シンボルと共に表示され、ツールヒトとして、プレビューすることができます。XQuery ペインとSQL エディター内のテキストにも同じルールが適用されます
- アクティブなマッピング内で関数が使用される全ての箇所を検索することができます(ライブラリウィボウ内の関数を右クトックし、「全ての呼び出しを検索する」を選択します)。
- 内部のアップデートと最適化

MapForce リース2016 の新規機能

- XSLT 1.0 コードの改善された生成(生成されたスタイルシートは読み込みやすく、より早く実行することができます)。
- 新規集計関数がMapForce コアライブラルで使用することができます: min-string とmax-string。これらの関数は文字列のシーケンスから最大値および最小値を取得することができます。
- 内蔵の実行エンジンのためにかかれたマッピングはデバックすることができます。(マッピングのデバッグを参照してくたさい)。
- <u>Visual Studio のためのMapForce プラグイン</u>はVisual Studio 2015(前の、ジョンへのサポトに追加してをサポトします。
- (前の デジョンへのサポトに追加して)新しいデータベースバージョンがサポートされます: SQL Server 2014、Oracle 12c、IBM DB2 10.5、PostgreSQL 9.4、MySQL 5.6。
- Firebird データベースはサポートされて、ます。(<u>Firebird(ODBC)への接続</u>を参照してくだ さい))

MapForce バージョン 2015 R4 の新規機能

- Eclipse のためのMapForce プラグイン。MapForce ファイル特有のコマナドは新規 MapForce メニューで使用することが できます。(<u>共通メニューと関数へのアクセス</u>を参照してくたさい)。
- 内部的な更新と最適化

MapForce バージョン 2015 R3 の新規機能

- XML 出力内の<?xml ... ?>宣言を抑制するオプション
- (EDI CSV、固定値フィールド、JSON、XMLを含むテキストベースのエレポーネトは、プレーンファイルコルえ、文字列を 解析およびシノアル化することができます。
- SQLite データベースへのサポート(SQLite 接続のセナアップを参照してくたさい)。
- 新規文字列・ディング関数: pad-string-left とpad-string-right
- 新しロンポーネト型 単純型出力
- 内部的な更新と最適化

MapForce リース2015 の新規機能

- 新規 language 引数は <u>format-date</u> と <u>format-dateTime</u> 関数で使用することができます
- 新規シーケンス関数: <u>replicate-item</u>

MapForce バージョン 2014 R2 の新規機能

- 新規 <u>シーケンス 関数</u>: シーケンスの生成、item-at など。
- 出力コンポーネント内のCDATA セクションを定義する機能
- <u>データベースの実行</u>のためのタイムアナの値を定義する機能
- コンポーネントの削除後接続を保存する方法
- データベースデータの<u>いしか転送</u>(全てをいしか挿入)
- ターゲオコポーネト内の必須アイテムの自動ハイライト

MapForce リース2014 の新規機能

- RaptorXML 検証への統合と XML Schema 1.1 への基本的なサポート
- 新規 RaptorXML XSLT とXQuery エンシの統合
- <u>XML スキーマブイルドカードへのサポート</u>、xs:anyとxs:anyAttribute
- XML ターゲオコンポーネント内のコントと処理命令へのサポート

- <u>年齢</u> 関数
- CSV データのナンタ<u>引用文字</u>を挿入

MapForce バージョン 2013 R2 SP1 の新規機能

• 高速変換エンジン <u>Raptor XML Server</u>

MapForce バージョン 2013 R2 の新規機能

- <u>MapForce Server</u> のザポト。
- コマド ラインやファイルメニューから<u>MapForce Server 実行</u> ファイルを生成することで、MapForce Server にて実行することができます。
- MapForce のマピングをFlowForce Server ヘデルイ することができます。
- Informix 11.7 データベースのサポートと、その他のデータベースのサポートの拡張。
- 出力ファイルで使用される行素のユーザー定義。ユーザー定義。
- 内部的な更新と最適化

MapForce リース2013 の新規機能

- マピングにて<u>ストアドプロシージャの</u>呼び出し
- データベース関数のサポート (ストアドプロシージャイン類似した機能)
- パラメーターを伴う<u>SELECT ステートメント</u>のサポート
- 内部的な更新と最適化

MapForce バージョン 2012 R2 の新規機能

- XSLT 2.0、XQuery、内蔵の実行エンジノこて ソートコンポーネント をサポート
- ユーザー定義ユンポーネト名
- SQL-W HERE 機能の拡張 ORDER BY
- IBM iSeries のデータベースコン対応した論理ファイルをサポート:論理ファイルがビューとて表示されます
- IBM DB2 論理ファイルのサポト: DB2 データベースのIBM iSeries エディション 搭載された論理ファイルにより、1 つまた は複数の物理ファイルが表されます。論理ファイルによりシーケンシャルなかたちでデータへアクセスすることができるほか、物理ファイル とは異なるフォーマナを行うことができます。IBM iSeries エレニーターへ接続すると、論理ファイルにより構成された既存のデータ ベースを扱うことしてる場合もあります。このようなデータへはこれまでアクセスすることができませんでしたが、バージョン 2012 リレース 2 からよそれが可能にないました。

MapForce リース2012 の新規機能

- XML、CSV、FLF ファイルコマオしてデータストリーミングを内蔵の実行エンジノこで処理
- 新たなデータベースエンジノによる ODBC ならびに JDBC 接続のサポート
- ウィボウマピングウィボウコおおコレポーネトの自動配置
- 新し 関数: parse-date と parse-time
- プロジェクトタブ/ウィンドウからのアイテムの検索
- <u>ターゲナ親</u>ノードへの接続を通知
- マピングにてコンポーネトのシーケンスを制御するルール

MapForce バージョン 2011R3 の新規機能

- <u>中間変数.NET Framework 4.0</u> アセンブリファイルのサポート
- コマボライから<u>StyleVision</u>形式でフォーマナされナギュメトを生成

MapForce バージョン 2011R2 の新規機能

- ビルイ実行エンジィンるストリーム出力サポート
- ライブラリウィボウライブラリウィボウニお1る検索機能
- 反転マピング
- 拡張可能な<u>IF-ELSE</u> 関数
- Core ライブラルは1るノード名ならびに解析関数キー生成設定の統合により改善されたデータベーステーブルアグノョンダイアログ
- マピングの<u>ドキュメンテーション生成</u>を行う際にStyleVision Power Stylesheetを使用するオプノョンを追加

MapForce リース2011 の新規機能

- <u>マピングチェーン</u>にて、中間コンポーネトのプレビュー(パススループレビュー)
- サポーされている全ての言語において dateTime とnumbers のフォーマナ機能が利用可能に
- <u>自動連番</u>機能の強化
- <u>remove-timezone</u> と<u>convert-to-utc</u> のタイムゾーメコ関する関数を新たら追加
- StyleVision チャートを含む<u>StyleVision</u> Power Stylesheets を使用するターゲオコンポーネトのプレビュー

MapForce バージョン 2010 リノース3 の新規機能

- C# ならびにC++ におて Visual Studio 2010 プロジェクトファイルの生成をサポート
- 生成されたC++コードにCMSXML 6.0をサポート
- Nillable 値ならびにXML インスタンスファイルにおち xsi:nil 属性をサポート
- XMLドキュメトにてターゲナに対する自動キャスティングを無効化

MapForce バージョン2010 リノース2 の新規機能

- 64ビットオペレーティングシステム上で<u>64ビット</u>バージョンのMapForce Enterprise / Professional エディションをサポート7 親コネクタの接続時に、対応する<u>子接続</u>を自動的は接続
- SQL Where コンポーネント にてフィールドのサポート
- <u>compiled Java</u>.class と.NET アセンブリファイルを追加可能に
- 入力文字列のトーケルによ更なる処理

MapForce リース2010 の新規機能

- コポーネト毎に複数の入力/出力サポート
- 相対パスサポトのアップグレード
- xsi:type のサポートによる<u>派生型</u>の使用
- 新たい追加されたデータ型システム
- 改善されたユーザー定義関数のナビゲート(ユーザー定義関数のナビゲートを参照してくたさい)
- XML 要素におる 複合エレテンツの処理を強化

24 はじめに

MapForce バージョン 2009 SP1 の新規機能

- ユーザー定義関数内の ラメーターの順序はユーザーにい定義されることができます(ユーザー定義関数内の ラメーターを参照 してくたさい)
- XML スキーオマ対して妥当ではないXML ファイルの処理
- 全般的(標準的)なユーザー定義関数における複雑な階層構造の引数サポート(<u>インラインと正規ユーザー定義関数</u>を参照してくたきい)
- C++ コードの生成時におお Apache Xerces 3.x のサポート

MapForce バージョン 2009 の新規機能

- マピングプロジェクトの<u>ドキュメンテーション</u>
- <u>SQL Server</u>のXML フィールドをネイティブサポート
- ノードコンテンソやノードのグループ化
- シーケンス内のノードの位置をベースリュしたデータのフィルタレング
- QName サポート
- コンポーネト内にあるアイテムノードの検索

MapForce バージョン 2008 リノース2 の新規機能

- XML ファイルのためコ<u>XML スキーマ</u>を自動生成
- 生成された Java ならびに C#コードの入出力とて ストリームオブジェクト をサポート
- C++ ならびにC# におお Visual Studio 2008 プロジェクト ファイルの生成
- 生成されたコードからデータベーススキーマ名を除外するオプション
- データベースコンポーネント内の視覚的なテーブルとて用意された<u>SQL SELECT ステートメント</u>
- <u>ローカルルーション</u>--主キー/外部キールーションシップを素早く作成
- Altova <u>グロー いレノース</u>のサポート
- パフォーマンスの最適化

MapForce バージョン 2008 の新規機能

- <u>集計</u> 関数
- Value-Map 検索コンポーネント
- XML 出力オプションの強化: XML 出力の整形、個々のエンポーネントに対してのエンコーディング設定や XML スキーマ参照の 省略
- 内部におお様々なアップデート

MapForce バージョン 2007 リノース3 の新規機能

- データベースフィールドへからXMLデータマング XMLデータをデータベースフィールドへ/からマング を参照してくたさい。
- データベースへの<u>直接のケエリ</u>
- <u>SQL-WHERE</u> フィルターならびにSQL ステートメントウィザード
- <u>コード ジェネレータ</u>の最適化ならび ジキュメンテーションの改善

1.2 サポートメモ

MapForce® は32/64 ビナ Windows アプリケーションで次のオペレーティングシステムで作動します:

- プラオフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10
- プラオフォーム更新済みのW indows Server 2008 R2 SP1 ませお以降

64ビナのサポーはEnterpriseとProfessional エディションでご利用レオオメニとが出来ます。

MapForce はオプタンで次の統合された開発環境にプラグインとしてご利用したさくことが出来ます:

- Visual Studio 2010/2012/2013/2015/2017/2019、<u>Visual Studio のためのMapForce プラブイン</u>を参照してくたさい。
- Eclipse 2020-12, 2020-09, 2020-06, 2020-03、Eclipse のためのMapForce プラグインを参照してくたさい。

MapForce はMicrosoft Office 製品を以下のよう、統合します。

- Access データベースへデータを、おける Access データベースへデータからデータをマップすることができます。サポートされるバージョン 理しては、データベースと MapForce を参照してくたさい。
- Word 2000 おけお以降のフォーマナ、でマナピングドキュメントを生成することができます。マナピングドキュメントの生成とカスタマイ 乙を参照してくたさい。

プログラムコード生成に適用することのできるサポート情報に関しては、コード生成についての紹介を参照してください。

サポートされるデータノースとターゲナ

MapForce マルングの変換言語を変更する場合、その特定の言語のために一部の機能がサポートされていたい場合があります。次のテーブルは、サポートされるデータノースとターゲル書式を言語ごとに、MapForce Professional Edition 別にリストしています。

脚注:

 "Built-in" はマナングを MapForce 内の 出力」 タブをクリックすることによりマナングを実行する、または MapForce Server 内でマナングを実行することを意味します。

マッピング	形式	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
XML ¹		۲	۲	۲	۲	۲	۲	۲	۲
ر ځ csv	7 7					۲	۲	۲	۲
バイナレス	アイル								۲
	ADO					۲	۲		۲
データベ ズ	ADO.NET						۲		۲
	JDBC							۲	۲
	Native SQLite								۲

マッピング形式		XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
	Native PostgreS QL								۲
	ODBC					۲	۲	۲	۲

脚注:

- 1. BUILT-IN を変換言語と、て使用する MapForce Enterprise Edition によりのみ電子署名処理を使用する XML はサポートされています。
- 2. データベースの種類とターゲトの環境により制限は適用されます。詳細に関しては、<u>異なる実行環境内のデータベースマメピング</u>を参照してくたさい。

生成されたコード内のサポートされる機能

MapForce Professional Edition 内の各言語内のコード生成とサポートに関連する機能を次のテーブルおノストしています。

機能	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
マピンバン ラメータを 提供する	۲	۲	۲	۲	۲	۲	۲	۲
マピングから動的に入 カファイル名を提供する	۲	۲	۲	۲	۲	۲	۲	۲
マピグ入力とでアイ ルドカードファイル名を提 供する1		۲	•	۲	•	۲	•	۲
マピングから動的に出 カファイル名を生成する		۲	۲		۲	۲	۲	۲
マピングから文字列の 値を返す	۲	۲	۲		۲	۲	۲	۲
<u>変数</u>		۲	۲	۲	۲	۲	۲	۲
<u>ユポーネトの並べ替え</u>		۲	۲	۲				۲
関数のグループ別け		۲	۲		۲	۲	۲	۲
2110-	۲	۲	۲	۲	۲	۲	۲	۲
<u>ユポーネトの泊化</u>								۲
<u>Value-Map ンポネ</u> 込	۲	۲	۲	۲	۲	۲	۲	۲

機能	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuery	C++	C#	Java	BUILT- IN
デフォルトとード関数								۲
マナングの別外		۲	۲	۲	۲	۲	۲	۲
<u>文字列の解析とメアル</u> 化 ²								۲
<u>動的なノード名</u>		۲	۲	۲	۲	۲	۲	۲
データベースジレク入力								۲
<u>入力パラメーターを持た</u> <u>ないデータベース SQL</u> <u>SELECT</u>					۲	۲	۲	۲
<u>入力/ データーを持つ</u> データベース SQL <u>SELECT</u>								۲
<u>データベースストアドプロ シージャ</u>								۲
<u>データベース例外の扱い</u> 3					۲	۲	۲	۲
<u>データベーストレースとエラ</u> <u>ーのエグ</u>								۲
<u>MapForce Server</u> 実行ファイルの生成								۲
<u>FlowForce Server</u> <u>אסדיציאיאי</u>								۲
<u>バイナリファイルからデータ</u> <u>を読み取る</u>								۲
<u>バイナリファイルニデータを</u> 書き込む								۲

脚注:

- 1. XSLT 2.0、XSLT 3.0 とXQuery はfn: collection 関数を使用します。Altova XSLT 2.0、XSLT 3.0 と XQuery エンシハの実装はフィルドカードを解決します。他のエンジノは異なる振る舞いをする可能性があります。
- 2. JSON に関しては解析ビリアル化 Java とC# はつれえサポトされています。
- 3. 前のテーブルで示されているとおり、データベース例外の扱いはマンピング言語が現在接続されているデータベースドライバーによりサポートされている場合に可能しております。

1.3 MapForce とは?

Altova W eb サイト: 🥜 <u>データマンピングルール</u>

MapForce は、プログラムコードを作成することよく視覚的なドラッグアンドドロップを使用するグラフィカルはコー・ザーインターフェイスを用いデ 一弦異なるデータ間、スキーマ間で変換するWindows ベースの多目的のIDE (統合された開発環境 統合開発環境)です。おこ、 事実上、MapForce は時際のデータ変換(おこデータマノピング)を行うプログラムコードを生成します。プログラムコードを生成しない場合、 (MapForce Professional おさまEnterprise Editionsで使用することのできる) MapForce 内蔵の変換言語を実行します。

内蔵の変換言語のためにデザインされているデータマンングはMapForce Server を使用した自動化に適しています。MapForce Server を使用すると、コマイドラインからのみではなく、API 呼び出し(.NET、Java、COM) から、おけよ予定された、おけは再帰的な ジョブとしてFlowForce Server の管理下でWindows、Linux、およびmacOS 上で実行することができます。詳細に関しては MapForce Server を使用した自動化を参照してくたさい。

MapForce を使用してデザインされたマンピングによりデータを便利に、ファイルベース、おうは他のフォーマトに変換することができます。使用するテクノロジー(現わらず MapForce は通常自動的にデータの構造を決定、おうはデータのためにスキーマを提供するオプションを与えます。MapForce はサンプルインスタンスファイルから自動的に生成することもできます。例えば、XML インスタンスファイルが存在し、スキーマ定義が存在しない場合、MapForce はスキーマ定義を生成し、他のファイル、おうはスォーマナのかがにXML ファイル内のデータを使用 できるようします。

MapForce Basic Edition	MapForce Professional Edition	MapForce Enterprise Edition
・ XML とXML スキーマ	 XML とXML スキーマ エンマで区切られた値(CSV)と固定長 フィールドフォーマナ(FLF)を含むフラットファイル データベース(Microsoft Access や SQLite データベースなどを含む全てのメ ジャーなデータベース) バイナリファイル(生 BLOB コンテンソ) 	 XML とXML スキーマ エンマで区切られた値(CSV)と固定長 フィールドフォーマナ(FLF)を含むブラットファイル Map Force FlexText を使用してマップ し、他のフォーマナトに変換することのでき るレガシーテキストファイルからのデータ。 データベース(Microsoft Access や SQLite データベースなどを含む全てのメ ジャーなデータベース) (UN/EDIFACT、ANSI X12、 HL7、IATA PADIS、SAP IDoc、 TRADACOMSなどを含む)EDI フォ ーマナのファミリー JSON ファイル Microsoft Excel 2007 と以降のファ イル XBRL インスタンスファイルとタクノノミ プロトコルレジファー バイナリファイル(生 BLOB コンテン)

マピングノースませる、ケイトとてサポートされるテクノロジーは以下のとおしです。

MapForce エディションにより、データの変換言語を以下のように選択することができます。

MapForce Basic Edition	MapForce Professional Edition	MapForce Enterprise Edition
• XSLT 1.0 • XSLT 2.0 • XSLT 3.0	 MapForce内蔵の変換言語 XSLT 1.0 XSLT 2.0 XSLT 3.0 XQuery Java C# C++ 	 MapForce 内蔵の変換言語 XSLT 1.0 XSLT 2.0 XSLT 3.0 XQuery Java C# C++

全ての変換の結果、おび生成されたXSLT おけXQuery コードをグラフィカルユーザーインターフェイスから移動することなくプレビューすることができます。デザインおけはマンピングをプレビューする際、MapForce はスキーマおよび変換の整合性を検証し、検証エラーを専用のウイドウで表示するけか、すぐに確認して、エラーの修正を行うことができます。

Java、C#、おけまC++を変換言語とて選択する場合、MapForceは必要とれるプロジェクトとリューションを生成し、Visual Studio おけまE clipse で直接開き、生成されたデータマメピングプログラムを実行することができます。高度なデータ統合のシナリオの場合、自身のコードを用いAltova とMapForce APIを使用して、生成されたプログラムを拡張します。

MapForce マピングの変換言語を変更すると、その特定の言語のために特定の機能がサポートされていたい場合があります。サポート

MapForce では、全てのマピングの変換を視覚的にデザインすることができます。例えば、XML の場合、全ての要素と属性を接続する ことができます、おけは要素おけよ属性他のXML ファイルの要素おけお属性のXML ファイルにコメントを挿入します。これにより、 MapForce にノース要素(おけよ属性)からデータを読み込み、ターゲオ 要素(おけよ属性)に書き込むことを命令します。



2つのXML ファイル間のサンプルデータ変換

同様に、MapForce Professional おけまEnterprise Editions でデータベースと作業する場合、MapForce のマピングエル内の データベースカラムを確認し、データをマゾル、視覚的な接続を作成することができます。Altova MissionKit 製品と同様、MapForce からデータベース接続を設定する場合、柔軟的にデータベースドライバーと接続の型(ADO、ADO.NET、ODBC、まけまJDBC)を既 存のインプラストラグチャとデータマッピングの必要に応じて選択することができます。更に、SQL クエルを視覚的に作成し、ストアドプロシージャ を使用し、まけはデータベースに直接問い合わせることができます。(データベース型、エディション、ドライバーイによりサポートは異なります)。



XML ファイルとデータベース間のサンプルデータ変換

MapForceを使用したマングデザインのとても簡単な例として、「Xからデータを読み込みYに書き込む」が挙げわれます。しかしながら、「Xからデータを読み込み、Yに書き込む。そしてYからデータを読み込みZへ書き込むはどのMapForceシナルを簡単にデザインすることができます。これらは「パススルー」まけは「チェーン」マンピングとして知られ、これらにより変換プロセスの中間のステージでデータにアクセスすることができます(例えばファイルとして保存するかか)。

MapForce内で作成することのできるデータマンピングは、単一かつ定義済みファイルだけではありません。同じ変換内で、ディレクトリから複数の入力ファイルを動的に処理することができ、複数の出力ファイルを生成することができます。ですから、「複数のXファイルわらデータを読み込み、単一のYファイルに書き込む」 おさは [ファイルX を読み込み複数のファイルY を生成する」といオシンナリか可能になります。

重要な点は、同じ変換で、使用中のMapForceエディションでサポートされる全てのデータ型の複数のソースと複数のターゲトを混合 てきる点です。例えば、MapForce Professional おけまEnterpriseの場合、2つの異なるデータベースを単一のXMLファイルニマー ジすることが可能です。おけま、複数のXMLファイルからのデーダをマージし、一部のデーダを1つのデータベースとして書き込み、一部のデータ を他のデータベースとして書き込むことができます。SQLステートメートをデータベースに定いたする前にプレビューすることができます。

ソースからターゲトへのデータの直接の変換だけが達成する事柄ではかません。多くの場合、データを特定の方法で処理する必要がある場合がかます(例:並べ替え、グループ化、おけまフィルター)などを最終プロセスの前に行う必要があります。この理由のため、MapForce は、プログラング言語の構成を簡素化するその他の関数コンポーネント(規則、変数、SQL-W HERE 条件、フィルターと並べ替えコンポ ーネントなど)を搭載しています。他方、MapForceは、全てのデータ操作をアンストする豊富かつ広範囲の関数ライブラル搭載しています。

必要であれば、ビルドインライブラを、MapForceで直接関数(しわかるユーザー定義関数、おけまUDF)をデザインすることにおし、またはXSLT、XQuery、Java、おけまC#言語で外部的に作成された関数やライブラを使用して拡張することも可能です

ライブラリ	▲ 廿 ×
🔁 core	^
😑 aggregate func	tions
avg	
count	
max	
max-string	
min	
min-string	
string-join	
sum	
conversion function	tions
boolean	
format-date	
format-dateTime	2
format-number	
format-time	
number	
parse-date	
parse-dateTime	
parse-number	
parse-time	
string	×
開数の検索	•
ライブラリを追加/	削除
差ライブラリ 🛅 プロジ:	ェクト

ライブラリウィンドウ

データマピングデザインファイルの数が多すぎる場合、マピングをプロジェクトに整理することができます(MapForce Professional と Enterprise エディションで使用することができます)。これにより簡単なアクセスと管理を行うことができます。重要な点は、プロジェクト内の個別のマピングのためにコードを生成し、更にプログラムコードをプロジェクト全体から生成することができます。

(MapForce Server API を使用したマピング変換の実行時などの高度なデータ処理のかとてマピングをデザイン、ランタイムで値を パス おけようシタイムから単純型文字列の値を取得することができます。この機能は、関数おけよ単純型文字列の値を生成するマピング 全体を素早くテストすることを可能します。MapForceのProfessionalとEnterprise エディションは、他のプログラング言語内で同 様に動作する、ランタイムでの文字列の解析とパアル化を行うコンポーネントも含みます。 MapForce Enterprise Edition では、Webサービス言語定義(WSDL)ファイルをベースにたSOAP 1.0とSOAP 2.0ウェイ ブサービスを視覚的にデザインすることができます。WSDLースタイル、おけはRESTースタイルWebサービスをマノピング内から呼び出すことも できます。

MapForce Professional とEnterprise Editions では、マピングデザインファイルの詳細なドキュメントをHTML、およびRTF フォ ーマ・ト で生成することができます。ドキュメンテーションデザインをカスタム化することができます、例、ドキュメントから特定のコンポーネントを包 含、および、削除することが選択可能です)。

Altova MissionKit 製品と共にMapForce を使用している場合、MapForce はれらの製品とAltova サーバーベース製品を以下のテーブルで示されている通り統合します。

MapForce Basic Edition	MapForce Professional Edition	MapForce Enterprise Edition				
生成されたXSLT を直接 MapForce 内で実行し、データ変換の結果をすくに確認することができます。パウォーマンスを向上するこ は、超高速 XML 変換エンジン RaptorXML Server を使用して、マメングを処理します。						
XMLSpy がインストールされている場合、同じマンンで、関連するMapForce コンテキストからを直接 XMLSpy を開き、サポートされ るファイル型を便利は開き編集することができます。(例[コンポーネント XMLSpy 内でスキーマ定義を編集] メニューコマンドは XML コンポーネントをクトックすると使用することができます)。						
	データ変換を、直接 MapForce 内で、また して異なるマシンペオペレーティングシステムに 方法は、Windows 上でマメピングをデザイ FlowForce Server か動作するWindow 実行することができます。	はコマドラインや自動化された実行を使用 デプロイル実行することができます。具体的な ンし、MapForce Server おけよ ws、Linux、おけはMac サーバーマンンで				
	StyleVision か同じマンノニインストールさす StyleVision スタイルシートを再利用し、マ はWord 2007+ドキュメイトとしてプレビュ	これる場合、デザイン、既存の ととノの結果をHTML、RTF、PDF、おこ ーすることができます。				

MapForce Professional とEnterprise エディンコンは、Visual Studio とEclipse の統合された開発環境のプラグインとしてインストールすることができます。これにより、優先する開発環境を移動することなく、MapForce の機能にアクセスレマナピングをデザインすることができます。

MapForce では、開発環境の概観や使用感を完全にカスタム化できるだけでなく(グラス・カルユーザーインターフェイス)、各技術と各マッピングのエンポーネントの型の他の関連する設定もカスタム化することができます。例

- XMLから、おゴはXMLへ、マピングを行う際、スキーマレフォレンズを含むか否かを選択することができ、おゴはXML 宣言が XML 出力ファイル内で抑制されるへきかを選択することができます。生成されたファイルのエンコードを選択することができます(例: UTF-8)。
- データベースから、おさまデータベースへマンピングする際、データベースステートメントの実行のタイムアウト期間、MapForce かデ ータベーストランザプションを使用するか、おさまコード生成の際データベーススキーマ名からテーブル名を削除するかなど、などの設 定を定義することができます。
- XBRL の場合、MapForce か表示する構造ビューを選択することができます(「プレゼテーション定義リングベース」ビュー、 [テーブルレクベース] ビュー、おけま[全てのコンセプト] ビューなど)。

MapForce の全てのエディンヨンは32ビットアプリケーションで動作します。更に、MapForce Professional とEnterprise エディンヨンは64ビットアプリケーションで使用することができます。

1.4 基本概念

このセクションはデータマピングを開始するこあさい必要な基本概念のアナラインを紹介します。

マピング

MapForce マルングデザイン(おけよ 略して「マルング」)は、データがどのようコンのフォーマトから他のフォーマトに変換さえるかの 視覚的な表現です。マルングは、データ変換を行うかのMapForce マルングエルア内で「ンのスキーマから他のスキーマに追加するユン <u>ポーネト</u>から構成されています。(例: XMLドキュメトを1つのスキーマから他のスキーマに変換)。有効なマルシグは、1つまけは複数のタ <u>ーゲナコンポーネト</u>に接続されている、1つまけは複数の複数の<u>ノースンパーネト</u>から構成されています。マルングを実行して、結果を 直接 MapForce 内で確認することができます。コードを生成し、外部で実行することも可能です。MapForce 実行可能ファイルにマル ングをエン・イルレ、MapForce Server おけよ FlowForce Server を使用してマルングの実行を自動化することもできます。 MapForce は、マルングを、mfdの拡張子を持つファイルとして保存します。



MapForce マピングの基本的構造

コンポーネント

MapForce 内では、「エポーネト」とう用語はデータの構造、スキーマンませまデータかどのようこ変換、関数)されるかを視覚的に表しま す。エパーネトはすべての<u>マノピング</u>を構築する中心的な役割を果たします。マノピングエリアでは、エンポーネトは正方形の枠で表示され ます。以下は、MapForce エンポーネトの例です:

- 定数
- デ−ダベース
- フィルター
- 条件
- 関数コンポーネト
- EDI ドキュメント (UN/EDIFACT、ANSI X12、HL7)
- Excel 2007+ ファイル
- 単純型 入力エルポーネト
- 単純型 出力コンポーネト
- スキーマおよびDTD

コネクタ

<u>コンポーネト</u>の左右に表示される小さな三角形です。コンポーネトの左に表示されるコネクタはそのコンポーネトのエトリポイトを表します。コンポーネトの右に表示されるコネクタはそのコンポーネトからのデータの終了ポイトを表します。

接続

接続としてのユネクタ間に描くことができる線です。接続を描くことにより、MapForce にデータを特定の方法で変換するように命令します。 (例: XMLドキュメントからデータを読み込み、他のXMLドキュメントに書き込みます)。

ソースコンポーネント

ソースエンポーネントとは、MapForce がデーダを読み込む元のエンポーネントです。マンピングを実行すると、ソースエンポーネントのコネクタ により与えられたデーダを読み込み、必要とされる型に変換し、ターゲントコンポーネントのコネクタに送信します。

ターゲトコンポーネント

ターゲトコンポーネントとMapForce はガデーダを書き込むエンポーネントです。マンピングを実行すると、ターゲトコンポーネントは、外部プログラム内で更に処理するためにMapForce にファイル(おけよ複数のファイル)の生成、おけは結果を文字列の値として出力するように命令します。ターゲトコンポーネントは、ソースコンポーネントの逆です。

1.5 ユーザーインターフェイスの概要

MapForceのグランホルインターフェイスは、統合された開発環境として整理されています。メインのインターフェイス設定は「ツール」カスタマイズ」メニューコマンドを使用して行います。

各ウィンドウの右上に表示される▼ [♀] × ポタンを使用してウィンドウの、表示、非表示、ピン、ドックを行します。ツール デーとウィンドウをデ フォルトの状態に復元する場合は、メニューコマンド「ツール」 ツール デーとウィンドウの復元」を使用します。

下のイメージコはMapForce グラフィカルなユーザーインターフェイス



メニューバーとソールバー

メニューコマイドは、メニューアイテムを表示します。各ツールレイはMapForce コマイドのボタンのグループを表します。希望する場所についドルをドラッグすることにより、ツールレーを移動することができます。

ライブラリウィンドウ

ライブラリウイドウは、ライブラにとこ整理されたMapForce内蔵の関数をリストします。使用することのできる関数は「出力」メニューまたは言語選択ツールドーから選択される変換言語により異なります。変換言語の選択も参照してくたさい。ユーザー定義関数を作成した場合、まだまたノポートした場合、これらの関数もライブラリウイドウニ表示されます。

ライブラ	У У	•	ņ	×
2 000	re		/	~
	aggregate functions			
	avg			
	count			
	max			
	max-string			
	min			
	min-string			
	string-join			
	sum			
🚅	conversion functions			
	boolean			
	format-date		_	
	format-dateTime			
	format-number		_	
	format-time			
	number		_	
	parse-date			
	parse-dateTime		_	
	parse-number			
	parse-time		_	
	string		1	/
関数の	の検索			-
	ライブラリを追加/削除			
2 51	ブラリ 🋅 プロジェクト			

名前ませば詳細による関数の検索は、ライブラリウィンドウの下のテキストボックスに検索の値を入力することにより行うことができます。(現在ア クティブなマメニング内で)関数の全ての発生を検索するコよ、関数を右クリックして、コンテキストメニューから全ての呼び出しを選択します。 また、ライブラリウィンドウから直接関数データ型と詳細を確認することができます。詳細に関しては、関数と作業を参照してくたさい。
はじめに

ライブラリウィンドウの管理

このケイドウから現在開かれているマメピングにより、コンパイル済みのJava クラスファイル.NET DLL アセンブノファイルを含む 使用されているすべてのユーザー定義関数(UDF)とインポートされナカスタムライブラを表示し管理することができます。

デフォルトではライブラノの管理ウインドウは表示されていません。表示するココお以下の一つを行うことができます

- 「表示」メニューから「ライブラリの管理」をクリックします。
- ライブラノウィンドウのベースのライブラリの追加/削除をクトックします。

ライブラリの管理	×
🔁 🤯 FormatNumber.mfd	-
日 自分のライブラリのインポート 追加	
🕞 🔂 Format.dll 🗙 オプション	
f_x Format, Version=0.0.0.0, Culture=neutral, PublicKey	i i
f _x Format, Version=0.0.0.0, Culture=neutral, PublicKey	i.
自分のライブラリのインポート	
😢 グローバルライブラリのインポート 追加	
	~

UDFとライブラを現在アクティブなマッピングのナックにのみ、ます」はすべての開かれているマッピングのナックに表示することを選択できます。現在開かれているすべてのマッピングのナックにインポートされた関数とライブラを表示するコよ、ウインドウ内を右クリックし、コンテキストメニューから「開かれているドキュメントの表示」選択します。

名前の代わりに開かれているマピングのいてを表示するコは、コンテキストメニューから「ファイル」なの表示」を選択します。

詳細に関しては提数ライブラリの管理を参照してくたさい。

プロジェクトウィンドウ

MapForce 複数のドキュメトィンターフェイスをサポートし、マメピングをマメピングプロジェクトとしてグループ化することができます。プロジェクトウィドウはプロジェクトに追加されたすべてのファイルとフォルダーを表示します。プロジェクトファイルコス 拡張子*mfp (MapForce プロジェクト) かあります。プロジェクト内でマメピングを検索するコス プロジェクト内をクリックして、「CTRL + F」を押します。詳細に関しては、マッピングプロジェクトとの作業を参照してください。



マピングペイン

マルングペイノはマッピングをデザインする作業エリアです。「挿入」メニュー(ユンポーネントをマッピングご追加を参照)からマッピングエリアに (ファイル、スキーマ、エンスタイ、変数などの)マッピングコンポーネント追加することができます。 ライブラリウイドウェ表示されたマッピングペイン 関数をドラッグすることもできます。 マッピングご関数を追加を参照してください。

XSLT ペイン

XSLT ペインは「XSLT」ボタンをクリックすると、マノピングから生成されたXSLT 変換コードを表示します。このペインを切り替えるゴは XSLT、XSLT 2 まけはXSLT3 を変換言語として選択し、同じ名前のタブをクリックします。

このペイム、ラインの対番とコードの折げオンサ機能を提供します。コードの一部を展開おけば折げオまけには、ウイドウの左側の「+」と「-アイコンをクルクします。折げオテまれたコードは、省略記号と共に表示されます。折げオーまれたコードを展開することなくプレビューするには、 マケスで省略記号をポイトすると、プレビューされるコードを表示したとことがでに表示されるように開かれます。プレビューされたテキストが大きくとっト内に表示できない場合、ヒートの後に追加表示記号が表示されます。



(インデント、行末マーカーを含む)表示設定を構成するコよ、ペインを右クトックして、コンテキストメニューから「テキストビュー設定」を選択します。ませよ、「テキストビュー設定」(国))ツール いーポタンをクトックします。

XQuery ペイン

XSLT ペインは、「XQuery」ボタンをクリックすると、マンピングから生成されたXQuery変換コードを表示します。このペインは XQueryを変換言語として選択すると使用することができます。このペインもXSLT ペインと同様の仕組みを持つラインの付番とコードの折り たまっか機能を提供します(上を参照)。

DB ケエッペイン

DB クエリペインでは全てのメジャーなデータベースに対して直接問い合わせを行うことができます。複数の異なるデータベースへのアクティブな接続と作業することができます。

データ	ベースブラウザー										
ą	altova	 ■ altova 		•	🗉 🕨 🛓	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	🐺 间				
٢	• 🔻 🕈 🔊			1	SELECT	[Primaryk	(ey],[For	reignK	(ey],		
	altova ADO			2	FROM [Address];	[street]],[21]	1		
14	コーテーブル	▼							s	OLI	ディター
	- □ (in 2-ザ- - □ (in 2-i) - □	テーブル ess irtment e テーブル									
1	1 🖻										
	PrimaryKey •	ForeignKey 🎈	city •	state 🏾	street		•	zip 🎈			
1	1	1	Vereno	CA	119 Oakstre	et, Suite 4876	5	29213	-(クエリ	結果
2	2	2	Brenton	MA	9865 Millen	ium Center, S	uite 456	5985			
0	取得の完了					行: 2, 列: 6	0.109 秒	16:	25:22		
結果	果 メッセージ										
_ <mark>√</mark> ୭	ピング XSLT	DB クエリ	出力								
結果	xvt-:	ジタブ									

更に詳しい情報に関しては、データベースの参照とフェレを参照してけたい。

出カペイン

出力ペインは、「出力」ボタンをクリックするとマッピングの変換結果を表示します、例 XML ファイルシ。マッピングが複数のファイルを生成する場合、それそれの生成されたファイルを順番にナビゲートすることができます。



XSLT ペインと同様の壮組みを持つラインの村番とコードの折りナナみ機能を提供します(上を参照)。

Style Vision 出力ボタン

Altova StyleVision (<u>http://www.altova.com/ja/stylevision.html</u>) がインストールされている場合、StyleVision 出力ボタインドレ マピングの出力をプレビューし HTML、RTF、PDF、とW ord 2007+ フォーマナで保存することができます。これは StyleVision でデ ザインされた StyleVision Power StyleSheet(SPS)をMapForce 内のマピングコンポーネントに割り当てることに、いう ます。

概要ウィンドウ

概要ウインドウは、マンピングペインの概観図を表示します。マンピングのサイズが大きい場合、マンピングエリアの特定の場所に移動する場合
し、使用します。マンピングの特定の場所に移動する場合、赤い枠の長方形をクリックアンドドラングします。



メッセージウィ・ドウ

メッセージウイドウコはマピングの<u>プレビュー</u> おけは検証中の警告やエラーメッセージが表示されます。

Messages	×
✓ ▼ ▲ □ □ □ ● ۹ & & ×	
💦 🖓 📴 🔗 BooksToLibrary.mfd: Mapping validation successful 0 error(s), 1 warning(s)	-
🕞 👍 þ <u>publication</u> has no input connection and will be generated only once.	
You can try to connect \triangleright book with \triangleright publication to resolve.	
🕞 🛄 🕕 🚺 The output component 🦉 library has no output file name set. A default file name will be use	d.
9	
6	-

情報、警告お台エラーシッセージをトリガーしたコンポーネトお台は構造のマッピングエアをハイライトする日よ、シッセージウイドウ内の下線のコ、サテキストをクリックします。

メッセージウインドウ内のマパング実行ませば検証オペレーションの結果は、以下のアイコンのIつと共に表示されます:

アイエン	説明
0	オペーション式功しまた。
	オルーションは響告と共に終了しまた。
8	オペーションに失敗しまた。

*メッセージウイドウは次のメッセージを追加して*表示する場合かります。情報メッセージ、警告、およびエラー

アイコン	説明
0	情報メッセージを表示します。情報メッセージはマングの実行を停止しません。
	警告メッセージを示します。 警告メッセージを表示します。 警告はマッピングの実行を停止しません。 例えば、 必須の入力コネクタへの接続が作成されなかす。場合、 警告メッセージが生成されます。 このような場合、 有効な接続を 持つエンポーネントのナックに出力が生成されます。
0	エラーメッセージを表示します。エラーが発生すると、マピングの実行に失敗し、出力は生成されません。XSLT おこ はXQuery コードのプレビューの不可能です。

メッセージウィボウ内の他のだタイことり以下のアクションを実行することができます:

アイコン	説明
V	重要度(情報メッセージ、エラー、警告)別にメッセージをフィルターします。「すべて選択」を選択して、全ての重 要度を選択します(これはデフォルトの振る舞いです)。
	「チェックをすべて解除」を選択して、フィルターからすべての重要度を解除します。この場合、全般的な実行おこ は検証の状況に関するメッセージのみか表示されます。
	次の行に移動させる。
	前の行に移動させる。
	選択された行をクリップボードにコピーする。
Đ	ネストされたラインを含む選択されたラインをクリップボードにコピーします。
A	タッセージウイボウの全てのエレテンンをクリップボードにコピーします。
٩	ッセージウイドウ内で特定のテキストを検索します。オプションで、特定の用語を検索する場合は、「単語の完全 マッチ」を選択します。単語の大文字とい文字を区別する場合「大文字とい文字を区別」を選択します。
đ	特定のテキストを現在選択されているラインから最後まで検索します。
t.p	特定のテキストを現在選択されているラインから最初まで検索します
×	シッセージウインドウをクリアする。

複数のマンピングファイルと同時に作業する場合、個々のマンピングのための個別のタブ内で情報、警告お日はエラーを表示する必要かあることがあります。この場合、マンピングを実行お日は検証する前に、メッセージウイドウの左横にある番号の付け、オンダンをクリックします。

アプリケーションステータスバー

アプリケーションウイドウの下部にあるアプリケーションステータスバーコよアプリケーションに関する情報が表示されます。ツールバーアイコイに対してマケスポインターを重ねると、ツールチップ情報がこのステータスバーに表示されます。アプリケーションステータスバーとデータベースクエリステ ータスバーは別のものです。64ビットバージョンのMapForceを使用している場合、アプリケーション名の後に(x64)という文字列が加えられます。32ビットバージョンでこのような表示にお行われません。

1.6 規約

サンプルファイル

このドキュメントで参照される.mfd 拡張子 と他の伴うインスタンスファイルを持つファイル)データマンピングデザインファイルの多くは次のフォル ダーで検索することができます:

- C:\Users\<username>\Documents\Altova\MapForce2021\MapForceExamples
- C:\Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\Tutorials

MapForce に伴うサンプルマッピングとインスタンスファイルは動作の多くのアスペクトを紹介し、MapForce を使用 するに当たり色々と試すことが奨励されます。オリジナルのサンプルへ直接変更を加えることを希望しない場合 は、変更前にバックアップを作成することが奨励されます。

グラフィカルユーザーインターフェイス

このドキュメトトに伴うイメージ(スクレーンショナ)の一部は、使用中のMapForce エディションでは、使用することのできない グラフィカルユー ザーインターフェイスの要素を含む場合があります。 関連したニロンテキストで、イメージは通常ソースマンピングデザイン(*.mfd) ファイルの名 前、画像が作成された MapForce のエディションを含みます。

2 チュートリアル

MapForce チュートリアルは、MapForce のデータの基本変換機能を短い時間で理解することをお手伝いします。チュートリアルの目的 は全てのMapForce 機能を説明することではおりませんが、MapForce の基本を加速します。ですから、チュートリアルを順番ど おりにフォローすることが奨励されます。コンセプトは順を追って複雑しなるさめ、各コンセプトを次のコンセプトに移動する前に理解することが 重要です。XML およびXML スキーマイン対する基本的な知識はとても有利です。

XMLを新しいスキーマニ変換

このチュートリアルは、コードを書く必要なくXSLT 2.0 言語を使用してXML 構造から他にデータを変換する方法について説明します。また、MapForce シーケンスとアイテム、マンピング接続作成、関数の使用方法、マンピングの検証およびプレビュー、と出力結果をディスク に保存する方法などについて学ぶことができます。

複数のノースから1つのターゲトにマップ

このチュートリアルは、異なるスキーでを持つこつのXML ファイルからのデータの読み込み方法や単一のターゲル XML ファイルへのマージの 方法を説明します。また、各マノピングコンポーネントの名前とインスタンスファイルの変更方法や入力の複製の方法などについて学ぶことがで きます。

複数のターゲナスキーマとの作業

このチュートリアルは、2つ以上のターゲト出力を作成する更に複雑なマンピングとの作業方法について説明します。具体的な方法は、同じ マンピング内でブックレコードのJストを保管するファイル、最初のXML ファイル内のブックのサブセトのみを含む他のXML ファイルを生成し、 発行年によりフィルターします。データのフィルタレングをサポートするコよ「フィルター」コンポーネント、関数、数値定数を使用します。

ファイルを動的に処理と生成する

このチュートリアルは、同じフォルダーニ存在する複数のXML インスタンスファイルからデータを読み込み複数のXML ファイルニ書き込む方法を説明します。また、XML とスキーマ宣言の削除および、関数を使用して、文字列を連結しファイル拡張子を抽出する方法についても学ぶことができます。

2.1 XML を新しいスキーマに変換

このチュートリアルは、MapForce開発環境の基本を学びつつ、2つのXML ファイル間のデータの変換方法について説明します。両方の XML ファイルはブックのリストを保管しますが、その要素は、異なる方法により名前を付けられ整理されます。(すなわち、2つのファイルは異なるスキーでを持ちます)。



データ変換の抽象的なモデル

下にしていされるコードは、データノースとして使用されるファイルからのサンプルのデータを表示しています。簡略化するために、XMLと名前空間宣言は省略されています。

```
<books>
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id="1">
<book id=>The Adventures of Tom Sawyer</title>
<book id=>Fiction</category>
<book id="2">
<book id=>2">
<book id=>2"<book id=>2"<book
```

books.xml

ターゲナ(デスティネーション)ファイルではデータは以下のように表示されます

library.xml

ソースとターゲト XML 内の一部の要素の名前は、同じではかません。このセグションの目的は、ソースファイル(<author>、 <title>、<category>、<year>)内の同等の要素から、ターゲトファイルの<author>、<title>、<genre>と <publish_year> 要素を表示することです。ソースXML ファイル内の属性 id は、ターゲト XML ファイル内の<id>要素にマップ される必要があります。最後に、ターゲトファイルの<last_updated> 要素をファイルが最後に更新された日付と時刻として表示するよう に設定する必要があります。

必要とされるデータ変換を行うには、以下のステップを行います。

ステップ 1: XSLT2 を変換言語として選択する

これを以下の方法で行うことができます。

- 「XSLT2」 リントーボタンをクリックします。
- 「出力」メニューから「XSLT 2.0」をクトックします。

ステップ 2: ソース XML ファイルをマピングに追加する

このマピングのナダのノースXML ファイルは、は、以下の、なで見つけることができます。<マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\books.xml。ファイルを以下の方法でマピングに追加することが できます。

- 「XML スキーマ/ファイルの挿入」 (型) ツールドーボタをクトックします。
- 「挿入」メニューからXML スキーマ/ファイル」をクリックします。
- XML ファイルをウィンドウエクスプローラーからマングエノアインデラッグします。

ファイルはマピングエリアに追加され、構造を一見することができます。MapForce では、この構造はマピングエレポーネトとして知られており、おとは単純にユンポーネト、コンポーネト内の要素を、折けまた(日)と展開するアイエン(田)をクリックすることにより、またに数字キー・パドの「+」と「-」キーを押すことによりすることに、拡張することができます。



マピングコンポーネント

マピングペイン内でエポーネトを移動するコよ、エポーネトヘッダーをクリックして、マウスを新しい場所にボラッグします。エポーネトのサイズを調整するコよ、エルポーネトの角をドラッグします」。角をダブルクトックするとMapForce は自動的にサイズを調整します。

トップレベリレード 💭 は、ファイル名を表します。このサンプルの場合、は、 XML インスタンスファイルの名前を表示します。構成内の、 XML 要素は 🖸 アイコンに、以表示されており、 XML 属性は 🖃 アイコンに、以表示されています。

コンポーネントの両側に表示されている小さな三角形は、(左側にある場合)データ入力を、(右側にある場合)データ出力を表します。 MapForce では、それぞれ入力コネクタ、出力コネクタと呼ばれます。

ステップ 3: ターゲナ XML スキーマをマッピングに追加する

ターゲト XMLを生成するコよ、既存のXML スキーマファイルを使用します。実生活のシナリオでは、このファイルは第三者パーティから提供される場合があり、おゴよXMLSpy などのソールを使用してユーザーが作成することができます。XML データのオゴのスキーマファイルが存在してい場合、MapForce は伴うスキーマおゴゴスキーマレファレンスを持たないXML ファイルをマメピングに追加すると、スキーマを生成するようにプロンプトします。

このサンプルでは、以下で見つけることのできる既存のスキーマファイルを使用します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\library.xsd。マパングに追加するコよ、ソースXML ファイルと

同様のステップを踏んでくたさい。(「XML スキーマ/ファイルの挿入」(
「」) ツール ー ポタンをクトックします)。 MapForce にインスタ ンスファイルを提供するよう コプロンプト されると、「スキップ」をクトックします。

MapFo	prce
?	MapForce を使うことで、XML スキーマをソースならびにターゲットとして定義することができます。 ソーススキーマにサンプル XML ファイルまたはグローバルリソースを付加して変換のプレビューを行う ことができます。
	サンプル XML ファイルまたはグローバルリソースを与えますか?または何も与えません。
	参照(B) スキップ(S)

この時点では、マピングデザインは、以下のようこなります:

😫 books		월 library
📮 🗋 File: books.xml File	7	🔁 🗍 File: library.xml File
books	¢	👌 🕀 🗘 library 🗘
book	¢	לא 🚽 🗘 last_updated 🖒
> = id	¢	🖟 🕀 🗘 publication 🛛
author	¢	
title	¢	
category	¢	
🖒 🦾 🗘 year	¢	

ステップ 4: 接続の作成

ソースXML ファイルのそれぞれの<book>内で、ターゲナ XML ファイル内に新しい<publication>を作成します。ですから、ソース コンポーネント内の<book>要素とターゲナコンポーネント内の<publication>要素を接続します。マメング接続を作成するコよ <book>要素の右の出カコネクタリンさよ三角形)をクリックし、ターゲナト内の<publication>要素の入カコネクタリン・ラックします。

これを行うと、MapForce は自動的に、ターゲホファイル内に同じ名前を持つノースファイル内のくbook>の子の全ての要素を接続します。ですから、4つの接続が同時に作成されます。この振る舞いな「子要素の自動接続」と呼ばれ、必要に応じて無効化ませまカスタム化することができます。



「マッチする子を自動的に接続する」の振る舞いを以下の方法で有効化ませま無効化することができます。

- 「子の自動接続の切り替え」())ツールレードタンをクルクします。
- 「接続」メニューから「マッチする子を自動的に接続する」をクリックします。

ターゲナコンポーネントの入力コネクタの一部が、MapForce により、必須アイテムを表示する、オレンジ色にマイライトされていることに注意してくたさい。XML ファイルの妥当性を確認するために、必須アイテムの値を以下のように提供してくたさい

- ターゲト内の<genre> 要素とノース内の<category> 要素を接続する。
- ターゲオ内の<publish_year> 要素とノース内の<year> 要素を接続する。

最後に <last_updated>要素への値を提供する必要があります。入力コネクタにマウスをポイントすると、要素はxs:dateTime型であることが表示されます。ビトを表示すること、「ビートの表示」 「「ジール・・・パン・カネカオ効化されている必要があることに注意してくたさい。



「データ型の表示」 ビン ツールバーボタンをクリックすることにより、常に各アイテムのデータ型を表示することができます。

現在の日時(すなわち、xs:dateTime の値)を関 XSLT 数の日付と時刻で取得することができます。マッピングに 対する XSLT 関数を検索するには、<u>ライブラリケイウの下の部分にあるテキストボック</u>スゴ date 」を入力します。おけよ マノピング の空のエリアをダブルクトックし、「current-date」を入力します。

context functions	Returns the current xs:date1ime.
current-date	result 7 /r
current-dateTime	result – fr
current-time	result = fr
default-collation	result = fr
implicit-timezone	result = fr
last	result = fr
adjust-date-to-timez	l time funci on result = fr
adjust-date-to-timez adjust-date-to-timez adjust-date-to-timez	I time funct on result = fr on result = fr

ルドーガタンの有効化されていることを確認してくたさい。

マピングは関数を追加するには、関数をマピングペインインラッグし、 <last_updated> 要素の出力を入力に接続します。



(books.xsd スキーマを持つ) books.xml インスタンスファイルのデータから(library.xsd スキーマを持つ) library.xml ファイル ヘデータを変換する MapForce マッピングデザイン(または「マッピング」) を作成することに成 功しました。各コンポーネントのヘッダーをダブルクリックすると、コンポーネント設定ダイアログボックス内のこれ らと他の設定を確認することができます。

🕑 コンポーネント設定	×
コンポーネント名: books	
ースキーマ ファイル(F)	
books.xsd	参照(W) 編集(T)
- 入力 XML ファイル(I)	
books.xml	参照(B) 編集(E)
出力 XML ファイル(O)	
	参照(R) 編集(D)

ソースのためのコンポーネント設定

	×
コンポーネント名: books	
- スキーマ ファイル(F)	·
library.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
	参照(B) 編集(E)
出力 XML ファイル(O)	
library.xml	参照(R) 編集(D)

ターゲートのコンポーネント設定

ステップ 5: マピングを検証し保存する

マルングの検証は、マルングを実行する前に、マルングエラーの可能性を確認し修正するオメの任意のステップです。マルングが妥当であるか確認するオメリ、以下を行います:

- 「ファイル」メニューから「マッピングの検証」をクリックします。
- 「検証」 ジール デオタ をクリックします。

「メッセージ」ウイズウは、検証の結果を表示します。

メッセージ	→ ‡ ×
Souther State Stat	
e se	
4	
42	

メッセージウィンドウ

この時点で、ファイルにマンピングを保存します。マンピングを保存するコよ、以下を行います

- 「ファイル」メニューから保存」をクリックします。
- 「保存」 📓 ツール いや おや をりかんます。

このチュートリアルで作成されたマッピングは以下で検索することができます: 〈マイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\\BooksToLibrary.mfd。ですから、これから先、自分で作成したマッピングを継続して使用するか、ませるBooksToLibrary.mfdファイルを使用することができます。

ステップ 6: マピングの結果のプレビュー

マピングの結果の結果を直接でMapForce でパビューすることができます。これを行うには、マピングペインの下にある「出力」、ドタンをクリックします。MapForce は、変換を実行し、マピングの結果を「出力」、ペインに表示します。



出カペイン

MapForce 内で変換の結果を確認することができます。

デフォルトでは、「出力」ペイン内にプレビューのナックに表示されたファイルはディスクに書き込まれません。その代わり、MapForce は、一時ファイルを作成します。「出力」ペインに表示されたファイルをディスクに保存するコは、メニューニマンド「出力|出力ファイルの保存」を選択するか、おけは「生成された出力を保存」(「」)ツールレーポタンをクトックします。

MapForce が一時ファイルではなく最終のファイルに直接的に出力を書き込むように構成するには、「ツール | オプション | 全般」 に移動し、「最終出力ファイルこファイルを書き込む」チェックボックスを選択します。チュートリアルを実行中は、故意にではなくもとの チュートリアルファイルを上書きする可能性があるため、このオプションを有効化することは奨励されません。

変換を行う生成されたXSLTコードもプレビューすることができます。コードをプレビューするコよマッピングペインの下にある「XSLT2」ボ タンをクリックします。





XSLT2 コードを生成しファイルに保存するには、メニューアイテム「ファイル | コード生成 | XSLT 2.0」を選択します。プロンプトされると、 生成されたコードが保存されるフォルダーを選択してくたさい。コード生成が完了すると、デスティネーションファイルは、以下の2つのファイルを含 みます:

- 1. ターゲナスキーマンちなんで名づたれたXSLT 変換ファイル(このサンプルでは、MappingMaptolibrary.xslt)。
- 2. **DoTransform.bat** ファイル。 **DoTransform.bat** ファイルにより RaptorXML Server 内でのXSLT 変換の実行を有 効化することができます。(詳細に関しては <u>http://www.altova.com/ja/raptorxml.html</u> を参照してくたとい)。

2.2 複数のソースから1つのターゲットにマップ

前のチュートリアルでは、ソースファイル(books.xml)からデーダターゲ水ファイル(library.xml)に変換しました。ターゲ水ファイル (library.xml)はマピングの実行前は存在しておらず、マピングの変換により生成されました。 library.xml ファイル内にすでにデータ が存在すると仮定し、このデーダを books.xml から変換されたデータとマージするとします。 library.xml 空のデーダを books.xml か らのデータとマージすると仮定します。

このチュートリアルの目的は merged_library.xml とう名前のファイルを生成するマメピングをデザインすることです。生成されたファイル は、以下の2つのノースからのデーダを含みます: books.xml とlibrary.xml。ソースとして使用される両方のファイルは異なるスキーマを 持つことに注意してくたさい。ファイルを動的に処理と生成するで説明されているとおり、ソースファイルと同じスキーマを持つ場合、異なるアプロ ーチを使用してデーダをマージすることができます



チュートリアルの目的を達成するためコよ次のステップを踏みます。

ステップ 1: マパングデザインファイルの準備

このチュートリアルは、 <マイドキュメント > ¥Itova ¥Iap Force2021 ¥Iap ForceExamples ¥Iutorial ¥ フォルダーからの Books To Library.mfd マメピングを開始点として使用します。このマメピングは XML を新しいスキーマニ変換する チュートリアル内で作 成済みです。チュートリアルを開始するコよ、 Map Force で Books To Library.mfd ファイルを開き、新しい名前で保存します。

複数のファイルを参照するため、 <マイドキュメント> #Altova Map Force 2021 #Map Force Examples #Tutorial #フォルダー内に新しいマンプが保存されることを確認してくたさい。



BooksToLibrary.mfd (MapForce Basic Edition)

ステップ 2:2番目のノースコンポーネントの作成

最初にターゲナコンポーネントを選択して、(「Ctrl + C」を押してコピーし、同じマピング内に「Ctrl + V」を押してい貼り付けます。新しいコンポーネントのヘッダーをbooks コンポーネントの下にドラッグします。



マピングロセンのノースエンポーネトと books とlibrary、と1つのターゲトコンポーネトかります: library。

マピングエポーネトを4つの左、右、上、下)方向に移動することができます。しかしなから、ソースエンポーネトをターゲトエンポー ネトの左に配置することことり、マピングの作成を簡単にし、おこ、他のユーザーにとし読み込みやすく理解しやすくなります。このド キュメントで説明されている全てのマピングにもこの規則は適応されており、おこ、MapForce インストールに伴うサンプルマピング ファイルにも同様の規則が適用されています。

ステップ 3: 入力/出力ファイルの検証とセナ

前のステップでは、新しいソースコンポーネントはターゲットコンポーネントからコピーし、貼り付けられましたので同じ設定を継承します。名前入 カ/出カインスタンスファイルが正確に設定されるように、各コンポーネントのヘッダーをダブルクリックし、コンポーネント設定ダイアログボックス内 で、下に表示されるように各コンポーネントの入力/出力ファイルを検証し、名前を変更します。

🕑 コンポーネント設定	×
コンポーネント 名: books	
スキーマ ファイル(F)	
books.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
books.xml	参照(B) 編集(E)
出力 XML ファイル(O)	
	参照(R) 編集(D)

最初のソースのコンポーネント設定 (books)

🕑 コンポーネント設定	×
コンポーネント名: library	
スキーマ ファイル(F)	
library.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
library.xml	参照(B) 編集(E)
出力 XML ファイル(O)	
	参照(R) 編集(D)

2番目のソースのコンポーネント設定 (library)

🕑 コンポーネント設定	×
コンポーネント名: merged_library	
スキーマ ファイル(F)	
library.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	券昭(n)
出力 XML ファイル(0)	
merged_library.xml	参照(R) 編集(D)

ターゲオのコンポーネント設定 (merged_library)

上に表示されるように、最初のノースエンポーネントはbooks.xmlからデータを読み込みます。2番目のノースエンポーネントは library.xmlからデータを読み込みます。2番目のノースエンポーネントはmerged_library.xmlとらファイルのターゲナコンポーネント 出力データを最後に読み込みます。

ステップ 4: 接続の作成

MapForce に2番目のノースからターゲトにデーダを書き込むようご命令するには、ソースlibrary 内エレポーネトのpublications アイテムの出力コネクタ(小さな三角形)をクリックし、ターゲト library コンポーネント内のアイテムの入力コネクタに、ラッグします。ターゲト 入力コネクタには既に接続かあり、次の通知メッセージが表示されます。

MapForce
接続しようとしている入力コネクタには既にコネクタが割り当 てられています。1つの入力コネクタに対しては1つの接続に しか割り当てることができません。希望する動作を選択してく ださい。
接続を置換え(R) 入力を複製(D) キャンセル

このチュートリアルでは、接続の置換えか目的ではなく、目的は2つのノースからのデータのマッピングです。ですから「入力の複製」をクリックします。これを行うことにより、ターゲトコンポーネントに新し、ツースからのデータを受け入れるように構成することができます。マッピングは、以下のよう」ないます。



ターゲトコポーネト内のpublication アイテムが複製されまた。新規 publication (2) ノードはノース library コンポーネントからのデーダを受け入れます。更に重要な点は、マピング内でこのノードは publication (2) と表示されますが、結果 XML ファイル内での名前は、想定された目的である publication どいます。

マルングペインの下の「出力」ボタンをクリックして、マルシングの結果を確認します。library.xmlとbooks.xmlファイルのデータが、新規ファイルmerged_library.xml にマージされたことを確認してくたさい。

2.3 複数のターゲットスキーマとの作業

前のチュートリアルの、複数のノースから1つのターゲルトにマップ では、複数のノーススキーマから単一のターゲルスキーマへのデータのマッピン グについて作成方法について説明しました。2 つのノースからのブックのレコードを保管する merged_library.xml と呼ばれるファイルを作 成しました。他の部署からこのXML ファイルのサブセルを提供するように要請かあオーと仮定します。特に、1900年以降に出版された書 籍を含むXML ファイルを提出するとします。

既存のMultipleSourcesToOneTarget.mfd マピングを使用し、完全なXML ライブラル・フィルターされたライブラルをに応じて、を生成することができます。変更を加えることができます。



データ変換の抽象的なモデル

上の図では、データは2 つの異なるスキーマ(books.xsd とlibrary.xsd)がmerged_library.xml と呼ばれる単一のXML ファ イルニマージされます。第2に、データはフィルター関数を使用して変換され、filtered_library.xml と呼ばれる XML ファイルを作成する、 次のエレポーネントに マンされます。「中間」コンポーネントは、データターゲットとノースとしての役割を果たします。MapForce では、このテク ニックオゴチェーシマッピング」と呼ばれ、このチュートリアルの主な内容です。ます。

このチュートリアルのゴールは、merged_library.xmlとfiltered_library.xmlを必要に応じて生成することです。チュートリアルの目的を達成するためには、次のステップを踏みます。

ステップ 1: マッピングデザインファイルの準備

このチュートリアルは、 <マイドキュメント> Altova Map Force2021 Map ForceExamples ¥ utorial ¥ フォルダーからの MultipleSourcesToOneTarget.mfd マナピングを開始点として使用します。このマナピングは<u>複数のノースから1 つのターゲットにマッ</u> <u>プ</u>チュートリアル内で作成済みです。チュートリアルを開始するコよ、Map Force で MultipleSourcesToOneTarget.mfd ファイ ルを開き、新しい名前で保存します。

複数のファイルを参照するため、 <マイドキュメント> #Altova #MapForce2021 #MapForceExamples #Tutorial #フォルダー内に 新しいマッピングが保存されることを確認してくたさい。



MultipleSourcesToOneTarget.mfd (MapForce Basic Edition)

ステップ2:2番目のターゲトコンポーネントの追加と構成

2番目のターゲトコンポーネントを追加するコよ「XML スキーマ / ファイルの挿入」(
望) ツール デボタンをクトックし、 <マイドキュ メント > ¥Altova ¥Map Force 2021 ¥Map Force Examples ¥Tutorial ¥ フォルダーニある library.xsd ファイルを開きます。サンプルイ ンスタンスファイルを提供するようにプロンプトされた場合、「スキップ」をクトックします。マンピングは、以下のようコントはます。



上に表示されるように、マピングには2つのノースエンポーネントと books とlibrary、と2つのターゲトコンポーネントがあります。ターゲットコンポーネントを区別するために、2番目のターゲトコンポーネントをとfiltered_libraryに名前を変更し、生成される XML ファイルの 名前も変更します。これを行うコよ、右端にあるコンポーネントをダブルクトックして、コンポーネントの設定を次のように編集します。

64 チュートリアル

🕑 コンポーネント	没定		—×
コンポーネント名:	filtered_library		
スキーマ ファイル(F)			
library.xsd		参照(W)	編集(T)
- 入力 XML ファイル(I))		
		参照(B)	編集(E)
出力 XML ファイル(O)			
filtered_library.xml		参照(R)	編集(D)

コポーネトの新しい名前はfiltered_librar どぶり、出力 XML ファイルコはfiltered_library.xml とら名前が付けたれます。

	😫 filtered_library	
	早 口ファイル : filtered_library.xml <u>ファイル</u> の	ł
Ľ	> [່] ເ⊐ຸ () library ເ	ł
Ľ	> 🕂 🗘 last_updated 🛛 🖓	ł
Ľ	> ^ເ ອ () publication ເ	ł
ľ	> 🕂 🗘 id ເ	þ
Ľ	> 🕂 🗘 author ເ	ł
Ľ	> 🕂 🗘 title 🛛 🕹	ł
Ľ	> 🕂 🗘 genre ເ	ł
G	> 🔷 Opublish_year 👔	P

ステップ 3: 接続の作成

merged_library内のアイテムpublicationからfiltered_library内のアイテムpublicationへの接続を作成します。これを行う際、通知メッセージが表示されます。

MapForce
マッピング プロジェクトにて複数の対象になるコンポーネントを作成しま した。特定のターゲット コンポーネントをプレビューするには、コンポー ネントのタイトルバーにあるプレビューボタンをクリックして、XSLT ま たは出力タブをクリックして結果をプレビューしてください。
■ 次回からこのメッセージを表示しない
OK

「OK」をクリックします。次の新規の†なっかターゲナコンポーネントの右上に表示されます:「プレビュー」(💁)と「ノ ススルー」 (Đ)。 次のステップでこれらの†なくは使用され、説明されます。



ステップ 4: データのフィルター

filtered_libraryに提供する前にデータをフィルターするコよ「フィルター」コンポーネントを使用します。フィルターコンポーネントを追加す るコよ merged_libraryとfiltered_library、間の接続を右クリックして、コンテキストメニューから、「node/row フィルターの挿 入」を選択します。



フィルターコンポーネントがマッピングに追加されました。



上に表示されるように、ブール入力コネクタは、入力が必要なことを示すオレンジ色にマイライトされます。コネクタ上にマウスをポイントすると、 入力の型 xs:boolean が必要なことがわかけます。ヒントを表示するコよ「ヒントの表示」(
) ソール・オキンの有効化されている必要があることに注意してくたさい。



フィルターコンポーネントはtrue おけまfalse を返す条件を必要とます。ブール型の条件は、現在のpublication シーケンスのデータ がターゲットにコピーされるとtrue を返します。条件が false を返すと、データはコピーされません。

このチュートリアルで使用される条件は1900年以降に出版された書籍をフィルターします。条件を整えるために、、以下を行います:

1. 1900 の植を持つ定数の数値の型を追加します。(「挿入」メニューから「定数」をクリックします)。「数値」を型として選択します。

ご 定 数 を 挿 入		×
1900		● 文字列(S) ● 数値(N) ● その他(O)
	ОК	キャンセル

- 2. ライブラリウイドウ内で、関数 greater を検索し マメピングペイン ボラッグします。
- 3. greater 関数へとgreater 関数から以下に表示されるように、マンピングを接続します。これを行うことしより、MapForce に以下を行うようこ命令します:「publish_year(出版年)が1900 以降の場合は、現在のpublication ソースアイテ ムをpublication ターゲットアイテム」にエピーします。



ステップ 5: 各ターゲナコンポーネントの出力をプレビューし保存する

ターゲトコンポーネトの出力をプレニーし保存することができます。複数のターゲトコンポーネトが同じマピングに存在する場合、「プ レビュー」(

) ボタンをクトックすることにより、プレビューの対象を選択することができます。「プレビュー」ボタンが押された状態(

)の場合、特定のコンポーネントが現在プレビューのために有効化されていることを示します(そして、この特定のコンポーネントが「プレビュー」 ペイン内に出力を生成します)。一度にコンのコンポーネントのみがプレビューのために有効化することができます。

ですから、merged_library(すなわち、「中間」)コポーネトの出力を確認し保存する場合は、以下を行います。

- 1. merged_library コポーネトの「プレビュー」 ボン(🚾)をクックします。
- 2. 「出力」ボタンマナングペインの下のをクリックします。
- 3. 出力をファイルに保存するコよ「出力」メニューから出力ファイルの保存」をクリックします。

filtered_library コンポーネトの出力を表示して保存するコよ以下を行います:

- 1. merged_lilbrary コポーネト上のパススレー」ボタン(🖶)をクルクします。
- 2. filtered_library コンポーネト上のプレビュー」ボタン(🚾)をクリックします。
- 3. マピングペインの下の出力」ボタンをクリックします。
- 4. 出力をファイルに保存するコよ「出力」メニューから出力ファイルを保存」をクリックします。

「パススルー」(

) ボタンー をクリックませまクリックしてい ことしより、複数のターゲトコンポーネントを持つマッピングに大きな違いをせたら します。このドダンが押された状態(

) である場合、MapForce は、データコ中間コンポーネントをパススルーさせ、マッピング全体の残 りをプレビューすることができます。

merged_libraryとfiltered_library間のマピングの部分のみをプレビューするコはボタン(🛃)をリリースします。後者の場合、 エラーか生成されます。中間コンポーネントはデータを読み込む、き妥当な入力 XML ファイルを持たなし すっかこの振る舞い は予期されます。 す。問題を解決するコよ、コンポーネントのヘッダーをダブルクリックし、編集して、妥当な入力 XML ファイルを提供します。

🕑 コンポーネント設定	×
コンポーネント 名: merged_library	
- スキーマ ファイル(F)	
library.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
library.xm;	参照(B) 編集(E)
出力 XML ファイル(O)	
merged_library.xml	参照(R) 編集(D)

複数のターゲオ コンポーネントを持つマッピングのデザインカ終わると、このチュートリアルの目的である、各ターゲナトの出力を確認して保存 することができます。ノウスルーコンポーネントに関しての詳細は、チェーンマッピング/ノウスルーコンポーネントを参照してくたさい。

2.4 ファイルを動的に処理と生成する

このチュートリアルは、複数のノースXML ファイルからデータを読み込み、同じ変換内で複数のターゲットファイルに書き込む方法を説明します。このテクニックを説明するために、次の目的を持つマッピングを作成します。

- 1. 同じディレクトリ内の複数のXML ファイルからのデータを読み込む。
- 2. 新規のXML スキーマに各ファイルを変換する。
- 3. 各ソースXML ファイルのために、新しいスキーマの下で新規ターゲナファイルを生成する。
- 4. 生成されたファイルからXML と名前空間宣言を削除する。



データ変換の抽象的なモデル

3つのノースXML ファイルをサンプルとして使用します。ファイルの場所はくマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\フォルダーで確認することができ bookentry1.xml、 bookentry2.xml, とbookentry3.xml と ら名前が付けられています。これらのファイルはそれぞれ単一のブックを保管します。



bookentry1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
<book id="2">
<author>Franz Kafka</author>
<title>The Metamorphosis</title>
<category>Fiction</category>
<year>1912</year>
</book>
</books>
```

bookentry2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
<book id="3">
<author>Herman Melville</author>
<title>Moby Dick</title>
<category>Fiction</category>
<year>1851</year>
</book>
</book>
```

bookentry3.xml

ソースXML ファイルは次のフォルダーで使用することのできる books.xsd スキーマを使用します: くマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\。ソースファイルを新規 XML スキーマニ変換するはよ(同じフォル ダーで見つけることのできる) library.xsd スキーマを使用します。変換の後、マメングは3つのファイルを新しいスキーマロゴに 生成します (下のコードリストを参照)。生成されたファイルが以下のようになるよう、マメングを構成します: publication1.xml、 publication2.xml、とpublication3.xml。XML 宣言と名前空間が削除されなければならない 点に注意してくたさい。

```
library>
<publication>
<id>1</id>
<author>Mark Twain</author>
<title>The Adventures of Tom Sawyer</title>
<genre>Fiction</genre>
<publish_year>1876</publish_year>
</publication>
</library>
```

publication1.xml

```
library>

<publication>

<id>2</id>

<author>Franz Kafka</author>

<title>The Metamorphosis</title>

<genre>Fiction</genre>

<publish_year>1912</publish_year>

</publication>

</library>
```

publication2.xml

publication3.xml

チュートリアルの目的を達成するために、次のステップを踏みます。

ステップ 1: マゾングデザインファイルの準備

このチュートリアルは、開始のポイントとしてくマイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥Tutorial¥フォル ダーからBooksToLibrary.mfd マピングを使用しています。すでに<u>XMLを新しいスキーマー変換する</u>チュートリアル内でこのマピン グをデザインしています。開始するコよ、MapForceでBooksToLibrary.mfdファイルを開き、新しい名前で同じフォルダービ保存し ます。



BooksToLibrary.mfd (MapForce Basic Edition)

ステップ2:入力を構成する

MapForce に複数のXML インスタンスファイルを処理するように命令するコよ、ソースエンポーネトのヘッダーをダブルクトックします。エンポーーネト設定ダイアログボックス内に入力ファイルとして bookentry*.xml を入力します。
	×
コンポーネント名: books	
ースキーマ ファイル(F)	
books.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
bookentry*.xml	参照(B) 編集(E)
出力 XML ファイル(O)	
	参照(R) 編集(D)

コンポーネント設定 ダイアログボックス

ファイル名内のアスタリスク(*) ワイルドカード文字は、マンピング入力にbookentry- プレフィックスを伴う全てのファイルを使用するように 命令します。パンは相対的であるため、MapForce はマンピングファイルと同じディレクトリ内の全てのbookentry-ファイルを検索しま す。マンピングファイル * ワイルドカード文字を使用しているため、必要であれば、絶対パンを入力することもできます。

ステップ 3: 出力を構成する

各出力ファイルのファイル名を作成するコよ concat 関数を使用します。この関数は引数に与えられた全ての値を連結します。

concat 関数を使用してファイル名を作成する

concat 関数をライブラリウイドウで検索し、マピングエノアドニッグします。デフォルトでは、この関数はシラメータか2つあるマピングに追加されます。しかし、必要であれば新しいシラメーターを追加することができます。関数コンポーネント内の「パラメーターの追加」(□)シンボリをクリックして、3番目のシラメーターを追加します。「パラメーターの削除」(図)シンボリをクリックするとシテメーター(排)除されることに注意してくたさい。

	<u>f</u> conca	it
Ľ	≥value1 ⊠	
Ľ	value2 🛛	result[;
Ľ	value3 ⊠	

2. 定数を入力するコよ(「挿入」メニューから「定数」をクトックします)。値を提供するようにプロレプトされると、「publication」 を入力して、文字列 オプションを変更しないでそのままにしておきます。

		×
publication		● 文字列(S) ○ 数値(N) ○その他(O)
	ОК	キャンセル

3. concat 関数のvalue1 と定数をと接続します。



4. ソースコンポーネトのid 属性を concat 関数の value2 と接続します。



5. ライブラ・ウイドウトで get-fileext 関数 を検索し、マレングエリアイボラッグします。ソースエンポーネント(File: books.xml) のトップノードからこの関数のfilepath / ウメータへ接続を作成します。get-fileext 関数の結果から concat 関数のvalue3 への接続を作成します。これを行うことにより、ソースファイル名から拡張子の部分(この場合、.xml)を 抽出することができます。



concat 関数の デメーターに、連結されると生成されるファイル名 (例: publication1.xml)を作成する3つの値を提供しました

パート	サンプル
定数「publication」は定数文字列の値「publication」を与えます。	publication
ソースXML ファイルの属性 id は 各ファイルのナメの一意の識別子の値 を与えます。これは全てのファイルが同じ名前で生成されないようにするナメかで す。	1
get-fileext 関数は、生成されるファイルの拡張子を返します。	.xml

MapForce にマンピングが実行される際に実際にファイル名を作成するように命令することができます。これを行うには、「ファイル」(「iel)) をクリックする、おけは ターゲトコンポーネントの「ファイル/文字列」(「iel/String」)、ポタンから「マンピングから与えられた動的なファイル 名を使用する」を選択します

覺 library ▶日□ファイル:<動的>	ファイ	
Clibrary		コンポーネント設定からファイル名を使用する(F)
■ O publication	~	マッピングにより与えられた動的なファイル名を使用する(D)
>Oid >Oauthor		XML に文字列を解析する(P)
title		文字列にXMLをシリアル化する(S)
publish_ye	ar	

マピングにより与えられた名前を伴うMapForceにインスタンスファイルを動的に生成するように命令しました。この特別な例では concat 関数により名前が作成されました。ですから、concat 関数の結果とターゲナコンポーネントのファイル く動的> ノードを接続 します。

Concat value2 @ result value3 @ () library () library () library () library () library () last_updated () publication () id () author () itle () genre () publish_year	
イル」テキストボックスが無効化されており、値が〈マッピングにより与えられたファイル名〉を示し 気がつくはずです。	ていることに
コンポーネント名: library	
スキーマ ファイル(F)	
library.xsd 参照(W) 編集(T)	
入力 XML ファイル(I)	
<マッピングにより与えられたファイル名> 参照(B) 編集(E)	
出力 XML ファイル(O)	
<マッピングにより与えられたファイル名> 参照(R) 編集(D)	

これは、インスタンスファイル名を動的にマッピングから提供したことを示し、コンポーネント設定内で定義することは関連性がありません。

最後に、ターゲナからXML名前空間とスキーマ宣言を削除する必要があります。これを達成するコよコンポーネト設定ダイアログボック スの「スキーマ /DTD レファレンスの追加…」と「XML 宣言の書き込み」チェックボックスから選択を解除します。

□ スキーマ/DTD参照を追加(スキーマの絶対パスを使用する場合はフィールドを空に)(A):
 □ XML 宣言の書き込み(X)

結果と生成されたファイル名を確認するために、マッピングを実行します。このマッピングは複数の出力ファイルを生成します。出力ペインの左上の左と右ボタンを使用して出力ファイル全体を移動することができます。ませよ横にあるドロップダウンリストからファイルを選択します。

4	プレビュー 1/4 📓 C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Tutoria\publication1.xml 🚽
1	c:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Tutorial\publication1.xml
2	last_updated>2015 C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Tutorial\publication2.xml
3	olymptication> cids1 C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Tutorial\publication3.xml
5	<authors c:\users\altova\documents\altova\mapforce2015\mapforceexamples\tutorial\publication4.xml<="" mark="" p="" tw=""></authors>
6	<title>The Adventures of Tom Sawyer</title>
7	<genre>Fiction</genre>
8	<publish_year>1876</publish_year>
9	
10	

3 一般的なタスク

このセクションは、マピングとの作業、コンポーネント、接続、マピングプロジェクトなどのMapForceの全般的なタスクとコンセプトを説明します。

3.1 マッピングとの作業

MapForce マピングデザイン(おけよ 略して「マピング」)は データがどのようコ つのフォーマトから他のフォーマトに変換さえるかの 視覚的な表現です。マピングは、データ変換を行うせの MapForce マピングエルア内で1 つのスキーマから他のスキーマに追加するユン <u>ポーネト</u>から構成されています。(例: XML ドキュメントを1 つのスキーマから他のスキーマに変換)。有効なマピングは、1 つまけは複数のタ <u>ーゲナコンポーネント</u>に接続されている、1 つまけは複数の複数の<u>ノースエンポーネント</u>から構成されています。マピングを実行して、結果を 直接 MapForce 内で確認することができます。コードを生成し、外部で実行することも可能です。 MapForce 実行可能ファイルにマピ ングをエンパイルし、 MapForce Server おけま FlowForce Server を使用してマンピングの実行を自動化することもできます。 MapForce は、マピングを、mfd の拡張子を持つファイルとして保存します。

新規マッピングの作成方法:

- 1. 以下の内のしつを行ってくたさい
 - 「ファイル」メニューカジ新規作成」をクリックします。
 - 「新規作成」(□)ツールシーオタンをクリックします。
- 2. 「マッピング」をクリックして、, and then click「OK」をクリックします。

新規ファイル	×
	OK キャンセル
標準 MapForceマッピング作成	

マルングが作成されまた、しかしながら、空のため何も起こりません。最低でも妥当なマルングはつの接続された、エンパーネートを必要と次のステップはマルングにエンパーホートを追加し接続線を描きます。

3.1.1 マッピングにコンポーネントを追加する

MapForce 内では、「ンポーネト」とう用語はデータの構造(スキーマ)おけボータがどのようご変換(関数)されるかを視覚的に表します。 す。 ンポーネトはすべての<u>マノピング</u>を構築する中心的な役割を果たします。 マノピングエリアでは、コンポーネトは正方形の枠で表示され ます。 以下は、MapForce コンポーネトの例です:

- 定数
- データベース
- フィルター
- 条件
- 関数コンポーネト
- EDI ドキュメナ (UN/EDIFACT、ANSI X12、HL7)
- Excel 2007+ ファイル
- 単純型 <u>入力工ポーネト</u>
- 単純型 出力コンポーネント
- スキーマおよびDTD

X

マッピングにコンポーネントを追加するには以下を行います

- 「挿入」メニューから追加するエンポーネトの型(例「XML スキーマ/ファイル」をクトックします。
- Windows ファイルエクスプローラーからファイルをドラッグしてマッピングエリアイドロップします。このオペレーションは互換性のあるファ イルベースのエレポーネントのみに対して使用することができます。
- コンポーネトの挿入ツール、一内の適切なボタンをクトックします。

コンポーネントを挿入 41 🕞 💵 🖳 🎇 🎬 🎬 🎬 🎼 🧔 🍫 💠 ਇ 🍙 🧟 🔭 🗎 🖬 🏘 🏘 🔚 🖬 🕇 🌇 🍋 🖬

コンポーネントの挿入ツールレー(MapForce Enterprise Edition)

それぞれのエレポーネントの型には、特定の目的と振る舞しめあります。コンポーネントの型が必要な箇所では、MapForce は、コンテキストウィザードステップおけよダイアログボックスで手順を踏んで、説明を行します。例、XML スキーマを追加する場合、通知ダイアログボックスが インスタンスファイルをオプションで選択するように促します。

コンポーネトについての説明は、コンポーネトとの作業を参照してくたさい。マンシグソースおけはターゲトとしてサポートされる個々の技術に関しては、データノースとターゲルを参照してくたさい。データを一時的に保管おご変換する MapForce ビルドインコンポーネントに関しては、おけはブノルタレグおけは並べ替えなどの)、マンピングのデザインを参照してくたさい。

3.1.2 URL からコンポーネントを追加する

マピングエポーネントとしてローカルファイルを追加するだけでなく、URLからファイルを追加することができます。この操作はエレポーネントを ソースエンポーネントとして追加した場合のみサポートされます。(すなオム、マピングは、リモートファイルからデータを読み込みます)。サポー トされるプロトコルはHTTP、HTTPS、とFTPです。

URL からコンポーネントを追加する

- 1. 「挿入」メニューから追加するコンポーネトの型を選択します(例 XML スキーマ/ファイル)。
- 2. 「開く」ダイアログボックスから「URL にスイッチ」をクリックします。

🕑 ファイルを開く		×	
ファイルの場所(I):	📜 MapForceExamples 🗸 📀 🍺 📂 🖽 🗸		
C.	名前	更新日時	
	📜 Tutorial	4/5/2016 3:37	
最近衣示した場	📾 837-Q2.xsd	1/25/2016 8:1	
7/1	address.xsd	1/25/2016 8:1	
	Altova_Hierarchical.xml	1/25/2016 8:1	
デスクトップ	Altova_Hierarchical.xsd	1/25/2016 8:1	
	Altova_SQLXML.xsd	1/25/2016 8:1	
F	altova-cmpy.xml	1/25/2016 8:1	
	altova-cmpy-extra.xml	1/25/2016 8:1	
ライブラリ	AltovaTools.xml	1/25/2016 8:1	
	Altova Tools.xsd		
		• •	
コンピューター	ファイル名(N): マー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	開(0)	
	ファイルの種類(T): 全てのファイルタイプ (*.xsd;*.DTD;*.xml) 🔹	キャンセル	
	URL にスイッチ (U) グローバル リソースにスイッチ (G)	.4	

3. 「ファイルURL」テキストボックスコファイルのURLを入力して、「開く」をクトックします。

開く	—
ファイル URL(U): http://qa-shp2013/Documents/BookList.xi	ml 🗸
- 開<形式: ◎ 自動(A)	ファイルのロード ③ 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 パスワードを保存
利用可能なファイル サーバーURL(V): Microsoft(R) SharePoint(R) Server	▼ 参照(B)
1	新規フォルダー(N)
ファイルダイアログにスイッチ(F) グローバル リソースにスイッチ(G)	解 キャンセル //

「ファイルURL」テキストボックス内のファイル型がステップ1で指定されたファイル型と同じであることを確認してくたさい。

サーバーが ペワード認証を必要とする場合、ユーザー名と ペワードを入力するよう こプロンプトされます。 次回 MapForce を起動する際 に使用できるように、ユーザー名と ペワードを保存する コよ、「開く」 ダイアログボックス内の「アプリケーション間で・ペワードを保存」 チェッ クボックスを選択します。

「開く形式」設定は、ファイルを開く際の、ーサーのための文法を定義します。デフォルトと奨励されるオプションは「自動」です。

アップロードするファイルに変更が加えられない場合、「キャッシュ/プロキシの使用」オプションを選択して、デーダをキャッシュし、ロードをスピードアップします。それ以外の場合、マンピングを開くたひ、ファイルを再ロードする場合は、「再ロード」を選択します。

Web分散オーサレグとデション管理(WebDAV)をサポトするサーバーは関しては、「サーバーURL」テキストボックスにサーバー URLを入力した後にファイルを「参照」をクリックして、参照することができます。参照は全てのファイルの型を表示しますが、上記のステップ 1で指定されたファイル型と同じ型が選択されていることを確認してくたさい。それ以外の場合、エラーが発生します。

サーバーがMicrosoft SharePoint Server の場合、「これは Microsoft SharePoint Server です」 チェックボックスをチェックし てください。これを行うことにより、プレビューエリア内のファイルのチェックインとチェックアナトの状態が表示されます。MapForce を使用中に他 のユーザーがファイルを編集できないようにするさっなはは、ファイルを右クトックして、「チェックアウト」を選択します。ユーザーにより以前 チェックアナトされたファイルをチェックするには、ファイルを右クトックして、「チェックアウト」を選択します。

Open	X
File URL: http:// /Documents/BookList.xsg	• t
Open as: Auto	load Use cache/proxy 💿 Reload
Identification User: Password: ••••••••••	Remember password between application starts
Available files Server URL: http://	■ Browse
☑ This is a Microsoft [®] SharePoint [®] Server	
Cache Profiles	A
in the second s	=
BookList.xml	
Booklist wed	
Empl Check In	-
Undo Check Out	New Folder Delete
Switch to File Dialog Switch to Global Resources	Open Cancel

開くダイアログボックス(URL に切り替える モード)

3.1.3 データストリーミングについて

データストリーミングとは任意に大きなデータノースをマピングの入力封さは出力として使用を許可するMapForce内蔵のメカニズムです。データストリーミングはMapForceにより生成されたコード内のストリームオブジェクトと間違わないように注意してくたさい。(後者はMapForceにより生成されたコードをカスタムのC#とJavaアプリケーションと統合するオメのデータの扱いを指します)。

データストリーミングは次のデータノースに適用することができます

- XML ファイル
- CSV ファイル
- 固定長 ファイル
- データベース

マピング内で上記のデータノースを入力おけ出力として持つ場合、MapForceは、メモルに全てのデータをロードするかわりに、データノースをデータのオープンストリームとして扱い、そのコンテンンを個別に処理します。

メモ BUILT-IN を変換言語とて選択している場合のみデータストリーミングの使用は可能です(変換言語の選択を参照)。

メモル使用の注意点

データストリーミング候補であるマンピング入力と出力と作業する場合、マンピングがランダムな入力ソースへのアクセスを必要とする場合、"メモリ不足"が発生する可能性があります。

例:マピングがノースデータニgroup-by 関数を適用するエンポーネントを含むと仮定します。group-by 関数を入力ファイルのソノー構造全体に適用した場合、メモルコードされるノールファイル全体が必要となり、この結果、ファイルストリーミングは使用できなくなります。並べ替えなどのマピングノース全体がシモルコードされることを必要とする操作にコお同様が適用されます。

上記のようなシチュエーションが発生した場合、十分な仮想メモルディスクスペースがシステム上に存在すると変換は完了します。

3.1.4 変換言語の選択

データ変換言語として次を選択することができます。

- XSLT 1.0
- XSLT 2.0
- XSLT 3.0
- XQuery
- C++
- C#
- Java
- BUILT-IN (MapForce のデフォルトのネイティブ変換エンジン)

変換言語を選択するコよ、以下を行います。

- 「出力」メニューから変換に使用する言語をクトックします。
- 言語選択ツール、一内の言語名をクトックします。

<u>XSLT XSLTa XSLTa XQ.</u> | JAVA C# C++ | Built +

マンピングの変換言語を変更する場合、MapForce機能の一部はその言語のためにサポートされたは場合があります。BUILT-IN は地の言語と比較して、MapForce機能の多くをサポートします。詳細に関しては、サポートメモを参照してくたさい。

BUILT-IN 変換言語

マピングの変換言語とて BUILT-IN I を選択すると、MapForce はオイティブの変換言語を使用してデータマピングを実行します。 選択された変換言語が Java、C#、ませまC++の場合、マピングの出力をプレビューすると MapForce はのオプロンを明示的に 使用します。

BUILTHN エンジイお外部プロセッサーを必要とすることよくマッピングを実行し、これはメモルの使用か考慮される場合に良いオプションです。 特定の言語でプログラムコードを生成する必要がよい場合、サポートメモで説明される通り、他の言語と比較して MapForce の多数の機 能をサポートするためBUILTHN をデフォルトのオプションとして使用します。更に、BUILTHN を変換言語として選択すると、MapForce Server を使用した実行のためにマッピングを使用することができます。更に、マッピングとMapForceの自動化を参照してくたさい。

3.1.5 マッピングの検証

MapForce は変換結果をパビューするけっと、マビングを自動的に検証します。結果をパビューする前 にマピングのみを検証することもできます。この機能は、マビングのエラーと警告のマビングを実行する前に認識し修正するけっというで ます。マピングの実行は、処理されたデータにより更なるランタイムエラーおけは警告を生成する場合があります。例属性にマップされた値の 上書きなど。

マピングを正確に検証するコよ以下を行います

- 「ファイル」メニューからマッピングの検証」をクリックします。
- 「検証」()ツール・オタンをクリックします。

「メッセージ」ウイドウは、検証の結果を表示します。例

メッセージ	→ ‡ ×
I I I A DOG 933 X	
<u>∽ □ ⊘ MultipleSourcesToOneTarget.mfd</u> : マッピング検証に成功 - 0 エラー、3 警告	
🔍 🖽 🛕 コンポーネント 🤱 <u>core.greater</u> : 入力 D <u>a</u> は接続が必要です。	
[の]田 🛕 コンポーネント 🔏 <u>core.greater</u> の入力コネクタ ▷a は接続されていなければなりません。	
▲コンポーネント ▼ <u>core_filter</u> : 出力は接続されていません。	
├──	

メッセージウィンドウ

マルングを検証するコよ、MapForce は、マルングの妥当性をチェックし、(無効ませま見つからない接続、サポートされないロンポーネントの種類などの)検証結果は、以下のステータスアイコンと共にメッセージケイドウニ表示されます。

アイコン	意味
0	変換は成功しまた。
<u> </u>	変換は警告と判定しまた。
8	変換は失敗しまた。

「メッセージ」ウイドウお以下のメッセージの型を表示する場合がおります:情報メッセージ、警告、とエラー。

アイコン	意味
•	情報メッセージを表示します。情報メッセージはマングの実行を停止しません。
	警告メッセージを表示します。警告はマメピングの実行を停止しません。警告メッセージは以下の場合に生成される 可能性があります。例: 必須の入力コネクターの接続が作成されなかった場合。このような場合、出力は、有効な 接続が存在する場所で、これらのコンポーネントのために生成されます。
0	エラーを表示します。エラーが発生すると、マンピングの実行に失敗すると、出力は生成されません。XSLT おけよ XQuery コードのプレビューを行うこともできません。

マピングエル内の情報、警告おけまエラーメッセージをトリガーしたエンポーネトおけば構造をハイライトするけま、メッセージケイドウカの下線のスサテテキストをクリックします。

デーダを変換するコンポーネントに関しては、(関数まとは変数など) MapForce 検証は以下のよう、動作します:

- 必須の入力コネクタが接続されていない場合、エラーメッセージが生成され、変換は停止されます。
- 出力コネクタ が接続されていない場合、警告が生成され、変換プロセスは続行されます。問題のあるコンポーネストとそのデータ は無視され、ターゲルドキュメントにマップされません。

個別のダブ内にそれそれの検証の結果を表示するコよ、メッセージウィドウの左横にある番号の付い、サーダブをクトックします。これはとても役に たちます。例:複数のマッピングファイルと同時に作業する場合。

メッセージウイドウ内の他の状なんことり以下のアクションを実行することができます。

- ハルージを型別にフィルターします(例: エラーおけは警告のみを表示)。
- エトーを上下に移動します。
- メッセージテキストをクリップボードにコピーします。
- ウィドウ内で特定のテキストを検索します
- メッセージウイドウをクリアします

メッセージウイボウの全般的な情報に関しては、ユーザーインターフェイスの概要を参照してくたさい。

3.1.6 マッピング出力の検証

「出力」、タブをクトックした後、マッピングをプレビューする場合は、「出力」、ペインで出力の結果を確認することができます。この出力を関連 するスキーマイズ・して検証することができます。例:マッピング変換がXML ファイルを生成する場合、結果 XML ドキュメントはXML スキー マイズ・して検証されます。

XML ファイルは関しては、コンポーネト設定ダイアログボックスの「スキーマのTD レファレンスの追加」フィールド内のインスタンスファイル に関連するスキーマを指定することができます(XML コンポーネト設定 を参照)。生成されたXML 出力によど参照されるスキーマファイル を、マスは指定します。これにより、マノニングの実行時、出力インスタンスか検証されることができます。このフィールドにhttp://、絶対おけは 相対・ツを入力することができます。「スキーマのTD レファレンスの追加」フィールドを選択しない場合、スキーマイズすする、出力ファイルの 検証は不可能です。このチェックボックスを選択し、空白のまましておくと、検証が終わると、コンポーネント設定ダイアログボックスのスキー マファイル名が出力に生成されます。

コンポーネント設定を開くには、以下を行います

• 「出力の検証」 「シール いちなをクリックします。

🕃 🕞 🖉 1 %?xml version="1.0" encoding="UTF-8"?> 2 PersonList xmlns:xsi="http://www.w3.org/2001/XML 3 <Person role="Manager"> 4 <First>Vernon</First> 5 <Last>Callaby</Last> 6 </Person> 7 <Person role="Programmer"> 8 <First>Frank</First>

• 「出力」メニューカジ出力ファイルの検証」をクトックします。

メモ 「出力の検証」ボタンと対応するメニューコマンド(「出力 | 出力ファイルの検証」)は、出力ファイルがスキーマイン対する検証 をサポートする場合のみ有効化されます。

検証の結果はメッセージウィドウ内に表示されます。例

🕝 <u>…\Tutoria\ExpReport-Target.xml</u>: 出力ファイルの検証が成功しました。 – O エラー、

検証に失敗した場合、メッセージは発生したエラーの詳細情報を表示します。

🕞 😵 C:¥Altova¥MapForce2012¥MapForceExamples¥Tutorial¥Tut-Person.xml: 出力ファイルは
- 🖸 🛙 要素 <name> 🎟 は要素 <company-person> 🚥 の下のこの位置で使用できません。</company-person></name>
回理由: 次の要素はこの位置で必要です。 (<u>下記参照</u>)
CompanyLogo> 1000
ニーエラーの場所: <u>Company-Person</u> / <u>Name</u>
│
品 <u>cvc-elt.5.2.1</u> : 要素 <u><company-person></company-person></u> 000 は、実際の型定義 '{anonymous' 1000

検証メッセージは更に詳し、詳細情報を含む多くの イパーレンを含みます。:

- ファイル マをクリックすると、MapForce の 出力」 タブの結果の出力が開かれます。
- 〈ElementName〉 レノカをクトックすると「出力」 タブ内の要素が イライトされます。
- WII アイコンをクリックすると、XMLSpy (インストールされている場合)内の要素の定義が開かれます。
- 詳細サブセクションの イパーレンズ例: cvc-model-group)をクリックすると、<u>http://www.w3.org/</u> W eb サイト上の対応す る検証のルールについての詳細が開かれます。

3.1.7 出力のプレビュー

MapForce マピングと作業する場合、結果出力を、外部のプロセッサーおけまコンパラーを使用して生成されたコードを実行おけまコン パルオることなく、プルビューすることができます。全般的には、生成されたコードを外部で処理する前に、MapForce内で変換出力をプレ ビューすることはは、考えです。

マピングの結果をプレビューする場合、MapForce は、マピングを実行して、結果出力を出力ペインに表示します。

出カペイノニデータが準備されると、必要に応じて検証し保存します(マンピング出力の検証を参照)。また、「検索」コマンドを使用して、(Ctrl + F キーの組み合わせ)簡単に特定のテキストの、ターンを出力ファイル内で検索することができます。

マピングの実行に関するエラー、警告、情報メッセージはメッセージウイドウに表示されます(ユーザーインターフェイスの概要を参照)。

変換の出力をプレビューする方法

- マピングウイボウ下の「出力」 ダブをクリックします。MapForce は 言語ツール・内で選択された変換言語を使用して マピングを実行し、結果の出力を使用して「出力」 ペインを作成します。
- メモ C++、C#、おさよJavaを変換言語とて選択した場合、MapForceはマングをビルドイの変換エンジンで実行します。「出力」ペインに表示される結果はJava、C++、おさはC#コードが生成、コンパイル、実行された状態と同じです。

変換の出力を保存するコよ、以下を行います

- 「出力」メニューから出力ファイルの保存」をクトックします。
- 「生成された出力の保存」ツール、ーボタンをクリックします。

部分的な出力のプレビュー

大きな出力ファイルをプレビューする場合、MapForce は、出力ペインに表示されるデータの量を制限します。具体的には、MapForce は、出力ペイン内でファイルの一部を表示します。そして、「更にロード…」、ボタインの下の部分に表示されます。「更にロード…」、ボタイン現在表示されているデータのファイルの部分の横に表示されます。

Result file size: 324.3 MB. 3% of the result are displayed.	Load more	Load all
---	-----------	----------

メモ 出力ペイン内にファイル全体がロードされると、「整形出力」ボタンの有効化されます。

オプションダイアログボックスの全般タブからプレビューの設定を構成することができます(MapForce オプションの変更を参照)。

3.1.8 テキストビューの機能

「出力」ペイン、「XSLT」ペイン、および「XQuery」ペインには、テキストをお簡単に表示するオーダの複数の視覚的な補助が搭載されています。以下のような機能が含まれます:

- <u>行番号</u>
- 構文の色
- <u>ブックマーク</u>
- ソースの折けさみ
- <u>インデオのガイド</u>
- <u>行末と空白文字のマーカー</u>
- <u>ズーム</u>
- <u>整形出力</u>
- <u>ワード ラップ</u>
- <u>テキストの イライト</u>

適用できる箇所では、上記の機能を「テキストビュー設定」、ダイアログボックスから切り替え、おけはカスタム化することができます。「テキスト ビュー設定」、ダイアログボックス内の設定は、アクティブルドキュメントだけではなく、アプリケーション全体に適用されます。

🕑 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 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	5		
◎ 次の選択に一致する:	▼ 文字		
▼大文字と小文字を区別			
キーマップ			
Up 1	行上へ		A
Down 1	行下へ		=
Left 1	Left 1列左へ		
Right 1	Right 1列右へ		
Ctrl+Up 1	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	Ctrl + Right 1単語右へ		
Shift + Up 上の行を選択			v
			11.

テキストビュー設定 ダイアログボックス

「テキストビュー設定」ダイアログボックスを開くには、以下を行います

- 「出力」メニューカジテキストビュー設定」を選択します。
- 「テキストビュー設定」 し、ノール いうかんたりかんます。
- 出力ペインを右クリックして、コンテキストメニューから、「テキストビュー設定」を選択します。

ナビゲーションの補助の一部はテキストビューソールバー、アプリケーションメニュー、キーボードショートカメカら切り替えることができます。

🖏 🖻 🚳 🖕

テキストビューツールバー

適用することのできるショートカナへの参照は、上記の「テキストビュー設定」、ダイアログボックスの「キーマップ」セクションを確認してください。

行番号

行番号は、「テキストビュー設定」ダイアログボックス内で切り替えることのできる行番号 マージンで表示されます。テキストのセグションが折り たまれると、折りたまれたテキストの行番号は表示されません。

構文の色

構文の色分けはテキストの構文の値に従い適用されます。例えば、XMLドキュメント内では、XMLノードが要素、属性、コンテンソ、 CDATA セグション、コメント、処理命令、ノード名(そして一部の場合ではノードのコンテンツにより異なる色分けが使用されています。

ブッケマーク

ドキュメント内のラインは参照とアクセスのオータリニブックマークとして使用することができます。ブックマークマージンオンに設定されていると、ブックマ 一クは、ブックマークマージンに表示されます。

	- •	
1		xml version="1.0" encoding="UTF-8"?
2		CompletePO xmlns:xsi="http://www.w3.org/200
3		Customer>
4		<number>3</number>
5		<firstname>Ted</firstname>
6		<lastname>Little</lastname>
7	۲	
8		<street>Long Way</street>
9		<city>Los-Angeles</city>
10		<zip>34424</zip>
11		<state>CA</state>
12		-
13		
14	۲	<lineltems></lineltems>
15		<lineltem></lineltem>
24		<lineltem></lineltem>
33		
34		🗢 <total></total>
35		<totalsum>595</totalsum>
36		<totalltems>2</totalltems>
37		-
38		CompletePO>
_		

それ以外の場合、ブックマークされたラインは水色で、イライトされています。



ブックマークマージンは、テキストビュー設定ダイアログボックス内で切り替えることができます。

ブックマークを以下のコマイドを使用して編集、および、移動することができます。

ブックマークの挿入/削除(Ctrl + F2)

次のブックマークへ移動(F2)

🝟 前のブックマークへ移動 (Shift + F2)

全てのブックマークを削除(Ctrl + Shift + F2)

上記の「出力」メニュー内で使用することができます。「出力」(おけよ「XSLT」おけよ「XQuery」)ペインを右クトックすると、コンテキストメニューを使用してブックマークコマンドを使用することができます。

ソースの折りたみ

ソースの折りオンみは、ノードの展開と折りオンム、およびノースの折りオンチマージン内に表示されるノードを表しています。テキストビュー設定 ダイアログボックス内でマージンを切り替えることができます。展開するコは、おけはテキストの一部を折りオナむコは、ウイドウの左側の「+」 と「-」ノードをクリックします。折りオナまれナコードは、省略記号と共に表示されます。これにより、プレビューされているコードを表示するとしト か開かれます。ヒントの終わりコは、省略文字が表示されています。

1	xml version="1.0" encoding="UTF-8"?
2	edited with XMLSPY v2004 U (http://www.xmlspy.com) by Mr. Nobody (Altova</th
	GmbH)>
3	Customers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:noNamespaceSchemaLocation="Customers.xsd">
4	
5	<number>1</number>
6	<firstname>Fred</firstname>
7	<lastname>Landis</lastname>
8	+ <address>\</address>
14	- Street>Oakstreet
15	
16	<number>2< <7ID>22200<!--7ID--></number>
17	<firstname> <ptata>MA</ptata></firstname>
18	

インデトのガイド

インデトのガイドは、ラインのインデトを示す垂直な点線です。「テキストビュー設定」ダイアログボックス内で切り替えることができます。

メモ 「挿入タブ」と「スペースの挿入」オプションは「出力 | XML テキストの整形出力」オプションを使用すると、効果が与えられます。

行末のマーカーと空白文字のマーカー

行末(EOL)マーカーと空白文字マーカーは、テキストビュー設定ダイアログボックス内で切り替えることができます。下のイメージは、行末と 空白文字のマーカーが表示されている状態を表示しています。矢印は、タブ文字を表し、「CR」は改行を表し、点は空白文字を表しています。

1	xml·version="1.0"·encoding="UTF-8"?
2	<pre>cbooks xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
	<pre>xsi:noNamespaceSchemaLocation="books.xsd"></pre>
3	$\ominus \longrightarrow \langle book \cdot id = "1" \rangle \blacksquare$
4	\longrightarrow \longrightarrow <author>Mark Twain</author> \mathbb{R}
5	→ <title>The Adventures of Tom Sawyer</title> CR
6	→→ <category>Fiction</category> C
7	\rightarrow
8	
9	CR

ズームインとズームアウト

「Ctrl キーを長く押して、スクロール(マウスのオイールをスクロール)することによりズームイン、とズームアウトを行うことができます。ませま、「Ctrl キーを押してから「-」、ませょ「+」キーを押してもズームイン、および、ズームアウトすることができます。

整形出力

「XML テキストの整形出力」コマンドは、アクティブな XML ドキュントを構成された形式で表示するなかに、テキストビュー内で再フォーマナします。デフォルトでは、各子ノードが親ノードより、つの空白文字によりオフセナで表示されます。これは、「テキストビュー設定」 ダイ アログボックスによりカスタム化することができます。

XMLドキュメイを整形出力するゴよ「出力 | XML テキストの整形出力」メニューコマイを選択、まゴは「整形出力」

ワードラプ

現在アクティブなドキュメント内で、ワード ラップを切り替えるコは、「出力|ワードラップ」メニューコマンドを選択、まけは「ワードラップ」

テキストのノイライト

テキストを選択すると、ドキュメント内で選択されたテキストに一致する箇所が自動的にフイライトされます。選択箇所は薄い青で、イライトされ、一致する箇所は薄いオレンジで、イライトされます。選択箇所と、一致する箇所は、灰色のマーカーによりスクロール、一上に表示されます。現在のカーノルの位置は、スクロール、一上で青いマーカーにより表示されています。

テキストの イライトをオノニするコはテキストビュー設定 ダイアログボックス内の「自動・イライトの有効化」を選択します。選択範囲は、 文字全体、おココ固定された文字数に設定することができます。大文字と小文字の区別も指定することができます。

文字の選択範囲は、選択範囲の文字の最初の文字から開始する、一致する文字の最小数を指定することができます。例えば、2つ、おこ は2文字以上の文字が一致するように選択することができます。この場合、1つの文字の選択は、一致とは考えられません。2文字以上の一 致のみが一致と考えられます。ですから、この場合、モーを選択すると、一致は表示されません。 ty を選択すると、ty に一致するが表示さ れます。 typ を選択すると、typ に一致するすべてが表示されます。

文字の検索に関しては、文字を区切るために以下か考慮されます:(角かっこのは、)要素名、要素タグの角かっこ、属性名、引用符無しの属性の値。

3.1.9 テキストビュー内の検索

「出力」ペイン、「XQuery」ペイン、および「XSLT」ペイン内のテキストは、広範囲のオプションの組み合わせと視覚的な補助を用いて、検索をすることができます。

検索を開始するコよ「Ctrl+F」を押します(お」はメニューコマンド「編集 | 検索」を選択します)。ダイアログ内に入力された検索用語を、ドキュメント全体で、お」よ、選択された範囲で検索することができます。

- 検索する文字列を入力、およ、コンボドックスを使用して最近使用した10件の文字列を選択します。
- 検索する文字列を入力、およま選択すると、全ての一致が イライトされ、 スクロール 、一内で一致が ージュ色で表示されます。
- 現在選択されている一致は、他の一致と異なる色で表示されます。一致の位置はスクロールレー上で濃い青のカーノルマーカーで表示されます。
- 検索用語フィールド内に一致の総数が現在選択されている一致のインデックス位置と共に表示されます。例えば 2 の 4 は、 4つ中の2番目の一致が選択されていることを示しています。
- 右下の前へ (Shift+F3) と次へ (F3) ボタンを選択することことり、1つの一致から次の一致へ両方向に移動することができます。



• 検索ダイアログを閉じるコよ 右上の閉じる × ポタンをクリック、まけよ Esc を押します。

以下の点に注意してくたさい

- 検索ダイアログバニモードは存在しません。これは、検索ダイアログは、テキストビューを使用する場合でも、開き続けることができます。
- ダイアログボックスを開く前にテキストが選択されている場合、選択されたテキストは、自動的に検索用語フィールドに挿入されます。
- 選択範囲内で検索を行うコよ、以下を行います:(i)選択範囲をマークします(ii)選択範囲内で検索
 オプシュンをオンにして、検索範囲をロックします。(iii)検索用語を入力します。他の選択範囲内を検索するコよ、現在の選択範囲を、選択範囲のを検索
 オプションをオスユてアンロックします。そして、新規の選択範囲を選択範囲内を検索
 オプションをオスユてアンロックします。

• 検索ダイアログが閉じられると、現在の検索はF3を押すことには順方向検索し、Shift+F3を押すことには逆方向検索します。検索ダイアログが表示されます。

検索オプション

検索の条件は、検索フィールドの下にあるがタノこと、指定することができます。オプションがオノフなっている場合、ボタンの色は青に変更されます。以下のオプションから選択することができます:

オプション	アイコン	説明
大文字と小文字を区別す る	Aa	切り潜えられると、大文字と小文字を区別する検索が行われます(「Address」は「address」 とは異なります)。
単語単位で検索	Abc	テキスト内の文字のみが一致されます。例えば、入力文字列 fit に対して、単語単位を一致 する がけくされていると、単語 fit のみか検索文字列に一致します。fitness 内のfit は一 致しません。
正規表現	*	オノこ切り替えられると、検索用語は、正規表現として読み取られます。「正規表現の使用」を 参照してくたさい。
アンカーの検索	Ξ	検索用語が入力されると、ドキュメト内の一致が、イライトされ、一致の内の1つが現在の選択 と、てマークされます。アンカーの検索の切り替えは、最初の選択がサーノルの位置に対して相 対的かを決定します。アンカーの検索がオンゴリ「替えられると、現在選択されている一致が選 択され、現在のカーノルの場所の次の一致に一致します。アンカーの検索がオフェブリ、替えられ ていると、現在選択されている一致」は、キュメントの最初から数えて最初の一致が一致します。
選択範囲内の検索	Llt	オイニ切り替えられると、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ニ切り替えて現在の選択範囲のロックを解除します。

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。これを行うけは、最初に 正規表現 🌁 オプノョンをオノゴ切 り替えます。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: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カニ、検索用語フィールドの右にある正規表現ビルダー 💐 ボタンをクリックします。これにより、検索用語フィールド内のテキストが正規表現として評価されます。下のスクリーンショットは、電子メールアドレスを検索するオカの簡単な正規表現を表示しています。



正規表現メタ文字のカスタムセナは、テキストを検索し置き換える際にサポートされます。

	任意の文字を一致する。これは単一の文字のプレースヤレダーです。
(abc)	(and) メタ文字は、タグ付け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ルファレンス) 目的のためこタグ、「記録」)する場合、役に立つ可能性があります。9個までのサブ式をタグ付け(そして後から参照)する ことができます
	例えば、(the) \1 は、文字列 the 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:以前に一致したタグ付けされた箇所が後に続き、スペース文字が後に続く、文字列 「the」を一致、(および、タグされた箇所として記録します。
∖n	n が1 から9 の箇所では、n は、最初から9番目のタグ付けされた箇所を指します(上を参照してくたさい)。
\<	単語の先頭に一致。
\>	単語の末尾に一致。
Λ	バックスラッシュが後に置かれている文字をエスケープします。すなわち、式 ¥ は、文字 x を文字通い使用することができます。例えば、\[は、文字のセナの開始としてではなく、[として解釈されます。
[]	このセナ内の文字に一致します。例えば [abc] は a, b ませま c の文字に一致します。範囲を使用することができます。例えば、小文字のために [a-z] を使用します。
[^]	このセナ内では内文字に一致します。例えば [^A-Za-z] は アルファベナ文字以外の文字に一致します。
^	(上記のとおりで水内で使用されていない限り行頭に一致します。

Altova MapForce 2021 Professional Edition

\$	行末に一致します。例えば A+\$ は 行末のA に一致します。
*	前の式のセン、おけは複数の発生に一致します。例えば、Sa*m Sm、Sam、Saam、Saaam などに一致します。
+	前の式の1つの、おは複数の発生に一致します。例えば、Sa+mはSam, Saam, Saam などに一致します。

特別文字の検索

3.1.10 XSLT コードのプレビュー

XSLT 1.0、XSLT 2.0、おけXSLT 3.0をデータ変換言語変換言語とて選択している場合、MapForce にお生成されたXSLT コードをルビューすることができます。

XSLTコードをルビューするコは、マピングウィドウの下のXSLT ダブ(おコはXSLT2 おコはXSLT3) ダブを必要に応じてクリックします。

メモ XSLT 1.0、XSLT 2.0、おけはXSLT 3.0 変換言語として選択すると使用することができます。

3.1.11 XSLT コードの生成

マルングは対応する言語のためにデザインされていると仮定して、マルングからXSLT 1.0、XSLT 2.0、おけはXSLT 3.0 コードを生成することができます。変換言語の選択も参照してくたさい。

XSLT コードを生成する:

- 1. メニューアイテム「ファイル | コードの生成 | XSLT 1.0 (おけは XSLT 2.0、 XSLT 3.0)を選択します。
- 2. 生成されたXSLT ファイルを保存するフォルダーを選択し、「OK」をクルクします。 MapForce は コードを生成し、 メッセージ ウイドウ内にオペレーションの結果を表示します。

生成されたxslt ファイルの名前には、 <A>MapTo.xslt の書式を持ちます:

- "<A>"はマピング設定内のアプリケーション名フィールド内の値です。
- 「"はターゲトマピングコポーネトの名前です。この値を変更するコはターゲトコンポーネトの設定を開き、コンポーネト名 フィールドの値を変更してくたさい(次を参照してくたさい)コンポーネト設定の変更)。

生成されたコード内のライブラルなも参照してくたさい。

RaptorXML Server を使用して自動化する

XSLT コードの生成後、.xslt ファイルを持つ同じディレクトリ内に DoTransform.bat という名前の ッチファイルが生成されます。 RaptorXML Server を使用して DoTransform.bat を実行することができます。<u>RaptorXML Server を使用した自動化</u>も参照し てください。

RaptorXML Server を使用して変換を行う

- 1. RaptorXMLをダウロードページからダウロードしインストールします(https://www.altova.com/ja/download。
- 2. 上の操作で指定した出力フォルダーに入りされているDoTransform.bat バッチファイルを起動します。

RaptorXML がインストールされた場所をオペレーティングシステムのpath 環境変数に追加する必要かある場合かあります。RaptorXML ドキュメントを確認することができます(<u>https://www.altova.com/ja/documentation</u>)。

3.1.12 XQuery コードのプレビュー

XQuery をデータ変換言語として選択している場合、MapForce にとせたはたXQuery コードをプレビューすることができます。

生成されたXQuery コードをプレビューする方法:

• マンピングウィンドウの下にある「XQuery」タブをクリックします。

メモ XQuery タブは「XQuery」か変換言語とて選択されていると使用することができます。

3.1.13 複数のマッピングウィンドウとの作業

MapForce は、複数のドキュメナインターフェイス(Multiple Document Interface)(MDI)を使用します。MapForce で開く各 マピングファイルコは個別のウィドウがあります。これにより、複数のマピングウィドウと作業、ませま並べ替えやサイズ調整をMapForce メイン(親)ウィドウで行うことができます。で行うことができます。全ての開かれているウィドウを標準ウィドウのレイアケトを使用して以下の ように並べ替えることができます:上下に並べる、左右に並べる、重ねて表示。

複数のマッピングがMapForce内で開かれている場合、マッピングペインの下の部分に表示されているタブを使用して素早く切り替えることができます。

	*
	_
4	
マッピング XSLT2 DB クエリ 出力	
New Design1 New Design2 New Design3 4 New Design3	~
	~

ウイドウ管理オプタンは「ウインドウ」メニューと「ウインドウ」ダイアログボックスで使用することができます。「ウインドウ」ダイアログボックスから、アクションを実行、おけよ現在開かれているマメピングウイドウで(保存、閉じる、おけよ最小化を含む)アクションを使用することができます。

ウィンドウ	×
ウィンドウの選択(W):	アクティブにする(A)
New Design1	OK(0)
New Design2	
Example New Design3	1本仔(3)
	ウィンドウを閉じる(C)
	重ねて表示(D)
	縦に並べる(Z)
	横に並べる(V)
	最小化(M)

ウィンドウダイアログボックス

「ウィンドウ」ウィンドウ」を使用して、ウィンドウダイアログボックスを開くことができます。ウインドウダイアログボックスから複数のフィンドウを選択するコよ必要なエントーを「Ctrl」キーを押しながらクリックしてください。

3.1.14 マッピング設定の変更

マンピング設定ダイアログボックスから現在アクティブなマンピングデザインファイルのドキュメント特有の設定を変更することができます。この情報は*.mfd ファイルコ保管されます。

マッピング設定ダイアログボックスの開きかた

• 「ファイル」メニューから「マッピング設定」をクリックします。

😢 Mapping Settings	×
Code Generation Application name: Mapping Java base package name: com.mapforce Make paths absolute in generated code Make paths absolute in generated code Reference libraries with paths relative to generated XSLT / XQuery files Ensure Windows path convention for file path output for files from a local file system	
Output File Settings Line ends: Platform default (supported in Built-in execution and C#, Java and C++ code generation)	~
XML Schema Version • v1.1 if <xs:schema vc:minversion="1.1"> v1.0 otherwise • Always v1.1 • Always v1.0</xs:schema>	
OK Cancel	

マッピンク設定ダイアログボックス

使用することのできる設定は以下の通りです。

アプリケーション名	生成されたXSLT ファイルおけませ成されたJava、C#、おけま C++ アプリケーションの名前を定義します。
ベース・シケージ名	このオプションは Java か変換言語とて選択されている場合に適 用することができます。 Java 出力のナックのベース・シケージ名を定 義します。
生成されたコード内でパスを絶対パスにする	このチェックボックスは(XSLT ライブラなどの)外部ライブラファイ ルへの) なを除き、マルングコンポーネント内のすべての) な」影響 を与えます。 チェックボックスは生成されたプログラムコード内でおよび MapForce Server Execution ファイル(mfx) 内で、および FlowForce Server 「ニデプロイされたマッピング内で ファイル な が相対, な化まける絶対, なかを定義します。 <u>多種の実行環境内</u> ての、なも参照してくたさい。以下も参照多種の実行環境内の、 ス
生成されたXSLT / XQuery ファイルロン相対的ない ひを持つ レファレンスライブラリ	マピング言語がXSLT おさなQuery*の場合このチェックボックスを適用することができます。

	XSLT おはXQuery ライブラをマピング参照する場合、およ びマピングからXSLT おはXQuery ファイルを生成する場合 このオプションは通常役に立ちます。ライブラル なを生成された XSLT おはXQuery コードのデルクトリニズはして相対的にする 場合、チェックボックスを選択します。 チェックボックスが選択されていない場合、ライブラル なは生成され たXSLT おはXQuery コード内で絶対パスなります。生成さ れたコード内のライブラル な
Windows ノマカウァイル への規則であることを確認	マナング言語がXSLT 2.0、XSLT 3.0、おけよXQuery* 2.0の場合このチェックボックスは適用可能です。
	チェックボックスはWindows の や規則が使用されていることを確認します。XSLT 2.0、XSLT 3.0、おけはXQuery を生成する場合、現在処理されているファイル名は、ローカルファイルのためにフォームfile://URI内のやを返す document-uri 関数を使用して取得されます。
	このチェックボックスがアクティブな場合、file://URI パマ仕様 は更なる処理のために自動的に完全なWindows ファイル ぷ (例、 "C:¥")に変換されます。
行末	このエンボボックスにより、出力ファイルの行末を指定することができ ます。「プラ・トフォームデフォルト」によりターゲートのオペレーティン グシステムで使われているデフォルトの設定(Windows には (CR+LF)、macOS には(LF)、Linux には(LF))が使用され ます。行末は手動でも選択することができます。MapForce Server 実行可能ファイル(.mfx)にマッピングをエンパルする、 おけま、他のオペレーティングシステム上で作動するFlowForce Server にマッピングをデプロイする際に重要です。
XML スキーマバージョン	マピングファイルで使用されたXML スキーマバージョンを定義する ことができます。バージョン 1.0 おけは 1.11ご準拠するスキーマをロ ードするかを定義します。すべてのバージョン 1.1 特有の機能は現 在サポートされています。
	xs:schema vc:minVersion="1.1" 宣言が存在する場合、 バージョン 1.1 が使用されます。それ以外の場合、バージョン 1.0 が使用されます。
	XML スキーマのバージョン ④ <xs:schema vc:minversion="1.1"> であれば v1.1 それ以外の場合は v1.0 ○ 常に v1.1(1) ○ 常に v1.0(0)</xs:schema>
	XSD ドキュメントがvc:minVersion 属性 おけは1.0 おこ は1.1 以外のvc:minVersion 属性の値を持たない場合、 XSD 1.0 がデフォルトのモードです。
	メモ vc:minVersion 属性とxsd:version 属性を混 同しないでくたさい。前者がXSD バージョン番号を持

つ場合、後者はボキュメントの、ージョン番号を持ちます。
既存のマピング内のこの設定を変更することは、選択されたスキー マ・チージョンの全てのXMLスキーでを再ロードすることを意味し、 妥当性を変更する可能性があります。

* MapForce Professional おさはEnterprise Edition を必要とます。

3.2 コンポーネントとの作業

コンポーネントは、MapForce内のマンピングデザインで中心的な要素です。全般的に、「コンポーネント」という用語は、データノースとデータターゲートとしての役割を果たすオブジェクト、おけま中間処理の段階にあるマンピング内のデータニ簡単な名称を与えるために役に立ちます。

コポーネトには以下の2つの主なカテゴルがます:構造コパーネトと変換コパーネト。

構造コンポーネントは、データの抽象的な構造とスキーマを表します。例(メニューコマンド「挿入 | XML スキーマ/ファイル」を使用して マンピングエレアに XML ファイルを追加すると、マンピングコンポーネントしています。構造コンポーネントとその特化についての詳細は、データノ <u>ースとターゲント</u>を参照してくたさい。一部の例外を除いて、構造コンポーネントは、アイテムとシーケンスから構成されます。アイテムはマンピン グユニントの最も低いレベルです(例 XML ファイル内の単一の属性、ませは単純型の要素)。シーケンスはアイテムのコレクションです。

変換コンポーネントは、データを変換(例:関数)、おけま変換の手助け、例:定数おけま変数)をします。これらのコンポーネントを使用して、 複数のデータ変換タスクを達成する方法に関しては、<u>マンピングのデザイン</u>を参照してくたさい。

構造コンポーネントを使用することにより、ファイルまたはその他のノースからデータを読み込み、おこ、ファイルまたはその他のノースへデーダを書き込み、マメピング処理内の中間の段階で、例: プレビューするためコテーダを保存することができます。この結果、構造コンポーネントにはよい以下の8つの型があります:

- ソース。マピングエルアの左側に配置し、MapForceにデータを読み込むように命令し、コンポーネントをノースとして宣言します。
- ターゲオ。マピングエルの右側に配置し、MapForce にデータを書き込むように命令し、エンポーネオをノースとして宣言します。
- パススルー。これは特別なコンポーネトの型でノースとターゲオとしての役割を果たします。(更に詳しい情報に関しては、<u>チェーンマッピング/パススルーコンポーネント</u>を参照)。

マピングエアでは、コンポーネトは三角形で表示されます。次のサンプルマピングは、3 つのノースコンポーネント、1つのXML コンポー ネント、ソースに書き込まれる前にデータが通過する複数の変換コンポーネント(関数とフィルター)を表しています。



CompletePO.mfd

このマッピングサンプルは以下の、次で見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\CompletePO.mfd。

3.2.1 コンポーネント内の検索

特定のノード/アイテムをコンポーネント内で検索する方法:

- 1. 検索するエレポーネトをクリックします。「CTRL+F」キーをクリックします。
- 2. 検索用語を入力して、「次を検索」をクリックします。

検索		×
検索対象 (<u>E</u>):	•	次を検索 <u>(N</u>)
オプション		前を検索 (<u>P</u>)
📃 単語の完全マッチ (<u>W</u>)		すべて検索 (<u>A</u>)
大文字と小文字を区別(C)		キャンセル
		詳細(⊻) ▲
検索要素	制約	
✓ 名前を表示(D)	◎ 全てのアイテム (<u>I</u>)
□ ローカル名 (L)	◎ 接続付きのアイテム (E)	
名前空間 URI(<u>U</u>)	◎ 接続なしのアイテム (M)	
注釈(<u>0</u>)		
型 (I)		

高度なオプションを使用して、どのアイテム(ノード)が検索されるか、おと特定の接続をベースコンた厳密な検索オプションを定義します。

3.2.2 コンポーネントの整列

コンポーネントをマッピングペイン内でドラッグすると、MapForce は、自動配置 ガイドラインを表示します。このガイドライは、マッピングウイン ドウ内で他のエンポーネントと位置を合わせる際とても役に立ちます。

下のサンプルマッピングでは、下のエンポーネントを移動します。ガイドラインは、マッピングの左のエンポーネントに位置を合わせることができます。



コンポーネント 自動配置 ガイドライン

このオプションを有効化または無効化する

- 1. 「ツール」メニューから「オプション」をクルクします。
- 2. 「編集」グループから、「マウスのドラッグ」にてコンポーネントを整列」のチェックボックをチェックします。

3.2.3 コンポーネント設定を変更する

コンポーネントをマッピングコリアに追加した後、から適用することのできる設定をコンポーネント設定ダイアログボックスから構成します。コンポーネント設定 ダイアログボックスを以下の方法で開くことができます:

- コンポーネトを選択して、「コンポーネント」メニューから「プロパティ」をクリックします。
- エポーネトヘッダーをダブルクトックします。
- コポーネトペッダーを右クリックして、「プロノティ」をクリックします。

コンポーネトの型によ使用できるオプションが異なることに注意してくたさい。それそれのコンポーネトの型に適応することのできる設定に関するレフォレンスは以下を参照してくたさい

- <u>XML コポーネト設定</u>
- <u>デーダベースンポーネト設定</u>
- <u>CSV コパーネト設定</u>
- <u>固定長フィーノレト コンポーネント 設定</u>

XML などの、ファイルベースのエレポーネント、「ファイル/文字列」(「File/String」) ポタンがレートノードの横に表示されます。このがみは、 単一のマンピング内で複数のファイルを処理ませませ成する際に適用することのできる高度なオプションを指定します(複数の入力ませませ) カファイルを動的に処理を参照)。更に、このがタノま文字列の解析ませまデータを文字列にシノアル化するせかの高度なオプションを有効化します(文字列の解析とノアル化を参照)。

3.2.4 入力の複製

1つ以上のノースからデータを受け入れるためにことポーネントを構成する必要がある場合があります。例えば、2つの異なるスキーマからのデー タを単一のスキーマに変換する必要があるとします。目的のスキーマが2つのノーススキーマからのデータを受け入れるようにするには、エレポーネ ント内の入力アイテムを複製して行います。入力の複製は、ターゲトエンポーネントであるエレポーネントにとってのみ意味があります。与えら れたターゲト、エンポーネントは、必要に応じて、必要な数複製することができます。

特定の入力アイテムを複製するコよ、アイテムを右クリックして、コンテキストメニューから「後/前に入力を複製する」を選択します。



上のイメージでは、アイテムLineItemが2番目のノースからマップを可能にするために複製されています。

入力を複製すると、元の入力と複製された入力の間をマップすることができます。例: これによりデータをノースA からもとの入力に、またデータをノースB から複製された入力にコピーすることができます。

メモ XML インスタンスを無効にするため、XML 属性の複製は許可されていません。スキーマ内の要素のmaxOccurs 属性の値に 関わらず、XML 要素の場合、入力の複製は許可されています。この振る舞いは、スキーマが後に変更することが可能なため、ま た、ソースデータが注意のため、意図的です。 例えば、マノピングは、マノピング上で入力が複製されても、単一のXML 要素を 生成することができます。

順序を追す一例は、複数のノースからしつのターゲットにマップを参照してくたさい。

3.3 接続との作業

マピングは、データを1つのフォーマトから他のフォーマトに変換することを意味します。基本的なマピングシナけては、マピングエノアニノ ースとターゲトデータを表すコンポーネントを追加します(例、ソース XML スキーマと目的のスキーマ)、そして視覚的なマピング接続を2つ の構造間に描きます。ですから、接続はノースから目的地へどのようにデータがマップされたかを表す視覚的な表示といえるでしょう。

コンポーネントには、マンピングで小さい三角形として表示される入力と出力があり、これらはコネクタと呼ばれます。出力コネクタは接続を描くアイテムの右横に配置されます。

2つのアイテム間に接続線を引く:

 ソースアイテムの出力コネクタをクトックし、目的アイテムまでドラックします。ドロップアクションの許可されると、ビトのレンカカーソ ルの横に表示されます。



入カコネクタは入力される接続のみを受け入れます。同じ入力に第2番目の接続を追加使用とすると、接続を新規の接続と置き換える か、ますは入力アイテムを複製するかを問うメッセージボックスが表示されます。出力コネクタは複数の接続を持つことができ、それぞれが異なった入力を持ちます。

異なったアイテムに接続を移動する

• 接続のスタッズターゲットに近い直線のセクション)をクリックして、移動先までドラッグします。



異なったアイテムへの接続をコピーする

• 接続のスタッズターゲトに近い直線のセクション)をクリックして「Ctrl」キーを押しながら移動先まで、ラッグします。

接続の対極にあるアイテムを表示する

 マウスポインターを(入力/出力アイコンの近くにある) コネクタの直線部へマウスオーバーすることで、コネクタが イライトされ、ポップ アップが表示されます。ポップアップコよ コネクタの対極にあるアイテムの名前が表示されます。同一の出力アイコンから複数の接続 が定義されている場合には、最大10の名前が表示されます。スクレーンショナ では、SinglePrice と value2 という複数のター ゲット に対して接続が行われていることが理解できます
接続設定を変更するには以下を行います:

- 「接続」メニューからプロパティ」をクリックします。(このメニューアイテムは、接続を選択すると有効化されます)。
- 接続をダブルクトックします。
- 接続を右クトックして、「プロパティ」をクトックします。

接続設定を参照してくたさい。

接続を削除するには、以下を行います:

- 接続をクルクして、「削除」キーを押します。
- 接続を右クトックして、「削除」をクトックします。

3.3.1 必須の入力について

マピング処理内での手助にをするために、MapForceはターゲオンポーネント内の必須の入力をオンジ色でソイライトします。

- XMLとEDI コポーネトは minOccurs パラメーターがで1 と同じおは1 よけき アイテムです。
- データベース内では、これらばゼロではない」として定義されるフィールドです。
- WSDL 呼び出しとWSDL レスポレス(全てのノード)
- 必須と定義された XBRL ノード
- 関数内では これらは、・ラメーターがマップされると、接続の必要な他の必須の、ラメーターが、イライトされる特定の必須の、ラメ ーターです。例、フィルター入力、・ラメーターの、つかマップされると、他の、ラメーターは自動的にン イライトされます。
- MS Excel シート内のフークシート名

例

… MapForceExamples フォルダー内にある、CompletePO.mfd などのマンピング作成の際、挿入された XML スキーマファイルは下に表示されるとおりです。



Customers コンポーネトのNumber 要素はCompletePO コンポーネトのNumber 要素に接続されて、ます。接続が作成される と CompletePO コンポーネトの必要なアイテムノードは、ハイライトされます。折りせきまれた「Article」ノード/アイコン も イライトされ ます。



3.3.2 **優先する接続線の表示の変更**

MapForce ではマピングウイドウに表示されるコネクタの表示方法を選択することができます

選択コンポーネントの接続線の表示により以下のように表示が引きわけます

- 全てのマピングゴンポーネントか黒色で表示されるか、
- 現在選択されているエレポーネントに関係したコネクタが黒色で表示されるとともに、その他の接続がリレーで表示されます。
- ソースからターゲットへの接続線の表示により以下の表示が切り替わます
- 現在選択されているエレポーネントに直接接続されているエネクタか黒色で表示されるか、
- 現在選択されているコンポーネントの入力ませる出力アイコンペレクされているコネクタが黒色で表示されます。

3.3.3 接続の注釈

個別の接続をマピングに詳細なコメントを与えラベル付けすることができます。このオプションは「全ての接続の型」で使用することができます。

接続に注釈をつける

ž

ድ

- 1. 接続を右クリックして、コンテキストメニューからを「プロパティ」選択します、
- 2. 現在選択されている接続の名前を「詳細」フィールドに入力します。これにより注釈設定グループ内の全てのオプションが有効化されます。

- 3. 残りのブループを使用して、ラベリの「開始の箇所」、「配置」、「位置」を定義します。
- 4. 注釈テキストを確認するコよ、表示オプションソールベー内の「注釈の表示」

🗄 🗂 footnote	扣保扣当	🕀 🕀 🗄 footnote
Mortgage Servicing Rights		🗕 🕂 🕀 🕞 🛑 🕂 🛑
= id		— i d
= unitRef	·	unitRef

✓モ 「注釈の表示」アイエカ無効化されている場合、接続にカーノルをポイントすると、注釈を表示することができます。「ヒントの 表示」 「「ツールバーボタンが表示オプションツールバー内で有効化されている場合、注釈のテキストは 吹き出し内に表示されます。

3.3.4 接続設定

コネクタを右クリックしてコンテキストメニューからプロパティを選択するか、コネクタをダブルクリックして接続設定 ダイアログボックスを開き、現在のコネクタに対して特別な(混合コンテンツ)設定を定義することができます。使用 することのできないオプションは無効化されています。

接続設定				
接続の種類				
◎ ターゲット 優先(標準マッピング)(S)				
◎ 全てをコピー(子アイテムをコピー(Y)				
◎ソース優先マッピング(混在コンテンツ)(0)				
 処理命令をマップ(P) 				
コメントをマップ(C)				
注釈設定				
説明: mixed				
開始ロケーション				
◎ソース接続				
◎ 中点				
◎ ターゲット 接続				
配置 位置				
◎ 水平				
◎垂直				
◎ 傾斜				
<u>ОК</u> キャンセル				

接続設定ダイアログボックス

complexType のアイテムのために、以下のマッピングのための接続型を選択することができます(これらの設定は、テキストノードを持たない complexType アイテムにも適用することができます):

ターゲオ優先(標準)	接続の型をターゲト優先に変更します(ターゲト優先/標準マピングを参照)。
<i>全てコ</i> ピー (<i>子アイテムの</i> <i>コ</i> ピー)	接続の種類を「全てコピー」 に変更し、ソースとターゲオコンポーネント内の全ての等し、 アイテムと自動的 「は接続します(<u>全てコピー接続</u> を参照)。
ソース優先 (複合ユンテン ツ	接続の種類を「ソース優先」に変更し、 おこマップされる追加要素の選択を有効化します。 マレングに適 するけっかに、 追加要素は XML ソースファイル内のマップされ <i>こアイテム</i> の子アイテムである必要があります。 「 処理命令をマップ」 おっち、「 コメントをマップ」 チェックボックスを有効化することにより、 これらのデータグ ループを出力ファイルに含むことができます。



注釈設定グループによ接続へ注釈を与えることができます(接続の注釈を参照)。

3.3.5 一致する子要素の接続

ソースとターゲナオンポーネントの両方内で同じ名前のアイテム間で複数の接続を作成することができます。「全てコピー」 接続は、デフォルトで作成されます(全てコピー接続を参照)。

「マッチする子を自動的に接続する」オプションをオンまたはオフに切り替えるには、以下を行います:

- 「一致する子要素の自動接続」() ツールーボタをクルクします。
- 「接続」メニューから一致する子要素の自動接続」をクルクします。

「マッチする子を接続する」のための設定を変更する方法:

- 1. 両方のエレポーネト内のと「子アイテム」しら名前を共有する2つの(親)アイテムを接続します。
- 2. 接続を右クルクして、「一致する子要素の接続」オプロンを選択します。



- 3. 必要なオプシュンを選択して(下のテーブルを参照してください)、「OK」をクリックします。ダイアログボックス内の設定定義に当て はまる、接続が同じ名前の全ての子アイテムのナックに作成されます。
- メモ 「子の自動接続の切り替え」(目))ツールレーポタンがアクティブは場合、ここで定義する設定が2つのアイテムを接続する際 に適用されます。

大文字と小文字の区別を無視	子アイテム名の大文字と小文字の区別を無視します。
名前空間を無視	子アイテムの名前空間を無視します。
再帰的	一致するアイテム間の接続を再帰的に作成します。すなわち、階層構造内でどれま ど深くアイテムがネストされていても、アイテムか同じ名前を持つ限り、アイテム間の接 続か作成されます。
属性と要素を混合	有効化されている場合、同じ名前を持つ属性と要素の間の接続を許可します。例 2つの「Name」アイテムが存在する場合、1つか属性、もう1つか要素である場合 でも、接続を作成します。
全コピー接続の作成	この設定は、デフォルトの設定では有効化されています。この設定は、「全てコピー」 とソースとターゲット アイテム間の接続を(可能であれば作成します。
既存の出力接続を無視	出力接続が既存の場合でも、追加の接続を一致するアイテムのために作成します。
保持	既存の接続を保持します。
上書きを保存	定義された設定に従い接続を再作成します。既存の接続は破棄されます。
既存を全て削除	新規接続を作成する前に、既存のすべての接続を削除します。

接続の削除

接続マッチする子を接続するダイアログを使用して作成された接続、おけよマッピング処理中の接続をグループ化して削除することができま す。

⊖ () Per () Fir ∰®	前に重複した入力を追加(B)	nme I		
OLa	後に重複した入力を追加(I)	nail Imes	sticDailvRat	
OPh 🏘	複製を削除(U)	reig	nDailyRate se-detail	
⊟ ()exp = tvi	コメント/処理命令を削除(M)	pens	se-item	6
ex	ー コンテンツを CDATA セクションとして書き込む(T)	ill-t	:0	
⊕ () Me	上へ(0)	rave	el vel-Cost	
	下へ(V)	Des Car	tination -Rental	
	接続の削除(D)	- ui	全てのダイ	レクト接続の削除(D)
	変数コンポーネントの作成(V) ▶		全ての入力	子要素の削除(I)
⊡OTr =mean) Lo	全ての出力 [.] cation	子要素の削除(0)

接続の削除:

- 1. コンポーネント内のアイテムの名前を右クトックします(このサンプルでは「Person」)。 2. 「接続の削除|全ての...接続の削除」を選択します。

全てのダイレクト接続を削除	直接マップされた、まとは現在のエンポーネトから他のノースまとしはターゲトコンポー ネントにマップされたすべての接続を削除します。
全ての入力子接続を削除	ターゲナコンポーネント内でアイテムを右クトックした場合のみアクティブコンはます。全 ての入力子接続を削除します。
全ての出力子接続を削除	ソースコンポーネント内でアイテムを右クリックした場合のみアクティブコンはます。全ての 出力子接続を削除します。

3.3.6 コンテキストメニューの接続

接続を右クトックすると、以下のコンテキストコマンドを使用することができます。



マッチする子を接続	「マッチする子を接続する」 ダイアログボックスを開きます。(<u>一致する子要素の接続</u> を参 照)。接続がマッチする子アイテムを持つことができる場合、このコマンドを有効化することが できます。
削除	選択された接続を削除します。
ソースへ移動: <item name=""></item>	現在の接続のノースコネクタを選択します。
ターゲットに移動: <item name=""></item>	現在の接続のターゲナコネクタを選択します。
ターゲオ優先 (標準)	接続の型をターゲオ優先に変更します(ターゲオを優先接続を参照)。
全てコピー(子アイテムのコピー)	接続の種類を「全てコピー」「変更し、ソースとターゲトコンポーネント内の全ての等しい アイテムと自動的「接続します(<u>全てコピー接続</u> を参照)。
	ソースアイテムとターゲオトアイテムの両方が子アイテムを持つ場合、このコマンドは有効化 されます。
ソース優先(複合型コンテンツ)	接続の種類を「ソース優先」に変更します(ソース優先マンピングを参照)。

	ソースアイテムシターゲットアイテムの両方が子アイテムを持つ場合、このコマンドは有効化 されます。
並べ替えの挿入: Nodes /Rows	ソースとターゲ水アイテム間のノートコンポーネントを追加します。(<u>データの置換え</u> を参照)。
フィルターの挿入: Nodes/Rows	ソースとターゲ水アイテム間のフィルターコンポーネントを追加します。(<u>フィルターと条件</u> を 参照)。
SQL-Where 条件0挿入	ソースとターゲ水アイテム間のSQL-Where ユンポーネントを追加します。(SQL WHERE / ORDER エンポーネントを参照)。.
Value-Map の挿入	ソースとターゲホアイテム間のValue-Map エンポーネントを追加します。(<u>Value-Map</u> の使用を参照)。
לבו אידיא	「接続設定」ダイアログボックスを開きます。(接続設定を参照)。

3.3.7 見つからない親接続の通知

ソースならびにターゲルアイテム間の接続を手動で作成する場合、起こりえるマンピング結果がMapForceにより分析されます。子アイテム 同士をマンピングした場合、ソースアイテムの親とターゲットアイテムの親も接続するように促すプロンプトか表示されます。

これはマピングのプレビューを出力ウィンドウェて行う際に、子アイテムの表示を1 コニ限定しないすめのたのです。ソースノードから得られるものが単一の値ではなく、シーケンス形式の場合、通常親アイテムの接続を行う必要があります。

<マイドキュメント>¥Altova¥AapForce2021¥MapForceExamples¥Tutorial¥フォルダーイコ収められているTut-OrgChart.mfd マピングを以下に示します。ソースコンポーネントにあるtext()アイテムかターゲットコンポーネントにあるtext()アイテムへ 接続されると、メッセージボックスが表示され、親アイテムのpara が接続されておらず、出力では一度だけ生成される旨が通知されます。複 数のpara アイテムをターゲットにて表示する」コよ、ソースならび、こターゲットアイテムのpara を接続する必要があります。



Tut-OrgChart.mfd (MapForce Basic Edition)

ターゲナト内で複数のpara アイテムを生成するコよソースとターゲナ para アイテムをそれそれ接続します。

上に表示されるようなメッセージを非表示にするコよ以下を行います。

- 1. 「ツール」メニューから「オプション」をクトックします。
- 2. 「メッセージ」グループをクルクします。
- 3. 接続を作成した際、祖先アイテムへの接続を提案アイテムチェックボックスをクリックします。

3.3.8 接続と子接続の移動

異なったコンポーネントに接続を移動する場合、MapForce は、自動的に同じ子接続をマッチし、この接続か新しい場所に移動されるべ きかプロンプトします。この機能の通常の使用方法は、既存のマンピングがある場合、ターゲットスキーマのルート要素の変更です。通常、こ のような状況が発生すると、すべての降順接続を手動でマンピングを再度行う必要がありますが、この機能はそのような状況を回避するた めこ役に立ちます。

<マイドキュメント> ¥Altova ¥Map Force 2021 ¥Map Force Examples ¥Tutorial ¥ フォルダーで使用することのできる Tut-ExpReport.mfd ファイルは、この別で修正され、別名で保存されます。



Tut-ExpReport.mfd (MapForce Basic Edition)

動作を理解するためコよ、以下を行います

- 1. Tut-ExpReport.mfd サンプルマンングを開きます。
- 2. ターゲナスキーマのCompany ルト要素をCompany-EU に変更するために、MapForce の外部でExpReport-Target.xsd スキーマを編集します。MapForce を閉じる必要はありません。
- 3. ターゲオスキーマのCompany ルト要素をCompany-EU に変更すると、MapForce ビ変更されたファイルプロンプトが内 に表示されます。

変更されたファイル			
これらのファイルまたはそれらのコンポーネントは修正されるか、またはアプリケーションの外部から削除されました:			
C:\Users\altova\Documents\Altova\MapForce2015\MapForceExam\ExpReport-Target.xsd	*		
	_		
」 ファイルを再ロードするか、変更を無視することができます。			
再ロード(R) キャンセル			

4. 更新されたスキーでを再ロードするために、「再ロード」ボタンをクリックします。ルート要素が削除されたため、コンポーネントは複数の見つからないレードを表示します。



5. コンポーネトの上の「新しいルート要素の選択」をクリックします。(コンポーネトへッダーを右クリックしてコンテキストメニューから 「ルート要素の変更」を選択することで、ルート要素を変更することができます)。

C	ノルート要素の選択		x
	名前	名前空間 URI	<u> </u>
	CashAdvance	http://my-company.com/namespace	
	Company-EU	http://my-company.com/namespace	
	Date	http://my-company.com/namespace	

6. Company-EU を新しいト要素とて選択し、「OK」をクリックして確認します。Company-EU ルト要素がユポーネントの上に表示されます。



7. ソースエポーネトのexpense-report アイテムとターゲオコポーネトのCompany アイテム間にある接続のターゲオス タッブをクトックして、Company-EU ターゲオコンポーネトのレート要素の上にドラッグアンドトロップします。



この接続のみ(O) 子孫の接続を含める(I) キャンセル

マッチした子孫の接続も一緒に移動しますか?

- 8. 「子孫の接続を含む」をクトックする。これにより、新しいレート要素の下で再度マップするようにMapForce に命令し、マルン グは再度有効となります。
- メモ マピングするノードが、ソースノードと同じ名前を持ち、異なる名前空間にある場合、通知ダイアログボックスに、以下を表示する 追加ボダンカ表示されます:「子孫とマップ名前空間を含む」。このボダンをクリックすると、同じ名前空間の子接続が同じ子ノード のノース親ノードとして移動されます。このボダンをクリックすると、異なる名前空間ノードの下の同じ子ノードのノース親ノードと同じ ように同じ名前空間の子接続を移動します。

3.3.9 コンポーネント削除後の接続の保持

他のエンポーネントに複数の(子) 接続を持つ、コンポーネントを削除すると、何か起こるかを決定することができます。例: フィルター おけよソ ートコンポーネント。全ての子接続を保持し、個別に復元する必要がないサックこの機能は投に立ちます。

コポーネトカ削除された後に、子接続を保持ませば復元すること、ませば全ての子接続をすくに接続することが選択できます。

現在の設定を確認するためゴッール オプション | 編集」(タブ)を選択します。チェックボックスのデフォルト設定は、非アクティブです。 すなわち「スマートなエンポーネント削除(役に立つ接続を保持)」が無効化されます。



例 … MapForceExamples フォルダー内で CompletePO.mfd マピングを使用するコよ チェックボックスがアクティブ化されていると、 Customer フィルターは 以下に表示される子アイテムコ接続された<u>全てコピー</u>接続しています。



Customer フィルターを削除すると、削除の確認を問うプロレプトが表示されます。「しは、」を選択すると、フィルターは削除されますが、子接続は保持されます。



その他の接続はまだ選択されています、すなわち、赤で表示される箇所)。削除する場合は、Del キーを押します。

マノピングエリアをクリックすると、コネクタの選択が解除されます。

「スマートなコンポーネト削除...」チェックボックスか無効化されている場合、フィルターを削除すると全ての子接続がすくに削除されます。

メモ フィルターコンポーネトが「on-true」と「on-false」出力に接続されている場合、両方の出力のための接続は保持されます。

3.3.10 見つからないアイテムの扱い方

時か経つご従って、XMLスキーマーこで定義された要素や属性が追加/削除され、マピング内にあるエレポーやトの構造が変更されるとうことがあります。MapForceではプレースホルダーアイテムを使用することで、全ての接続を保持し、アイテムが削除された際にエレポーやトト間で関連する接続データの保持を行います。

サンプル

MFCompany.xsd スキーマファイルをサンプルファイルとて使用します。スキーマファイルをMyCompany.xsd とう名前でコピーし、両 スキーマ間にて Company アイテムこ対する接続を作成します。これで子要素の自動接続が有効になっている場合、コンポーネント間にあ る全ての子アイテムに対する接続が作成されます。



XMLSpy にてMyCompany.xsd を編集し、スキーマ内にある First ならびに Last アイテムを削除します。MapForce へ再度切り替えると、スキーでを再ロードするよう促されます。 再ロードすることで、MapForce 内にあるコンパーネント か更新されます。



削除されたアイテムならびにコネクタが、アイコンとともにMyCompany コンポーネントに表示されます。必要な場合は地のアイテムへコネクタを再接続することができるまか、コネクタを削除することもできます。

この状態でマピング(ませまたされたコード)のプレビューを行うこともできますが、警告がシッセージンイドウニ表示されます。プレビューならび、コード生成にないて、上のアイコンで示されるような、見つからないアイテムは無視されます。

ハイライトされているコネクタをクトックし、削除することで、見つからさいアイテム(この場合はMyCompany内部のLast)がコンポーネントから削除されます。



名前の変更されたアイテム

例えばPerson とう親アイテムがZPerson とう名前に変更された場合、オバンナル(Person)の親アイテムニネクタか保持され、子ア イテムのニネクタか削除されます。



「全てコピー」接続と見つからないアイテム

全てコピー接続は通常の接続と同じように扱われますが、見つからないアイテムへの接続は保持も表示もされません。



名前変更ませば削除されたエンポーネントソース

スキーマや、データベースなどのエレポーネントのデータソースが名前変更ませ」
消除された場合、エレポーネントに含まれる全てのアイテムが
ハイライトされます。エレポーネントの外周であるフレームにより、スキーマませ」
お示されます。この状態でプレビューやコード生成を行うことはできません。



マカスカーノルをハイライトされているエンポーネントへマウスオー・デーすることで、ポップアップが表示されます。



ハイライトされたエンポーネトのタイトルレーをダブルクトックすることで、コンポーネト設定ダイアログボックスか表示、異なるスキーマを選択することができます。 コンポーネト設定の変更を参照してくたさい。

コンポーネントがデータベースの場合、ダイアログボックス内の変更ボタンをクリックすることで、別のデータベースやテーブルを選択することができる ようしています。同名のテーブルへの接続は保持されます。

コンポーネント設定	[×
データベース		
データソース:	[:] orce2015¥MapForceExamples¥people.sqlite	C)
接続名 <mark>:</mark>	people	
データベースの種類:	SQLite	
接続文字列:	C:¥Users¥altova¥MapForce2015¥MapForceExamples¥ people.sqlite	*

全ての妥当な(正確な)接続 同様の構造を持オースキーマ まけはデータベースを選択した場合全ての妥当な(正確な)接続が保持されます。

3.4 マッピングプロジェクトとの作業

スタイアロンのマピングの作成に追加して、複数のマピングを含むマピングプロジェクトも作成することができます。プロジェクトに追加されるマピングはプロジェクトウィイ・つから簡単にアクセスすることができます。



プロジェクト ウィンドウ(MapForce Enterprise Edition)

プロジェクトの主な利点は、特別なプロジェクトに含まれる全てのマッピングのナメの(ターゲット言語と出力ディレクトリなど)共通のコード生成設定を定義が可能な点です。まプロジェクト内にフォルダーを作成することができ、プロジェクト内の個別のフォルダーのナメのカスタムコード生

成(C++、C#、とJava)の設定を指定することも可能です。MapForceーにより生成されたプログラムコードに関する詳細は、<u>コード</u> ジェネレーダを参照。

MapForce Enterprise エディションでは、Webサービスプロジェクトを追加して作成することができます。このようなプロジェクトは、既存のWebサービス記述言語(WSDL)ファイルをベースした、SOAPWebサービスを実装するJava おけよC#プログラムコードを生成することを可能してます。

3.4.1 プロジェクトを開く、検索する、閉じる

MapForce プロジェクトファイルは*.mfp 拡張子を持ちます。既存のMapForce プロジェクトを(「ファイル」メニューから 開く」をクリックして、マッピングと同様に開きます。

MapForce でマッピングプロジェクトが開かれた場合、プロジェクトウィンドウは、プロジェクトに追加された全てのファイルとフォルダーを表示します。デフォルトでは、MapForce をはじめて実行した場合、プロジェクトウィンドウ内で MapForceExamples.mfp プロジェクトをロードします。

プロジェクト内でファイルを検索する方法:

- 1. プロジェクトウィンドウで、検索するプロジェクトませるフォルダーをクリックします。
- 2. 「CTRL + F」を押します。
- 3. オプションで、検索オプションを選択します。例:検索にフォルダー名を含むには、「フォルダー名から検索」オプションを選択します。

検索	×
検索対象(N): Format オプション 単語の完全マッチ(W) 大文字と小文字を区別(C) マカルダー名を検索	次を検索(E) 前を検索(E) キャンセル

プロジェクトを閉じる方法:

• 「プロジェクト」メニューからプロジェクトを閉じる」をクルクします。

3.4.2 新規プロジェクトの作成

新規プロジェクトを作成する

- 1. メニューから ファイル 新規作成」を選択します。
- 2. プロジェクトファイルアイコンを選択して、「OK」をクトックします。

新規ファイル	×
マッピング プロジェクト ファイル スプロジェ…	OK キャンセル
複数マッピングを含む MapForceプロジェクトの作成	

- 3. 「プロジェクトに名前をつけて保存」ダイアログボックスにてプロジェクト名を入力し、「保存」をクリックします。
- 4. プロジェクトフォルダーがプロジェクトタブに追加されます。

プロジェクト	Ŧ	ņ	×
🔊 Project1			
🔤 ライブラリ 🛅 プロジェク	Þ		

プロジェクトにマンピングを追加します。

現在アクティブなマッピングをプロジェクトに追加するには、以下を行います

- 「プロジェクト」メニューで「プロジェクト にアクティブファイルを追加する」 📴 をクリックします
- プロジェクトを右クトックして、「プロジェクトにアクティブファイルを追加する」

プロジェクトに既存のマッピングファイルを追加するには、以下を行います:

- 「プロジェクト」メニューで「プロジェクト にファイルを追加する」
- プロジェクトを右クトックして、「プロジェクト にファイルを追加する」 🔂 を選択します。
- ヒント: 複数のファイルを開くには、開くダイアログボックス内でファイルを選択するときに「Ctrl」キーを押します。

プロジェクトからファイルまたはフォルダーを削除するには、以下を行います:

- プロジェクトウイドウ内のファイルを右クトックし、コンテキストメニューから「削除」を選択します。
- プロジェクトウィンドウ内のファイルを選択して、「削除」を押します。

3.4.3 コード生成設定の設定

全てのプロジェクトに対して、プロジェクト内の全てのマッピングに影響するコード生成設定を指定することができます。「プロジェクト設定」ダ イアログボックスを開くには、以下を行います:

- 「プロジェクト」ウィボウ内のプロジェクト名を右クリックして、「コンテキスト」メニューから「プロパティ」を選択します。
- 「プロジェクト」メニューカシ「プロパティ」をクリックします。

プロジェクト設定		×
プロジェクト 名(P):	MapForceExamples	
プロジェクトディレクトリ(D):	C: \Users\Documentation\Documents\Altova\MapForce20	
出力設定		
出力名:	MapForceExamples	
出力ディレクトリ(0):	C: \Users \Documentation \Documents \Altova \MapForce 20	参照(R)
言語(L):	Java 🔻	
Java 設定		
ベースパッケージ名(A):	com.mapforce	
	ОК	キャンセル

プロジェクト設定ダイアログボックス

使用することのできる設定は、以下のとおりです。プロジェクト名とプロジェクトディレクトリはプロジェクト作成後に変更することができません。

出力名	入力された値は生成されるプロジェクトおけまノリューションの名前、おた生成されたコ ード内のオブジェクトの名前を決定します。
出カディレクトリ	(このプロジェクト内の全てのマッピングから)生成されたコードが保存される Windows フォルダーを決定します。デフォルトでは、出力はプロジェクトディレクトリ内 の出力/ディレクトリニ保存されます。

言語	このプロジェクト内のすべてのマンピングファイルのためのロード生成言語を定義します。
	この設定は、Javaを変換言語とて選択する場合適用することができます。Java プロジェクト内で生成されたベースパッケージ名を定義します。

3.4.4 プロジェクトフォルダーの管理

プロジェクト内のマピングをフォルダーに整理する場合、必要に応じてフォルダーを作成することができ、マピングに追加(マピング)ボラッグ) することができます。これらのフォルダーゴ 仮想」であり、 MapForce プロジェクト内でのみ意味があります、すなオち、オペレーティングシステ ム上の実際のフォルダー1コは対応していません。フォルダーを作成する利点の一つは、その特定のフォルダーのある全てのマピングファイルに対 して共通のコード生成設定(ターゲト言語と出力ディレクトリなど)を定義することができることです。

プロパティ			×
名前: コード生成設定 ③ デフォルトのプロジ の 以下の設定を使け 出力ディレクト	I 小設定を使用 用: -リ:		参照
言語:	XSLT	•	
		ОК	キャンセル

フォルダープロパティダイアログボックス

MapForce プロジェクト内にフォルダを作成する

- 1. 以下を行います:
 - 「プロジェクト」メニューから フォルダーの作成」 6 をクリックします。
 - 。 プロジェクトを右クトックして、「フォルダーの作成」 🖾 を選択します
- 2. プロ/ ディダイアログボックスで、必要とされるコード生成設定を入力して、「OK」をクリックします。

フォルダープロ・ティダイアログボックス内で定義することのできる設定は、以下のとおりです。

名前	フォルダーの名前

デフォルトプロジェクト設定を使用	これはデフォルトのオプションで現在のフォルダーのコード生成設定はプロジェクトと同じ 設定であることを意味します。ですから、プロジェクトからコードを生成する場合、 MapForce は、フォルダーレベルではな」、プロジェクトレベルで定義されたコード生 成設定を使用します。
	(プロジェクトレベルで設定されている以外の場合でフォルダーがりスタムコード生成設定を必要とする場合、「以下の設定の使用」を選択して、コード出力ディレクトリと言語を必要に応じて指定してくたさい。
出力ディレクトリ	(このフォルダー内の全てのマピングで)生成されたコードが保存されるWindows フォルダーを定義します。
言語	このフォルダー内の全てのマンピングファイルのナメのコード生成言語を定義します。

4 マッピングのデザイン

Altova Web サイト: 🔗 <u>データ統合ツール</u>

このチャプターは、データマッピングのデザイン方法とマッピングエアトのデータの変換方法について説明します。また、マッピングデザイノニ関連した、注意点に関しても説明します。以下のロードマップを利用して、特定のタスクまけコエンセプトに素早くアクセスすることができます。

を実行するコよ	以下を参照してくたさい
その他のスキーマの、又参照とマッピングで使用されるインスタンスと 他のファイルの作成と編集を行います。	相対と絶対パマの使用
必要に応じてデータマンピングを微調整します。(例: ターゲトコン ポーネント内のアイテムのシーケンスの調整)。	<u>マルング接続の種類</u>
異なるスキーマを持つ複数のノースからのデータを単一のスキーマに マップします。	複数のスキーマからデータをマージする
コンポーネントの出力を他のコンポーネントの入力として使用します。	チェーンマング/パスルーコンポーネント
複数のファイルを同じマンングにノースませお一ゲトとして処理します。(例: ディレクトリ内の全てのファイル)。	複数の入力ませ出力ファイルを動的に処理
(文字列・ラメーターなど)外部の値をマメピングにってします。	マンングレーターを与える
ファイルの代わりに、マンピングから文字列の値を取得します。	マンプから文字列の値を返す
マンピングデータの一部を後の処理のために一時的に保管します (プログラング言語内の変数に類似)。	<u>変数の使用</u>
昇順ませは降順の順序でデータを並べ替える。	<u>データの置換え</u>
特定の条件に従いnodes/rows をンパター、おけは値を処理します。	<u>ンルターと条件</u>
異なるスキーでを持つ複数のノースからデータをマージ、おゴンゴインします。	<u>データのジョイン</u> 複数のスキーマからデータをマージする
キーと値のペアの処理を行います。例 数値の表示を月数に変換 します。(01、02、などを) テキストの表示(一月、二月、などに) 変換します。	<u>Value-Map の使用</u>
特定の条件が発生した場合、エラーを返すようニマンピングを構成 します。	例外の追加
複雑なマピングのデザインで予期しない結果の発生を回避します。	マルングのレール単路
W ord 2007+, HTML、おはPDF などの読みやすい書式に マピングにより生成された結果を変換します。	StyleVision を使用してマピングの出力のスタイルを指定する方法
マルングからドキュメンテーションを生成します。	マレングドキュメンテーションの生成

重要な点は、MapForce は 多種の処理タスクを助ける、広範囲なビルドイン関数ライブラを搭載している点です(<u>関数ライブラルファ</u> レンスを参照)。ビルドインライブラルでも、十分ではない場合、自身のカスタム関数をMapForce で作成し、外部 XSLT ファイルおよび .dll ますは Java クラスライブラルで再利用することができます。更に詳しい情報に関しては、<u>関数の使用</u>を参照してください。

4.1 相対と絶対パスの使用

マンピングデザインファイル(*mfd)は、複数のスキーマとインスタンスファイルを参照する場合があります。MapForce により使用されるスキー マファイルは、マップされるデータの構造を決定し検証します。インスタンスファイル、は他方、スキーマイニ対してノースデータを読み込み、プレビュ ーし、検証することが要求されます。

マピングは PDF、HTML とW ord などの出力のナガニデータをフォーマト するナガニ Style Vision Power Stylesheets (*.sps) ファイルへの参照を含む場合がみます。 おこ、マピングは Microsoft Access おけよ SQLite などのファイルベースのデータベースへの参照を含む場合がみます。

マピングにていポーネートを追加する場合、マピングデザインで使用されるファイルへの参照はすべて MapForce により作成されます。ですが、上記の、又参照を常に手動で設定ませば変更することができます。

このセクションは設定のための命令、おけはマンピングにより参照されるその他のファイル型につなの変更方法、相対と絶対パマの使用が指する意味について説明します。

4.1.1 コンポーネント上で相対パスを使用する

コンポーネント設定ダイアログボックス(XML コンポーネントのために下に表示されています)は、コンポーネントにとど参照される可能性のある 複数のファイルのための絶対ませる相対パマを指定するオプションを提供します。

- 入力ファイル(すなわち、MapForce がデータを読み込むファイル)
- 出力ファイル(すなわち、MapForce かデータを書き込むファイル)
- スキーマファイル(スキーマを持つこレポーネントに適用可能)
- 構造ファイル(ユーザー定義関数の入力お台は出力/デメーター、お台は変数などの複雑な構造を持つエレポーネトに適用可能)
- PDF、HTML とW ord などの出力のオーダのデータをフォーマナするために使用される、Style Vision Power Stylesheet (*.sps)ファイル

対応するテキストボックス内で相対いてを直接入力することができます、下のイメージで赤線の枠内に表示されています)。

相対ファイル ジを入力する前に、マメピングファイル(.mfd)を最初に保存してくたさい。それ以外の場合、全ての相対パスはの Windows のペーソナルアプリケーションフォルダーニ対して解決されます(ドキュメント Witova WapForce 2021)これは、予期しない 振る舞しの場合かも少ます。

.mfd ファイルマングンオンオンマオンができます。下のサンプルマオベでを保存するようこMapForce に命令することができます。下のサンプル イメージでは、「MFD ファイルニ相対的なすべてのファイル マを保存する」 オプロンジ注意してください。(デフォルトかつ奨励されるオ プロンである チェックボックスが有効化されていると、コンポーネントによど参照されるファイルの マは、マンングデザインファイル(.mfd)のパ スニ相対的に保存されます。これはコンポーネントに参照される全てのファイルニ影響を及じます(イメージで赤線の枠内に表示されています)。

コンホーネント設定
コンポーネント名: books
- スキーマ ファイル(F)
C:Users¥altova¥Documents¥MyMapping¥books.xsd 参照(W) 編集(T)
- 入力 XML ファイル(I)
C:Users¥altova¥Documents¥MyMapping¥books.xml 参照(B) 編集(E)
出力 XML ファイル(O)
参照(R) 編集(D)
ムーゲット 名前空間のプレフィックス (P)・
✓ スキーマ/DTD参照を追加(スキーマの絶対バスを使用する場合はフィールドを空に)(A):
■ YML 宣言のまき込み(2)
▼ ▲ML 旦言の書き込み(ス) ■ 値をターゲットの型にキャスト(無効な出力を書き込むリスクがある時に数値またはデ
✓ ータ値のフォーマッティングの保持を無効化)(G)
■デジタル署名を作成(ビルトイン実行のみ)(L) 署名設定(N)
作成に失敗した際には: 〇処理を中止
◎ 署名無しで処理を続ける
出力エンコーディング
Iンコーディング名(C): Unicode UTF-8 ▼
バイト オーダー: リトル エンディアン ▼ □ バイトオーダーマークを含む(M)
StyleVision Power Stylesheet ファイル
参照 作成
☑ min/macOccursをベースにした入力の最適化処理を有効にする(E)
✓ 全ファイルパスを MFD ファイルへの相対パスで保存(S)
ОК ++>>セル

(C) 2015–2021 Altova GmbH

コンポーネント設定ダイアログボックス

上で説明されているコンポーネントはXML コンポーネンですが、設定「MFD ファイルに相対的なすべてのファイル マを保存する」は、下のファイルでも同じようて動作します。

- 複雑なユーザー定義関数の入力ませま出力パラメーターと複雑な型の変数により使用される構造ファイル
- 入力おけ出力 フラオファイル*
- XML フィールドをサポートするデータベースコンポーネントにより参照されるスキーマファイル*
- 入力おけ出力 XBRL、FlexText、EDI、Excel 2007+、JSON ファイル**

* MapForce Professional とEnterprise Edition ** MapForce Enterprise Edition のみ

上のエンポーネントを例として、.mfd ファイルがbooks.xsd とbooks.xml ファイルと同じフォルダーン存在する場合、ハマルは、以下のとお変更されます

C:\Users\altova\Documents\MyMapping\books.xsd は books.xsd に変更されます。 C:\Users\altova\Documents\MyMapping\books.xml は books.xml に変更されます。

ローカルではよいドライブませよURLを使用する。なは相対的にはよいません。

「上書き保存」メニューコマドを使用して、マンピングを新規のフォルダーに保存する場合、チェックボックスか有効化されている場合、 MapForce は、コンポーネントにより参照されるファイルを記録します。また、全てのファイルがマンピングと同じディレクトリに存在する場合、 ディスク上の新しい場所にディレクトリ全体を移動しても、マの参照か壊されることしまかりません。

相対パスを使用することと「MFD ファイルに相対的なすべてのファイルパスを保存する」チェックボックスを有効化)は多くの場合とても 重要です。例:

- オペレーティングシステム上のマッピングのロケーションが将来変更される可能性のある場合。
- (SVNなどのデジョンエトロールシステムを使用してマングがソースコトロール化にあるディレクトリに保管されている場合。
- 異なオニマシノニ、おけま異なオ・オペレーティングシステムニマパングを実行のためニデプロイする場合。

「MFD ファイルニ相対的なすべてのファイル・マを保存する」チェックボックスが無効化されている場合、マッピングの保存は、ファイル・マトを変更しません(すなわち、コンポーネント設定ダイアログボックス内で表示されるままコンリます)。

4.1.2 ファイルベースのデータベースへパスを設定する

Microsoft Access 封はSQLite マピングなどのデータベースファイルを追加する場合、(データベース接続ウィザードの開始を参照)、絶対パマの代わりに相対パマを使用します。相対パマを使用するには、「データベース接続ウィザード」内の「参照」をクリックする代わり に必要な相対パマを入力します。

相対ファイル ジを入力する前に、マンピングファイル(.mfd)を最初に保存してくたさい。それ以外の場合、全ての相対パンはの Windows のペーソナルアプリケーションフォルダーニ対して解決されます(ドキュメント Witova WapForce 2021)これは、予期しない 振る舞しの場合かあります。

🕑 データベースを選択	र 🗾
よう 接続ウィザード	SQLite に接続する
既存の接続	ブラウザーをクリックして SQLite データベース を選択する。[次に]をクリックしてデー タベースに接続する。
mm ADO接続	mydbfile.eqlite 参照
ODBC 接続	
) JDBC 接続	
グローバル リソー マ	
~	<前へ(B) 接続(C) 閉じる(L) //

データベース接続ウィザード

データベースがSQLite データベースの場合、以下の条件がtrue の場合、「接続」ボタンが有効化されます

- マピング(.mfd) ファイルマオして相対的に解決されることのできるファイルをパメコポイントしています。
- 参照されるファイルはSQLite データベースです。

マピング内に既存のデータベースの、マエレポーネントを変更するコよ、以下を行います

- 1. データベースエポーホトのヘッダーを右クルプして、「プロパティ」(<u>エレポーホト設定を変更する</u>も参照を選択します。また は、エレポーホトタイトルトーをダブルクルプします。
- 2. コーポーネト設定ダイアログボックスから「変更」をクリックします。

コンポーネント設定	<u>-</u>	×
データベース データソース: 接続名: データベースの種類:	⁻ orce2015¥MapForceExamples¥people.sqlite people SQLite	変更(C)
接続文字列:	C:¥Users¥altova¥MapForce2015¥MapForceE people.sqlite	xamples¥ 🔺

これにより、ハマを含むデータベース接続プロノティを変更することのできる、データベース接続ウィザードが再度開かれます。

「接続文字列」は、常に絶対パマを含むことに注意してくたさい。コンポーネント内の構造情報のためコはデータベースが使用されています。 「データノース」内の相対パマレ、コンポーネントが相対ファイルリマと共に作成されたことを示します。

メモ プログラムコードを生成する際に、MapForce Server 実行ファイル(.mfx)をエンパル、おけよ FlowForce Server にマピ ングをデプロイする際に、マピング設定かび生成されるコード内でパマを絶対パマルこする」が選択されている場合、相対パンが 絶対パマニ変換されます(生成されたコード内の、マノニントでを参照)

4.1.3 破損したパスの参照を修正する

マピング内でファイルレファレンスを追加ませる変更する場合に、なか解決されない場合、MapForce は、警告メッセージを表示します。こうすることに以、MapForce は、破損したいなへの参照を削減します。しかしなから、以下の場合破損したいなへの参照が起こる可能性があます。

- 相対パマを使用し、マンピングファイルをスキーマとインスタンスファイルを移動することなく、新しいディレクトリン移動する場合。
- マッピングファイルと同じディレクトリ内のファイル絶対パマを使用し、ディレクトリを他の場所に移動する場合。

このような状況か発生すると、MapForce はこのエレポーネトを赤で イライトします。例



破損したいの参照

この場合の解決方法は、コンポーネトへッダーをダブルクトックして、「コンポーネント設定」ダイアログボックス内で破損した。マを更新します(コンポーネト設定を変更するも参照)。

4.1.4 多種の実行環境内のパス

コードをマルングから生成し、MapForce Server 実行可能ファイル(.mfx) に対してマルングをエンパイルする場合 生成されたファイル は、MapForce で実行することはできません。選択されたターゲト環境で実行できるようになります。(例: RaptorXML Server、 MapForce Server、おとて、アプリケーション)。マルングの実行に成功するこは、全ての相対的ないなはマルンノが実行される個所 で意味をなす必要があます。

従い、インスタンスまたはスキーマファイルに対してマッピングが相対パンを使用する場合、各ターゲット言語のベースパンに関しては以下のテーブルを考慮してくたさい

ターゲナ言語	ネーシュ		
XSLT, XSLT2, XSLT3	XSLT ፓግቢው ዪ		
XQuery*	XQuery ፖንኅルወን ኳ		
C++、C#、Java*	生成されたアプリケーションの作業ディレクトリ		
BUILTHN*(MapForce でマピングをプビューする場合)	マンング (.mfd) ファイルの &		
BUILTHN*(MapForce Server でマピングを実行する場合)	現在の作業デルケトリ		
BUILTHN*(FlowForce Server の管理下で、MapForce Server でマピングを実行する場合)	ジョブの作業ディレオリませまFlowForce Server の 作業ディレオリ		

*MapForce Professional とEnterprise エディションで使用可能な言語

必要な場合、マピングのナガニードを生成する際、MapForce に全ての やを相対から絶対パスニ変換するように命令することができます。このオプションは、マピングコードを、同じオペレーティングシステム内(おけはMapForce Server 実行ファイル)で、おけはマピングによ り使用される絶対パスへの参照かある他のオペレーティングシステムで、実行する際に役に立ちます。

生成されたコード内で全ての、父を絶対、ツレ変換する方法

- 1. マピングの空のエリアを右クリックします。マピングの設定ダイアログボックスか表示されます。
- 2. 「生成されたコードでは絶対パスを使用する」チェックボックスを選択します。

コードを生成する際、チェックボックスが選択されていると、MapForceは、マンピングファイル(.mfd)のディレクトリをベースはた相対パマを解決し、生成されたコード内でこれらを絶対パスにます。この設定は、以下のファイルのパイン影響を及ぼします。

- すべてのファイルベースのコンポーネント型のための入力と出力インスタンスファイル
- マピングエポーネトとて、使用される SQLite データベースファイルへアクセス

チェックボックスが選択されていない場合、ファイルマは、コパーネト設定内で定義されたとお保存されます。

生成されたコード内のライブラルマ

マピングファイルコオプションで多種のライブラレヘの や参照が含まれている場合があります。例えば、マピングファイルは他のマピングまた はカスタムXSLT、XQuery*、C#*、またはJava* ライブラリまたは、mff*(MapForce 関数) ファイルからの関数からのユーザー定義 関数をインポートする場合があります。
<u>関数ライブラレの管理</u>も参照してくたさい。

*MapForce Professional とEnterprise エディションで使用可能な機能

オプション「生成されたコード内でいたを絶対いなにする」はマンピングコンポーネントのみに適用され、外部ライブラレクのなる影響を与えません。代わりにXSLTとXQuery以外のすべてのライブラルのためこライブラルなけ解決され、生成されたコード内で絶対いなご変換されます。例えば、マンピングファイルに、NET、dll おけましょい。ことのアイルは含まれ、生成されたコードを他の環境(例えば他のコンピューターで)で実行する場合、参照されるライブラルなーゲト環境内の同じ、なご存在する必要があります。

XSLT おけはXquery ファイルをマンプから生成する場合、生成されたXSLT またはXquery ファイルに相対的なライブラルマを以下のように作成することができます。

- 1. マピングの空のエリアを右クトックします。マピングの設定ダイアログボックスが表示されます。
- 2. チェックボックス「XSLT / Xquery ファイルロン相対的なパマを持つレファレンスライブラリ」を選択します。

上のチェックボックスを選択するとXSLT おけはXquery ライブラノファイルは実際にその、マルニ存在することを確認してくたさい。

XSLT お台はXquery ライブラリファイルへの なを生成されたコード内で絶対、なにするコはこのチェックボックスをクリアします。

4.1.5 コピーと貼り付けと相対パス

マピングからエンポーネントをコピーし、貼り付ける場合は、スキーマファイルの相対パンを目的のマピングのフォルダーマ対して解決することができます。もし、パンが解決されない場合、相対パンを絶対パンパースマピングのフォルダーを使用して変更するようにプロンプトされます。最初に目的のマピングに保存することが奨励されます。それ以外の場合、相対パンス、個人のアプリケーションフォルダーマ対して解決されます。

4.2 マッピング接続の種類

マピング接続を作成すると、オプノョンで接続、子アイテムを持つノースとターゲトアイテムの種類を以下から選択することができます。

- ターゲト優先(標準)
- ソース優先(混合コンテンソ)
- 全てコピー(子アイテムのコピー)

接続の種類はマンピングにより生成された出力内の子アイテムのシーケンスを決定します。このセクションは各接続の種類と役に立つシナリオについての情報を提供します。

4.2.1 ターゲット優先マッピング

ターゲナ優先(標準マピング)は、MapForceにおったデフォルトのマピング方法で、ターゲナスキーマ内のノードのシーケンスにお出力 が決定されます。このマピングの種類は大部分のシナリオに適しており、MapForceのデフォルトのマピングとして設定されています。

標準マピングは実線で表示されます。

ķ	() Name	└> 檀淮	4	()Name	4
ķ	🕀 🗘 Office	175-1		🕀 💙 Office	¢
¢	() Name	•	 	A Name	þ
¢	🕀 🔇 Desc	¢	Ļ	🕀 🗘 Desc	¢

ターゲナ優先マピングは文字データや子要素などの複合コンテキストを含むXMLノードをマップする場合適していない場合があります。 例:

This is our <i>best-selling</i> product.

複合コンテンツが含まれるマンピングの場合、ソースファイル内で現れるアイテムのシーケンスを保持ため、ソース優先マンピングを使用することが 奨励されす(ソース優先マンピングを参照)。

4.2.2 ソース優先マッピング

ソース優先(混合コンテンツ)マンピングを使うことで、テキストならびに子ノードを、XML ソースファイルにある順序通りにマンピングすることができます。

- 混合コンテンソテキストノードコンテンソがサポートされます。
- 子ノードの順序はノースXML インスタンスファイルに依存します。

ķ	🕀 () Desc	۲ <u></u>	> 🕀 🗘 Desc 🗦
¢.	🕀 🌔 para		🕨 🖓 para 🛛 🖓
¢.	Rbctext0	I	Abctext0 [>
l 🖒	()bold		bold 🖓
ķ	() italic	t	italic 🖓

混合コンテンツのマッピングは点線により表示されます。

勿論、ソース優先 / 混合エンテンソマメピングを XML スキーマの複合型に適用することもできます。 XML ソースファイルの順序に従って子ノ ードのマメピングは行われます。

ソース優先 / 混合コンテンソマピングでは以下がサポートされます:

マピング

- **ソースコンポー**ネトとして:
 - XML スキーマ複合型(mixed=true どこっている混合コンテンソを含む)
 - データベースフィールドへ組み込まれたスキーマ内にある、XML スキーマ複合型(混合コンテンンを含む)
- ターゲットコンポーネトとして:
 - XML スキーマ複合型(混合コンテンンを含む)
 - データベースフィールドへ組み込まれたスキーマ内にある、XML スキーマ複合型(混合コンテンンを含む)

メモ CDATA セクションはデキストとして扱われます。

4.2.2.1 混合コンテンツのマッピング

以下のサンプルでは….\<u>MapForceExamples\Tutorial\</u>フォルダー以下にあるTut-OrgChart.mfd、Tut-OrgChart.mfd.xml、Tut-OrgChart.mfd.xsd, Tut-Person.xsd を使用します。

ソースXML インスタンス

このセクションで使用されている Tut-OrgChart.XML ファイルの一部を以下に示します。ここで注目したいのは「bold」ならびに「italic」子ノードも含まれている「para」の混合コンテンツです。

para 要素には処理命令(<?sort alpha-ascending?>)だけではな、コメントテキスト(<!--Company details... --->)も含まれていることに注目してくたさい。これらのデータセマメングすることができます。



XML インスタンスファイルにあるテキストのシーケンスならびに bold/talic ノードは以下のような構成(順序) どなっています:

<para> The company... <**bold**>Vereno</**bold**>in 1995 ... <**italic**>multi-core...</**italic**>February 1999

```
<bold>Nano-grid.</bold>The company ...
```
<italic>offshore...</italic>to drive... </para>

初期のマピング

Tut-OrgChartmfd を開いたきのマルングを以下に示します。



上のマピングを出力

上に示されるマノピンク結果を以下に示します。組織図や個々のオフィス名が表示されます。

1	xml version="1.0" encoding="UTF-8"?
2	E <company-person td="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nonames<=""></company-person>
3	<name>Organization Chart</name>
4	
- 5	<name>Nanonull, Inc.</name>
6	
- 7	🔶 <office></office>
8	<name>Nanonull Europe, AG</name>
9	
10	
11	

para 要素のマピング

以下のスクノーンショナは混合エンテンソマッピングの例です。混合エンテンソのpara要素にコネクタが接続されており、点線によりそれが混合エンテンソであることを表しています。text()ノードにコよテキストデータが含まれており、ターゲナトコンポーネントへテキストを表示するコよこのノードをマッピングする必要があります。



コネクタを右クトックしてプロッティを選択することで、コネクタビ注釈を追加し、ラベルを付与することができます。詳細については<u>接続の注釈</u>を参照ください。

以下のイメージはTut-OrgChart.xsd スキーマファイルにあるDesc 要素のエレテンソモデルを表しています。この例の場合、この定義は ソースならび、マターゲトスキーマの両方にて共通したものとなっています。



コンテンツモデルにある para 要素について、以下の事柄を理解できます:

- para は「TextType」とう型で、mixed="ture"の複合型です。
- **bold** ならびにitalic 要素は両方とも "xs:string" 型となっており、この例では再帰的な型として定義されていません(つまり bold も italic も "TextType" 型ではおりません)。
- bold ならびにitalic 要素は、para内部においてどのような順序でも無制限の数だけ出現することができます。
- para 要素内では bold ならびに italic 要素ともに、任意のテキストを入力することができます。

アイテム間で混合コンテンソの接続を作成:

- 1. メニューオプションの「接続 | 子要素の自動接続」を選択して、有効にします(既に有効になっていない場合)。
- 2. ソーススキーマースこて para アイテムから、ターゲオーマの para アイテムへの接続を行います。MapForce にコネクタをノース 優先にするかを問うメッセージが表示されます。

MapForce					
混合コンテンツを持った2つの要素を接続しました。このような場合、接続の種類はソース優先で、text() ノードが互いに接続されているべきです。					
このような接続に変更しますか?					
□ 次回からこのメッセージを表示しない					
(はい(Y) いいえ(N)					

- 3. 「はりをクリックして混合コンテンツ接続を作成します。
- メモ para は混合エンテンソであるため、この時点でメッセージが表示されます。自動接続オプションを無効した状態で、para アイテムを直接接続した際にも混合エンテンソのメッセージが表示されます。

para アイテムこある全ての子アイテムか接続されました。para アイテムへ接続しているコネクタか点線で表示され、それが混合コンテンツであることを表しています。

4. 出力ダブをクリックして、マッピングの結果を確認します。



5. 出力ダブのアイコンバーコあるワードラップアイコン となりシグして、全てのテキストを出力ウィンドウェて表示します。

3 <name>Organization Chart</name>
4
5 <name>Nanonull Inc </name>
5 Charles Charles Charles Charles
7 Constant The company was established in shelds. Verenos (holds in 1995, Nenonull develops)
Spansal and the company was established in the location of
handleterroring technologies to attain while one processors. All and whether the same at the
unveiling of the first prototype <bookstands and="" company="" e="" expand="" hopes="" in="" its<="" stands="" td="" to=""></bookstands>
operations <italic>offshore</italic> to drive down operational costs.
8 −
9 < <p><</p>
10 -
11 -
12 -
13 🖕 <office></office>
14 <name>Nanonull Europe, AG</name>
15 🖕 <desc></desc>
16 spara>In May 2000, Nanonull <italic>Europe</italic> was set up in Vienna. The team
consists of <body <="" body="" five="" research="" scientists="">and one administrative staff.</body>
17 - «Desc»
18 - (Office)
ao soonipany-reisona
20

各オンイス・関する記述である混合コンテンソテキストが正しくマッピングされました。テキストだけでなく bold ならびに italic タグコン テンツが、XML ソースファイルにある通りにマッピングされました。

6. マピングビューイン切り替えます。

混合コンテンツからテキストノードを削除する:

1. text() ノードのコネクタをクリックして、Del キーを押下することで接続を削除します。



2. 出力ダブをクリックしてマッピングの結果を確認します。



結果:

- para 要素内にある全てのテキストノードが削除されまた。
- それまでマピングされていた bold ならびに italic 要素のテキストはまた残っています。
- bold ならひにitalic アイテムのシーケンスもソース XML ファイルと同様のものとなっています。

処理情報とコメントのマッピング:

- 1. コンテンソ接続を右クトックして、プロパティを選択します。
- 2. ソースドライブ(混合コンテンソ)から、処理命令をマップとコメントをマップのチェックボックスを選択します。

4.2.2.2 混合コンテンツのサンプル

以下のサンプルでは MapForceExamples フォルダー以下にある ShortApplicationInfo.mfd を使用します。

この例で使用するXML ソースファイルの一部を示します:

_	
Γ	<page <="" td="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></page>
ŀ	xsi:noNamespaceSchemaLocation="SectionedPage.xsd">
l	<item></item>
l	<title>XMLSpy</title>
l	<mainsection author="altova"></mainsection>
l	Altova <trademark>XMLSpy</trademark>
l	SubSection>Altova <trademark>XMLSpy</trademark> 2005 Enter
ŀ	is the industry standard <keyword>XML</keyword> development environment
ŀ	editing, debugging and transforming all <keyword>XML</keyword> technolo
ŀ	automatically generating runtime code in multiple programming languages
l	
l	

以下にマピングを示します。以下の点に注意してくたさい

- 「SubSection」アイテムのロネクタは混合コンテンソで、ターゲオ XML/スキーマの詳細 アイテムへマピングされています。
- text()ノード同士がマピングされています。
- Trademark テキストがターゲナのBold アイテムへマピングされています。
- Keyword テキストがターゲットのItalic アイテムへマピングされています。



マピングの結果

各説明文の混合コンテンソテキストが正しくマソピングされ、テキストだけではなく、bold とitalic タグコンテンソがXML ソースファイルにある順序の通り、正しくマソピングされます。

1	xml version="1.0" encoding="UTF-8"?
2	Shortinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:noNamespaceSchemaLocation="
	C:/PROGRA~1/Altova/MapForce2005/MapForceExamples/ShortInfo.xsd">
3	⇔ <info></info>
4	<title>XMLSpy</title>
5	<description>Altova <bold>XMLSpy</bold> 2005 Enterprise Edition is the industry standard</description>
	<talic>XML</talic> development environment for modeling, editing, debugging and transforming
	all <ttalic>XML</ttalic> technologies, then automatically generating runtime code in multiple
	programming languages.
6	-

4.2.2.3 混合コンテンツアイテムに対して標準接続を使用する

上記のように、ソース優先の(標準ではない) 接続は、通常、複合コンテンソノードからデータをマルングする際に使用されます。 それ以外の 場合、結果する出力は、希望する結果ではない可能性があります。 通常の標準(ターゲメ優先)接続を使用して、複合コンテンソノードか らデータをマルングする場合の結果を確認すること、以下の手順を踏んでください

- 1. <u>〈マイドキュメント〉 料ltova ¥Map Force2021 Wap ForceExamples ¥Tutorial</u> ジオルダーからマピング Tut-OrgChart.mfd を開きます。
- 2. ソース内のpara ノード とターゲナト内のpara ノード 間に接続を作成します。MapForce が接続をノース優先と定義するか を問うメッセージが表示されます。「しいえ」をクリックします(これにより、MapForce による提案を破棄し、通常の接続を作成 します)。



- メモ 接続が上て表示されているように標準(ターゲナト優先)であることを確認してくたさい。
 <u>すべてこピー</u>接続が自動的に作成される
 と接続を右クリックして、コンテキストメニューから「ターゲット優先」(標準)を選択します。
 - 3. 「出力」ダブをクリックして、マッピングの結果を表示します。



上に示されるように、混合コンテンツのアイテムに対して標準マッピングを適用すると、以下のような結果となります:

- text() ソースアイテムのエレテンパターゲルにエピーされます。しかしなから、出力内の子ノード(この場合、bold と italic) のシーケンスはターゲル XML スキーマ内のシーケンスに対応します。すなわち、子ノード(この場合、bold と italic) は、複合コンテンソノードテキストの後に表示されます。
- それぞれのpara 要素に対して、MapForce は、text()ノードを最初にマップし、すべてのbold アイテムと、最後に全てのitalic アイテムをマップします。この結果、複数のboldとitalic アイテムは、互いに積み上げられて表示されます。各アイテムのエンテンソは、接続がノースから存在する場合に表示されます。

4.2.3 全てコピー接続

全てコピーという種類の接続方法では、作業領域にあるマンピングをより単純なものとして、ソースならびにターゲトコンポーネントにある全ての同一アイテムを自動的に接続します。つまり、ソースならびにターゲトコンポーネントにおける型に従うかたちで、ソースならびにターゲトアイテムの型が同一の場合か、ませ」はターゲットの型がxs:anyTypeの場合、該当する全てのノースアイテムがターゲットコンポーネントへコピーされます。

ソースならび、こターゲトの型か同一で無い場合、かつターゲトの型がxs:anyTypeで無い場合、ソースデータは同一の階層にある同一名のアイテムこマピングされます。ターゲトアイテムの名前か異なる場合、そのターゲトアイテムこを対するマピングは作成されません。



全てコピー接続

マルング上の2つの構造間にマルングを描く場合シースとターゲオ構造に互換性があると検出した場合、MapForce は、「全てコピー」 接続を自動的に作成します(これは、両方の構造が同じ型、おけおターゲオがノース型のサブ型であることを意味します)。マルングのラン タイムでは、子を含む全てのインスタンスデータはノースからターゲオトに再帰的にコピーされます。

「全てコピー」接続を手動で作成するコよ、子アイテムを持つ同類の2つのノード間の既存の接続を右クトックし、コンテキストメニューから全て コピー(子アイテムのコピー)を選択します。

次に注意してくたさい

- 「全てコピー」接続か意味を成され、おさまサポートされていたいマンテキストでは、この種類の接続を手動で作成することはできません。
- 「全てコピー」接続をXML/スキーマコンポーネントのroot 要素に対して作成することはできません。
- ユーザー定義関数のスキーマと、ラメーター間に「全てコピー」接続を作成する場合、2つのコンポーネトは同じスキーマをベースにしている必要があります。しかしなから、同じルート要素を持つ必要はありません。

「全てコピー」接続の例は次のステップで作成することができます

- 1. 新規マンプを作成します。
- 2. 「挿入」メニューカらXML スキーマ/ファイルをクリックし、フォルダー <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\内にある books.xml ファイルを参照してくたさい。
- 3. 「挿入」メニューからXML スキーマ/ファイルをクリックし、フォルダー <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial、内にあるlibrary.xsd ファイルを参照してくたさい。
- 4. 「books」 コパーネントのbook ノードと「library」 コパーネントのpublication ノード間にマピング接続を描きます。
- 5. 新規の接続を右クリックして、コンテキストメニューから全てコピー(子アイテムのコピー)を選択します。



ソースとターゲオ構造間に差異が存在する場合、「全てコピー」接続が列挙します。マッピングのランタイムでは、(要素と属性などの)ソースアイテムは、ターゲメ型内に存在するものだけがコピーされます。これは再帰的に繰り返されます。

例えば、上のマンピングでは、2つの構造(authorとtitle)の2つの子アイテムが同一です、ですから、ターゲットにマップすることができます。アイテムidは、ソース属性内の属性とターゲット内の要素であるため、自動的には含まれません。マップする必要かある場合、例えば、categoryからgenre、ヘマップする場合、異なるアイテムのため、「全て二ピー」接続を使用することはできません。

入力コネクタがコンポーネントの横の小さ、三角で表示されている「全てコピー」接続を受け取る場合、他の接続を受け入れることはできません。上のサンプルでは、categoryとgenre間の接続を作成する場合、MapForce置換え、おとは入力の複製を行うようにプロンプトレます。

MapEorce	×					
mapi orce	^					
This input connector has already a connection assigned to it via a copy-all connection. Only one connection can be defined per input connector. What action should be taken?						
Replace Connection Duplicate Input	Cancel					

ここでは必要みませんが、入力の複製は、ターゲルが「つ以上の入力からデータを受け入れる場合に役に立ちます(次を参照: 入力の複製)。「全てコピー」接続を置き換える場合、メッセージボックスは、「全てコピー」接続を解決、ませば削除するようにプロンプトします。

MapForce				×			
Connecting child nodes, What action	Connecting 'publication' requires to either resolve the copy-all connection to 'publication' and to keep all connections to the other child nodes, or to remove the copy-all connection including all its child connections. What action should be taken?						
	Resolve copy-all connection	Delete child connections	Cancel				

「全てコピー」接続を対応する子アイテムの標準のターゲオ優先接続と置換えるコよ、全てコピー接続を解決するをクリックします。「全てコピー」接続を削除するコよ「子接続の削除」をクリックします。

4.3 チェーンマッピング

MapForce ではチェーン構造により複数のコンポーネントを含むようなマンピングがリポートされます。チェーン構造のマンピングとよ 少なくとも 1 つのコンポーネントがソースかつターゲットコンポーネントとして機能するマンピングのことです。このようなコンポーネントの出力は、それ以降の マンピングで入力として使用されます。このようなマンピングのことを「中間」コンポーネントと呼びます。

例えば、下に示されるマピングは、(XML 書式の)2つの段階で処理される経費報告書を表示しています。A からB へのマピングの部分 は、「Travel」とマークされる経費のみをフィルターします。B からC へのマピングの部分は、1500 以下の「Travel」経費のみをフィルタ ーします。コンポーネント B は、入力と出力接続の両方が存在するため、「中間」コンポーネントです。このマピングは、次の、次で見つける ことができます: <マイドキュメント > #Itova #Map Force 2021 #Map Force Examples #Tutorial #Chained Reports.mfd。



ChainedReports.mfd

「パマスルー」と呼ばれる機能がチェーンマンピングに搭載されています。「パマスルー」は、出力ウイドウ内でチェーンマンピングの各段階による 出力を確認することのできるプレビュー機能です。例えば、上のマンピングでは、A からB への および、B からC への 結果する XML 出 力をプレビューして、保存することができます。

メモ XML、CSV、TXT ファイルといオファイルをベースした「中間」コンポーネントだけが「リウスルー」機能を搭載します。データベー スコンポーネントを中間コンポーネントとして使用することはできますが、リウスルーコンポーネントは表示されません。 プレビューやコードの生成を行うたび、ロー間コンポーネントは再度生成されますが、生成を行う前に一度削除しなければならないサムシ、データベー スコンプンジン、中間コンポーネントは再度生成されますが、生成を行う前に一度削除しなければならないサムシ、データベー スコンプンジン操作を行うことはできません。

MapForce Server おけよ、生成されたコードによりマピングが実行される場合、マピングのチェーン全体が実行されます。チェイン内の 各ステップで必要な出力ファイルをマピングは生成し、マピングのチェーンのステップの出力が次のマピングステップへ入力として転送されま す。

中間コンポーネトのために、動的なファイル名を生成することもできます。コンポーネトが対応して構成されていることを条件に、マンピングから "File: "アイテムへの接続を受け入れることができます。 更に詳しい情報に関しては、次を参照してくたさい、 複数の入力、 および、 出力ファイルを動的に処理する。

プレビューボタン

コンポーネント B ならびにC にはプレビューボタンカ表示されます。この機能を使用することにより MapForce 内で B の中間マピング結果のプレビューだけではなく、チェーン構造の最終結果をプレビューすることができます。対応するコンポーネントのプレビューボタンをクルクス、 出力ボタンをクルクすることでマピング結果を確認することができます。 ノなスルーポタンが有効化されている中間コンポーネントをプレビューすることはできません。同時にデータをプレビューし、ノなスルーすることは 意味を成されため、プレビューポタンは、自動的に無効化されます。このようなコンポーネントの出力を確認するためには、ノなスルーポタンを クトックして、無効化してから、プレビューポタンをクトックします。

→ パススルーボタン

中間コンポーネント B には、更に「パススルー」とう名前の状かがコンポーネントのタイトル いーと表示されます。

パマスルーボタンの有効 🔁 になってしる場合、エポーネイ A からエポーネイ B へ そしてエポーネイ C へとうかたちで全てのデ ーダをプレビュー・ウィズ・ウヘマ・ピングします。2 つの結果が生成されます。

- マピングエポーネト A の結果が中間エポーネト B ヘマピングされます。
- 中間コンポーネント B の結果がターゲトコンポーネント C へマンピングされます。

パマスルーボタンが無効 ➡ してょっている場合、マッピングチェーンの一部だけが実行されます。生成されるマッピング結果は、(コンポーネント BとCのかち)とちらのプレビューボタンがアクティブ てょっているかに依存します。

- エレポーネント B のプレビューボタンのアクティブ なってい る場合、エレポーネント A から B へのマッピング結果が生成されます。マッ ビングチェーンはエレポーネント B で停止し、プレビューを行うの ニエレポーネント C は全く使用されません。
- エンポーネント C のプレビューボタンのアクティブ こなってし る場合、中間エンポーネント B から C へのマンピング結果が生成されます。 パスパレー機能が無効になってしなけっか、コンポーネント B に対する自動チェーンが中断されます。 マンピング右側にあるチェーン だけが実行され、コンポーネント A は使用されません。

¹パマスルー⁻¹ ボタカ無効化されている場合、中間コンポーネントは、「入力 XML ファイル」と「出力 XML ファイル、フィールド内で同 ーのファイル名を持つ必要があります。これにより、B とC の間のマメピングの部分をプレビューする際に、出力として生成されるファイル がA とB の間のマメピングを入力と使用することができます。また、生成されたコード内で、ませよ MapForce Server 実行内で、 マメピングのチェーンが壊れないことを保証することができます。

前記のようこ、マピング MapForce Server、おけさ生成されたコードにお実行された場合、すべてのコンポーネントの出力が生成されます。この場合、コンポーネント B の マスルーボタン、および現在選択されているプレビューコンポーネントは破棄されます。上のマピングを 例 コンて、2つの結果ファイルは以下のように生成されます。

- 1. A からB へのマピングコンポーネントから結果する出力ファイル
- 2. B からC へのマピングコポーネトから結果する出力ファイル

次のセクションである。例、パススルーか有効な場合と例、パススルーか無効な場合は、パススルーポタンか有効化、おけは無効化されている。 場合に、どのようにノースデータが転送されるかについて説明しています。

4.3.1 例: パススルーが有効な場合

このサンプルで使用されて、るマピング(Chained Reports.mfd)は、くマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\フォルダー内におります。このマピングは、旅費の経費を含む ReportA.xmlとう名前のXMLファイルを処理します。簡素化のために、名前空間宣言とexpense-item要素の一部は省略されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<expense-report currency="USD" detailed="true">
<Person>
<First>Fred</First>
<Last>Landis</Last>
<Title>Project Manager</Title>
```

```
<Phone>123-456-78</Phone>
      <Email>f.landis@nanonull.com</Email>
  </Person>
   <expense-item type="Travel" expto="Development">
     <Date>2003-01-02</Date>
      <Travel Trav-cost="337.88">
         <Destination/>
      </Travel>
      <description>Biz jet</description>
  </expense-item>
   <expense-item type="Lodging" expto="Sales">
     <Date>2003-01-01</Date>
      <Lodging Lodge-cost="121.2">
         <Location/>
      </Lodging>
      <description>Motel mania</description>
  </expense-item>
  <expense-item type="Travel" expto="Marketing">
      <Date>2003-02-02
      <Travel Trav-cost="2000">
         <Destination/>
      </Travel>
      <description>Hong Kong</description>
   </expense-item>
</expense-report>
```

ReportA.xml

マピングの目的は、上のファイルを基にして、2つの更なるレポートを生成することです:

- **ReportB.xml**-このレポートは、"Travel"型の旅費のみを含みます。
- ReportC.xml このレポートは、1500 を超えない"Travel"型の旅費のみを含みます。

この目的を達成するためコよマピング(コンポーネント B)の中間コンポーネントでは、表示されているとおり、マスルーボタン・ か有効 化されています。これによりマピングは段階的に実行されます: A からB は、そして、B からC へ実行されます。中間コンポーネントにより 作成される出力は、B とC の間のマピングのために使用されます。



マピングのチェーン内の各段階で生成される出力ファイルは、各コンポーネントの設定により決定されます。(コンポーネント設定を開くには、 右クリックし、コンテキストメニューからプロパティを選択します)。具体的には、最初のコンポーネントは、ReportA.xml とう名前の XML ファイルからデーダを読み取るように構成されています。ソースコンポーネントであるからであり、出力 XML ファイルフィールドは関連性 がないため、空白のままです。

コンポーネント 名: ReportA	
スキーマ ファイル(F)	
ExpenseReport.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
ReportA.xml	参照(B) 編集(E)
出力 XML ファイル(O)	
	参照(R) 編集(D)

ソースコンポーネントの設定

下に示されるように、2番目のエンポーネント(ReportB)は、ReportB.xml とう名前の出力ファイルを作成するようご構成されて、ます。入力 XML ファイルフィールドは、灰色で表示されています。(このサンプル内で示されているように)、タスルーカ有効化されている場合、中間エンポーネントの入力 XML ファイルフィールドは、自動的に無効化されます。マンピングを実行するために入力ファイル名か存在する必要はよく、マンピング内のこの段階で生成される出力は、一時的なファイル内に保管され、マンピングの後の段階で再利用することができます。また、(下で示されるように出力 XML ファイルが定義されている場合、中間出力ファイルのファイル名のために使用されます。出力 XML ファイルが定義されていた。場合、デフォルトのファイル名が自動的に使用されます。

コンポーネント名:	ReportB		
ースキーマ ファイル(F)			
ExpenseReport.xsd		参照(W)	編集(T)
 入力 XML ファイル(I)			
ReportB.xml		参照(B)	編集(E)
出力 XML ファイル(O)			
ReportB.xml		参照(R)	編集(D)

中間コンポーネントの設定

最後に、3番目のエンポーネントは、ReportC.xmlとう名前の出力ファイルを生成するように構成されています。これはカーゲトコンポー ネントであるナムシ、入力 XML ファイルフィールド には期連性はありません。

コンポーネント名: ReportC	
スキーマ ファイル(F)	
ExpenseReport.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
	参照(B) 編集(E)
出力 XML ファイル(O)	
ReportC.xml	参照(R) 編集(D)

ターゲオコンポーネントの設定

マピングウィドウ内の出力ダブをクリックして、マピングをプレビューする場合は、2つのファイルが出力に表示されます:

- 1. ReportB.xml は A からB へのマピングの結果を表します。
- 2. ReportC.xml は B からC へのマンピングの結果を表します。

2件の生成された出力ファイルから選択するコよ、出力ウイドウ内に表示されているファイルを矢印ボタンをクリックして、おコボロップダウンリストからエトリを選択します。

•	プレビュー 1 /1(2) 🛛 📾 C:¥Users¥t.a.yasaka¥Documents¥Altova¥MapForce2017¥MapForceExamples¥Tutorial¥ReportC.xml
1	xml_version="1.0" e 🔜 0.¥Users¥Documents¥Altova¥MapForce2017¥MapForceExamples¥Tutorial¥Report0.xml</td
2	<expense=report reportb="" td="" xsing="" からの中間出力<="" コンポーネント="" 倒=""></expense=report>
_	http://www.w3.org/200 📾 Report B: C:¥Users ¥Documents¥Altova¥MapEorce2017¥MapEorceExamples¥Tutorial¥Report B xml
3	
4	
D	
7	
6	
a u	
10	
11	
12	
13	Setimation >
14	
15	<pre><description>Biz iet</description></pre>
16	-
17	Keypense-item type="Travel" expto="Accounting">
18	Date>2003-07-07
19	
20	Contraction Contractic Contraction Contractic
21	
22	Security Constraints and Co
23	<pre></pre>
24	-
20	

生成された出力ファイル

MapForce によりマピングが実行される場合、(「ツール | オプション | 一般」から構成することのできる設定「出カファイルに直接書き 込む」により、中間ファイルが一時的なファイル、おけよ物理的ファイルとして保存されるかを決定します。MapForce 内でマピングが道接 プレビューされる場合のみ、これは妥当であることに注意してくたさい。このマピングがMapForce Server により、おけよ生成されたコード により実行された場合、実際のファイルは、マピングのチェーンの各段階で生成されます。

StyleVision がインストールされている場合、そして、StyleVision Power Stylesheet(SPS)ファイルカターゲトコンポーネントに (このサンプル内で示されているようこ)割り当てられている場合、最後のマンピング出力をHTML、RTFファイルとして確認(保存)すること ができます。MapForce内でこの出力を表示するコス、対応する名前を持つタブをクトックしてくたさい。



Personal Expense Report					irrency:		lars O Euros	s 🔾 Yen	Currency \$	
					✓ Detailed report					
Employee Information										
	Fred Landis First Name Last Name				Project Manager Title					
	f.landis@nanonull.com E-Mail					123-456-78 Phone				
Expense	e List									
Тур	e	Expe	nse To	Date (yyyy-mm- dd)	I	Expense	s \$	Descr	iption	
Travel	~	Develop	ment 🗸	2003-01- 02	Trav 337.8	rel La 38	odging	Biz jet		
Travel	~	Account	ing 🔽	2003-07- 07	Trav 1014	r el L .22	odging	Ambassad	lor class	
Mapping	DB Quer	y Out	put 🚺 🛛 H	TML ORT	= <mark>0</mark> 1	PDF (Word 200	17+		

生成された HTML 出力

マピングのチェーンの最後のターゲトコンポーネントの出力のみが表示されることに注意してくたさい。中間コンポーネントのStyle Vision出力を表示するコよノマスルーポタンを無効化して中間コンポーネントを(例ノマスルーが無効な場合内で表示されているとおりプレビューする必要があります。

4.3.2 例: パススルーが無効な場合

このサンプル(Chained Reports.mfd) で使用されているマピングは くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\フォルダー内で見つることができます。このサンプルは、中間コンポー ネントの マスルーポタン シ が無効化されている場合、出力がどのように生成されるかを説明しています。



例 パススレーが有効な場合で説明されているように、マナングの目的は、2件の個別のレポートを生成することです。前のサンプルで、パススレーボタン→ は有効化されており、両方のレポートは期待されるとおり生成され、出力 タブ内で確認することができました。しかしながら、レポートの1件(ReportB.xml ませま ReportC.xml)のみをプレビューする場合、パススレーボタン(→)は無効化されている必要があります。更に具体的には、パススレーボタンを無効化することは、次の目的を達成するためには有効な場合があります。

- B からC へのマピングを無視して、A からB にお生成された部分をプレビューする。
- A からB へのマピングを無視して、B からCにより生成された部分をプレビューする。

上に示されるように ダスルーボタンを無効化すると、ReportB おっはReportC を選択することができます(両方に) 👁 ボタンが表示されます)。

パマスルーポタンの無効化により、中間コンポーネントかどの入力ファイルを読み取るかを選択することができます。多くの場合、これは出力 XML ファイルフィールド(このサンプル内で示されているように)内で定義されたファイルと同じです。

コンポーネント名: ReportB	
- スキーマ ファイル(F)	
ExpenseReport.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
ReportB.xml	参照(B) 編集(E)
─出力 XML ファイル(O)	
ReportB.xml	参照(R) 編集(D)

中間コンポーネト上に同じ入力と出力ファイル約存在することは、マピングからコードを生成する場合、おけはマピングをMapForce Server 上で実行する場合、特に重要です。前記のようにこの環境は、マピングチェーン内の各コンポーネトにより作成される出力により生成されます。ですから、中間コンポーネントが処理するファイル(この場合はReportB.xml)を受け取り、異なるファイルを検索するよりも、同じファイルを次のマピングに送ることは意味かあります。(パマスルーボタンか無効化されており中間コンポーネント上で同じ入力と出力ファイル名が存在してよと、生成されたコード内で、おけよMapForce Server 実行中にゴシステムが指定されたファイルを見つけることができません」などのエラーが発生する場合かあります。

3番目のエレポーネント(ReportC)のプレビューオタン Control MapForce 内でマンピングをプレビューしようと試みると、エラーが 発生します。上記の設定に従い、ReportB.xml とう名前のファイルが入力として期待されるナム、これは予想されている振る舞いです。 しかしなから、マンピングは、(パマスルーボタンが有効化されておらず、BからCへのマンピングの部分のみが実行されているナム) おごの 名前のファイルは生成されていません。この問題は簡単に解決することができます:

- 1. 中間コンポーネトのプレビューボタをクリックします。
- 2. マピングをプレビューするために、出力 ダブをクトックします。
- 3. マピング(<マイドキュメント>¥AltovaWapForce2021WapForceExamples¥Tutorial》と同じフォルダー内に結果 出力ファイルを **ReportB.xml** とこで保存します。

3番目のエレポーネント(ReportC)のプレビューボタンをクリックすると、エラーは表示されなくなります。

ノウスルーボタンが無効化されている場合、関連したStyleVision Power StyleSheet(SPS)ファイルを持つ各コンポーネントのナメの StyleVisionーにお生成された出力をプレビューすることができます。特に、(最後のレポートに加え)中間レポートのHTML バージョンを確認することもできます:

中間コンポーネントの設定



Personal Expense Report				Curren	cy: 🔘	Dollars O Euro	os 🔾 Yen	Currency \$	
r ersonar Expense Report			🗸 Det	ailed rep	port				
Employ	oo Infa	ormatio	n						
Linploy	ee mit	madu	Л						
	Fred		Landis			Projec	t Manager		
	First Nan	1e	Last Name		1	Title			
	flandi	allenana	nu11.com			102 /	56 79		
	1.181101	s@nano	nun.com			123-4.	50-78		
	E-Mail				F	Phone			
Expense	Expense List								
Тур	e	Expe	ense To	Date (yyyy-mu dd)	m-	Expe	nses \$	Desc	ription
тур Travel	•	Expe	oment 🗸	Date (yyyy-mu dd) 2003-01- 02	m- Tr 33	Expe avel 7.88	nses \$ Lodging	Desc Biz jet	ription
Typ Travel Travel		Expe Develop Account	ense To oment 🗸 ting 🗸	Date (yyyy-mu dd) 2003-01- 02 2003-07- 07	m- 33 Tr 10	Expe ravel 7.88 ravel 14.22	Lodging Lodging	Desc Biz jet Ambassa	ription dor class
Typ Travel Travel Travel		Expe Develop Account	ense To oment 🗸 ting 🔽	Date (1777)-ma 2003-01- 02 2003-07- 07 2003-02- 02	n- 33 Tr 10 Tr 20	Expe ravel 7.88 ravel 14.22 ravel 00	Lodging Lodging Lodging Lodging	Desc Biz jet Ambassa Hong Ko	ription dor class ng

中間コンポーネントのHTML 出力

4.4 複数の入力または出力ファイルを動的に処理

マンプが実行される際、MapForceに複数のファイルを処理するように構成することができます(例: ディレクトリ内の全てのファイル)。この機能を使用して、以下のようなタスクを解決することができます:

- マピングに処理する入力ファイルのリストを与えます。
- 単一の出力ファイルの代わりにファイルのノストのマンピング出力を生成します。
- 入力と出力ファイル名がランタイムで定義されている箇所のマッピングアプリケーションを生成します。
- ファイルのセナを他のフォーマナに変換します。
- 大きなファイル(おさまデータベース)を小さよい ~ ソン分割します。
- 複数のファイルを1つの大きなファイルニマージする(おけよデータベースコードする)

MapForce コノポーネントレ複数のファイルを処理するようこ以下の方法で構成することができます:

- コンポーネント 設定で必要とされる入力お台は出力ファイルに固定されたファイル名の代わりにフイルドカード文字を使用していたを 与えまず(ユンポーネント設定を変更するを参照)。すなわち、ワールドカード*と?をコンポーネント設定ダイアログボックスに入 カし、MapForce は、マンピング実行される際に対応するいたを解決します。
- パンを動的に提供する、コンポーネントシーケンスルートノードに接続します。(例 replace-fileext 関数の結果)。マンピン グが実行されると、MapForce は、全ての入力ファイルを動的に読み込み、おっ全ての出力ファイルを動的に生成します。

目的にお異なりますが、同じマンングで「つまけは両方のアプローチを取ることができます。しかしながら、同じエンポーネント上で同時に両方のアプローチを使用することは、あまり意味をなしません。特定のエンポーネントのために、MapForce かどのアプローチを使用するかを命令するには、エンポーネントのルートノードの横の「ファイル」(「File」)まけは「ファイル/文字列」(「File/String」)がなをクルックします。この代をノーより以下の振る舞しを指定することができます。

コンポーネント設定からファイル名を使用する	コンポーネントがっつまけは複数のインスタンスファイルを処理する場合、この オプションはMapForce にコンポーネント設定ダイアログボックスで定義さ れたファイルを処理するように命令します。
	このオプランを選択すると、ルートノードには入力コネクタか与えられません。
	④ Articles □ つァイル: (デフォルト) <u>ファイル/文字列</u> □ 〇 Articles □ 〇 Article
	コンポーネント設定ダイアログボックス内でおう入力おけは出力ファイルを指 定していない場合、ルートノードの名前はファイル(デフォルト)です。そ れ以外の場合、ルートノードは、出力ファイル名、セミコロン(;)が後につい た入力ファイル名を表示します。
	入力ファイル名が出力ファイルと同じ場合、ルートノードの名前として表示 されます。

	<pre> Articles 日□ファイル: Articles.xml ファイル/文字列 口() Articles ① Article ① Article ② ① Article ② ① ① ① ①</pre>
	このオプシンお」は「マルングから与えられた動的なファイル名を使用 する」オプシンを選択することができます。
マッピングから与えられた動的なファイル名を使用する	このオプロンは、コンポーネントのルートノードの値は接続することにより、 マンピングコレアで定義するファイル名を処理するようにMapForce に命 令します。
	このオプランを選択すると、ルートノードがマンピングの実行中動的に処理されるファイル名を与える入力コネクタの値は接続します。コンポーネント設定ダイアログボックス内に定義されナラアイル名を持つ場合、これらの値は無視されます。
	このオプションが選択された場合、ルートノードの名前は以下として表示されますファイル く動的〉。
	 G Articles ○□ファイル: <動的> ファイル/文字列 ○□() Articles ○□() Article
	コンポーネント設定からファイル名を使用するオプションと共にこのオプ ションは相互排他的です。
XML 二文字列を解析、JSON 二文字列を解析、 CSV 二文字列を解析、FLF 二文字列を解析、 EDI 二文字列を解析	このオプションがオノフなると、コンポーネントが文字列の値をルートノードへの入力として受け入れ、それぞれ、XML、JSON、CSV、FLF、まけは EDI構造に変換します。詳細に関しては、文字列の解析とノリアル化を 参照してくたさい。
XML を文字列にシリアル化する、JSON を文字列に シリアル化する、CSV を文字列にシリアル化する、 FLF を文字列にシリアル化する、EDI を文字列に シリアル化する	このオプションがオノフなると、コンポーネントは構造を入力として受け入れ、 文字列に変換します。入力構造は、それぞれXML、JSON、CSV、 固定長フィーリド、おけはEDIであることができます。詳細に関しては、文 字列の解析とシリアリルとを参照してくたさい。

複数の入力ませ出力ファイルを以下のエレポーネトのために定義することができます。

- XML ファイル
- テキストファイル(CSV*, FLF*ファイルとFlexText**ファイル)
- EDIドキュメント**
- Excel スルボシート**
- XBRLドキュメント**
- JSON ファイル**

プロトコルバッファーファイル**

*MapForce Professional Edition 必須 ** MapForce Enterprise Edition 必須

以下のテーブルは、MapForce 言語内での動的な入力と出力ファイルおよびアイルドカードへのサポートについて説明されています。

ターゲット言語	動的入力ファイル名	入力ファイル名のためにサポートされてい るワイルドカード	動的な出力ファイル名
XSLT 1.0	*	XSLT 1.0 におりサポートされていません	XSLT 1.0 によりサポートされてい ません
XSLT 2.0	*	*(1)	*
XSLT 3.0		*(1)	*
XQuery	*	*(1)	XQuery におりサポートされていません
C++	*	*	*
C#	*	*	*
Java	*	*	*
BUILT-IN	*	*	*

レジェンド:

*	サポートされている
(1)	XSLT 2.0、XSLT 3.0 とXQuery は fn: collection 関数を使用します。Altova XSLT 2.0、XSLT 3.0 と XQuery エンジン内の実装はフイバドカードを解決します。他のエンジンは異なる振る舞しをする可能性があります。

4.4.1 複数の入力ファイルを単一の出力ファイルにマップする

複数の入力ファイルを処理するコよ、以下を行います:

コンポーネト設定ダイアログボックスに入力ファイルと、Cワイルドカード(*おけま?)を持つのファイルレジを入力します。一致する全てのファイルが理されます。下のサンプルは、入力 XML ファイルフィールド内でマメピング入力として名前が「Nanonul-「で開始する全てのファイルを提供する*ワイルドカード文字を使用します。ターゲトコンポーネトに動的コネクタが存在せず、ソースコンポーネトはワイルドカード*を使用して複数のファイルにアクセスしているオーダ、複数の入力ファイルは単一の出力ファイルとてマージされます。ターゲトコンポーネト内のルートノードの名前はFile: <default>です。これはコンポーネト設定ダイアログボックス内で出力ファイルレジカ定義されていないことを示します。複数のノースファイルは、ですから、ターゲナトドキュメント内で追加されます。

Altova_Hierarchical Image: Constraint of the second sec	Altova_Hierarchical つファイル: (ブフォルト) ファイル 〇 Altova 〇 PrimaryKey 〇 Office 〇 PrimaryKey 〇 ProreignKey 〇 Pax 〇 Phone 〇 Phone 〇 Department
---	--

MergeMultipleFiles.mfd (MapForce Basic Edition)

ソースコンポーネントのファイルレードに文字列のシーケンスをマップします。シーケンス内の各文字列が、つのファイル名を表します。
 文字列は自動的に解決されるワイルドカードを含むこともできます。XML ファイル、データベーステキストフィールドなどのコンポーネントがファイル名のシーケンスを提供します。



MergeMultipleFiles_List.mfd (MapForce Basic Edition)

4.4.2 複数の入力ファイルを複数の出力ファイルにマッピングする

複数のファイルを複数のターゲトファイルにマップするコよ、一意の出力ファイル名を生成する必要かあります。出力ファイル名が入力データ 内の文字列から派生する場合かあり、他の場合、入力ファイル名から出力ファイル名を派生させることが役にナニつ場合かあります。例: ファイ ル拡張子の変更により。

以下のマピングでは、concat 関数を使用してプレフィックス "Persons-" を追加し、出力ファイル名を入力ファイル名から派生させます。



MultipleInputToMultipleOutputFiles.mfd (MapForce Basic Edition)

メモ 処理関数を使用することなく、入力と出力ルートノードを直接接続することを回避します。これを行うことによりマピングが実行される際、入力ファイルが上書きされます。上記のconcat 関数などを使用して、出力ファイル名を変更することができます。

メニューオプタン「ファイル | マッピング設定」によりマッピングで使用されるファイル マ設定をグローノ ジレニ定義することができます (マッピ ング設定の変更を参照)。

4.4.3 ファイル名をマッピングパラメーターとして提供する

マピングにカスタムファイル名を入力パラメーターとして提供すること、以下を行います

- 1. 単純型入力コンポーネトをマンプに追加する(「関数」メニューから「入力の挿入」をクトックします)。コンポーネトについての更に詳しい情報は、マンピングロン・ラメーターを提供するを参照してくたさい。
- 2. ソースエンポーネトの「ファイル」(「iel) おけは「ファイル/文字列」(「iel/String」) ボタンをクトックし、「マッピングから与 えられた動的なファイル名を使用する」を選択します
- 3. マピングソースとして振舞う、コンポーネントのルートノードに単純型入力パラメーターを接続します。

成功例に関しては例名前をマピングリティーターとして使用するを参照してくたとい。

4.4.4 複数の出力ファイルをプレビューする

プレビューウィンドウニマッピングの結果を表示するなかに、「出力」 ダブをクトックします。マッピングが複数の出力ファイルを作成する場合、各ファイルは、「出力」 ダブ内でペインごとご番号が付けられます。 矢印をクリックして、それぞれの出力ファイルを確認してくたさい。

	•	Preview 1 of 2 C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\Persons-Nanonull-Branch.xml 🔹
1	E	*?xml version="1.0" encoding="UTF-8"?> Generated by Altova MapForce (http://www.altova.com/mapforce) using<br C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\MultipleInputToMultipleOutputFiles.mfd> <personlist xmlns:xsi<br="">="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="PersonList.xsd"><!--6 PersonList xmlns:xsi<br-->input file C:\Users\altova\Documents\Altova\MapForce2015\MapForceExamples\MultipleOutputFiles.mfd><personlist p="" xmlns:xsi<=""></personlist></personlist>
2 3 4 5		<pre>"> <first>Steve</first> <last>Meier</last> </pre>
6 7 8 9	e	<pre><first>Theo</first> <last>Bone</last> </pre>
10 11 12 13	e	<pre><first>Max</first> <_Last>Nafta </pre>
14 15 16 17 18	¢	<pre></pre> <person> <person role="Support Engineer"> </person></person>
19 20 21 22	¢	<pre><last>Franken</last> Mark</pre>
23 24 25		<last>Redgreen</last>

MultipleInputToMultipleOutputFiles.mfd

生成された出力ファイルを保存するためコお以下を行います

- 「出力」メニューから「すべての出力ファイルを保存」(💷)をクリックします。
- 「すべての生成された出力を保存」(🗊)ツールシーボタンをクリックします。

4.4.5 例: 1つの XML ファイルを複数のファイルに分割

このサンプルは単一のノースXML ファイルから動的に複数のXML ファイルを作成する方法を説明します。このサンプルニ使用されるマピン グは以下で検索することができます: <マイドキュメント > ¥Altova Wap Force2021 Wap ForceExamples ¥Tutorial ¥Tut-ExpReport-dyn.mfd。

ソースXML ファイル(マピングと同じフォルダーで見つけることのできる)は、「Fred Landis」とう個人の経費レポートから構成されており、5つの異なる種類の経費アイテムを含んでいます。このサンプルの目的は個々の経費アイテムのためにXML ファイルを生成することです。

A Person							
		First	Fred				
		() Last	Landis	Landis			
		() Title	Project Manag	er			
		Phone	123-456-78				
		() Email	f.landis@nand	onull.com			
expens	e-ite	m (5)					
		= type	= expto () Date () Travel (() Lodging	
	1	Travel	Development	2003-01-02	🖬 Travel Trav-cost=337.88		
	2	Lodging	Sales	2003-01-01		🔳 Lodging	
	3	Travel	Accounting	2003-07-07	Travel Trav-cost=1014.22		
	4	Travel	Marketing	2003-02-02	🖬 Travel Trav-cost=2000		
	5	Meal	Sales	2003-03-03			

mf-ExpReport xml (XMLSpy グルドビューインで表示)

type 属性は特定の経費アイテムの型を定義し、これはノースファイルを分割するために使用されるアイテムの型です。このエクササイズのゴ ールを達成するコよ、以下を行います

- 1. (「ライブラリ」ペインの「core | string 関数」 ライブラ かぶ ラッグして concat 関数を挿入します。
- 2. 定数を入力するコよ(「挿入」メニューから「定数」をクトックして) ".xml" を値とて入力します。
 3. (「ライブラリ」 ペインの「core | generator 関数」からドラックして) auto-number 関数を入力します。
- 4. 「ファイル」(「IIII) おけなーゲトコンポーネトの「ファイル文字列」(File/String)) ボタンをクルクし、「マッピングから 与えられた動的なファイル名を使用する」を選択します
- 5. 下に表示されるとお、接続を作成し、「出力」タブをクリックしマピングの結果を確認します。



Tut-ExpReport-dyn.mfd (MapForce Basic Edition)

結果的出力ファイルは、以下のとおり動的に名前が付けられます

- type 属性は ファイル名の最初の部分を提供します。(例: "Travel")。
- 自動連番 関数は、ファイルの連番を提供します。(例 "Travel1"、"Travel2"、など)。
- 定数は ".xml" であるファイル拡張子を提供します。このため "Travel1.xml" は最初のファイル名です。

4.4.6 例: データベーステーブルに対して複数の XML を生成

この例は、データベーステーブルのそれそれのレコードに対して、動的に複数のXML ファイルを生成する方法を説明します。この例は、以下で参照することができます: <マイドキュメント > ¥ltova Wap Force 2021 Wap Force Examples ¥utorial ¥PersonDBdyn.mfd。

ソースデータベースファイルコは5 つのテーブルがあり、Person テーブルコは21 名の人物に関するレコードが収められています。Person テ ーブル内のそれぞれのレコードに対し、個別のXML ファイルを生成することがこのサンプルの目的です。

Person				
PrimaryKey 👻	ForeignKey 👻	EMail 👻	First 🔹	Last 🔹
1	1	v.callaby@nanonu	Vernon	Callaby
2	1	f.further@nanonu	Frank	Further
3	ի	l.matise@nanonu	Loby	Matise
4	2	j.firstbread@nanc	Joe	Firstbread
5	2	s.sanna@nanonul	Susi	Sanna
6	3	f.landis@nanonul	Fred	Landis
7	3	m.landis@nanonu	Michelle	Butler
8	3	t.little@nanonull.	Ted	Little

"PrimaryKey" フィールドの内容にてテーブル内の人物を(データベースでは)識別しているため、このアイテムを使用してノースデータベースにあるデータを複数のファイルへ分割します。このエクササイズのゴールを達成すること、以下を行います:

- 1. concat 関数を挿入します。(「ライブラリ」 ペインの「コア | 文字列関数」 ライブラ から関数をドラッグすることができます)。
- 2. 定数を入力するコよ(「挿入」メニューカジ 定数」をクリックし) ".xml"を値とて入力します。
- 3. 「ファイル」(「IEI) おさなーゲナコンポーネトの「ファイル/文字列」(<u>File/String</u>) ボタンをクリックし、「マッピングから与 えられた動的なファイル名を使用する」を選択します
- 4. 下で表示されるようは接続を作成します。「出力」タブクルクしてマピングの結果を確認します。



PersonDB-dyn.mfd (MapForce Professional Edition)

結果的出力ファイルは、以下のとおり動的に名前が付けられます

- 「PrimaryKey」フィールドは、名前の最初の部分を提供します(例 "1")。
- 定数は ファイル拡張子 (".xml")を提供します。このため 1.xml は最初のファイルのファイルのファイル名です。

4.5 マッピングにパラメーターを与える

 ハラメーターを入力とてとるマメングを作成する場合、「単純な入力コンポーネト」と呼ばれる特別なコンポーネト型を追加して行うことが できます。単純型入力コンポーネトは常に(例えば、文字列、整数などの単純型データ型をアイテムとノーケンスの構造の代わりに持って います。例えば、下に表示されるマメングでは単純型入力コンポーネント count が存在します。役割は、ラメーターとして(値 10 をデ フォルトとして持つソース XML ファイルから抽出される行の最大数を提供します。重要な点は、N が、ラメーターの値である個所でマメン グが最高 N 気温のみを出力するように<u>first-items</u> 関数に入力として提供されるノードは並べ替えコンポーネントを使用して並べ替えられ ます。



FindHighestTemperatures.mfd

他の単純型入力コンポーネントの比較的一般的な使用方法はマンピングへのファイル名の提供です。これは入力ファイルを読み取る、おさよ 出力ファイルを動的に読み取るマンピング内で役に立ちます。複数の入力おささ出力ファイルを動的に処理するを参照してくたさい。

単純型入力 コンポーネントを次のMapForce 変換言語内で使用することができます:

- BUILT-IN (「プレビュー」 タブから Map Force 内で直接マピングの変換をプレビューする場合)
- BUILTHN (コンパルにわた MapForce Server 実行ファイルを実行する場合)
- XSLT 1.0、XSLT 2.0、XSLT 3.0、
- XQuery
- C++
- C#
- Java

MapForce Server を使用してマピングを実行、おけま生成されたコードを使用して実行する場合、単純型入力コンポーネトは、コマン ドライン・ゲラメーター」ないます。マピングがXSLT 変換として生成される場合、単純型入力コンポーネトは、生成されたXSLT ファイル 内のスタイルシート・ゲーター」一対応します。

単純型入力コンポーネト(おけまでメーター)を任意おけお必須で作成することができます<u>単純型入力コンポーネトを追加</u>を参照してく たさい。必要であれば、マメピング入力パラメーターのためにデフォルトの値を作成することができます。<u>デフォルトの入力値を作成する</u>を参照して くたさい。マメピングの実行時につ、ラメーターの値を明示的に提供していない場合でもこれにより、マメピングを安全に実行することができます。例 <u>: ファイル名をマメピングパラメーターとして使用する</u>を参照。

メインマッピングエリアに追加される入力、ラメーターをユーザー定義関数内の入力、ラメーターと区別してくたさい。これら2つの類似点と相違 点は、以下のとおりです。

マッピング上の入力パラメーター	ユーザー定義関数の入力パラメーター
「関数 入力の挿入」メニューから追加する。	「関数 入力の挿入」メニューから追加する。
単純型データ型を持つことができます(文字列、整数など)。	単純型と複雑型のデータ型を持ってとかできます
マピング全体に適用することができます。	定義された方法で関数のコンテキストに適用することができます。

逆のマピングを作成する場合、(メニューコマド「ツール | 逆順マピングの作成」を使用すると単純型入力コンポーネトは、単純型 出力コンポーネトになります。

4.5.1 単純型入力コンポーネントの追加

マッピングに単純型入力コンポーネントを追加する

- 1. マピングウイドウに(ユーザー定義関数ではない)メインマピングが表示されていることを確認してくたさい。
- 2. 以下のつを行ってくたさい
 - •「関数」メニューから入力」をクルクします。
 - •「挿入」メニューカジ入力の挿入」をクルクします。
 - •「入力の挿入」^{19]} ツールドーボタンをクリックします。

🙁 入力(D作成 ×
名前(N):	input
データ型 (ロ	D): string ~
	☑入力は必須(1)
ーデザイン	時実行
□値を	指定(S)
值 <mark>(V)</mark> :	
	OK キャンセル

- 3. 名前を入力して、この入力に必要なデータ型を選択します。入力が必須のマンピング・ラメーターとして扱われる必要がある場合、 「入力は必須」チェックボックスを選択します。設定の完全なリストは、<u>単純型入力コンポーネント設定</u>を参照してくたさい。
- メモ ・、ラメーター名は文字、数値、およびアンダースコアのみを含むことができます。他の文字は許可されていません。これによりマッピング がコード生成言語すべてのために使用できます。
 - 4. 「OK」をクリックします。

ここで定義されている設定は後に変更することができます、単純型入力コンポーネイト設定を参照)。

- 1. マピングウイドウに(ユーザー定義関数ではない)メインマピングが表示されていることを確認してください。
- 2. 「関数」メニューカジ入力」をクリックします。
- 3. 名前を入力して、この入力に必要なデータ型を選択します。入力が必須のマッピングパラメーターとして扱われる必要がある場合、 「入力は必須」チェックボックスを選択します。設定の完全なリストは、<u>単純型入力コンポーネト設定</u>を参照してくたさい。
- メモ ハラメーターの名前は、文字、数字、アンダースコアを含むことができますが、その他は許可されていません。これにより、マピングは 全てのコード生成言語と作業可能にないます。

4. 「OK」をクリックします。
🕑 入力の作成 🛛 🗾 🔀
名前(N): input
デー久型(D): string ▼
▼ 入力は必須(I)
デザイン時実行
🔲 値を指定(S)
値(V):
<u>ок</u> ++>>セル

入力の作成 ダイアログボックス

ここで定義されている設定は後に変更することができます、単純型入力コンポーネト設定を参照)。

4.5.2 単純型入力コンポーネント設定

単純型入力コンポーネントに適用することのできる設定をマンピングエリアに追加する際に定義することができます。後に設定を「入力の編集」ダイアログボックスから変更することも可能です。

「入力の編集」ダイアログボックスを開くには、以下を行います

- コンポーネントを選択して、「コンポーネント」メニューからプロノティ」をクリックします。
- エンポーネントをダブルクトックします。
- コパーネトを右クリックして、「プロノティ」をクリックします。

名前(N): InputFileName
データ型(D): string 〜
□入力は必須(1)
デザイン時実行
值(V):
OK キャンセル

入力の編集 ダイアログボックス

.使用することのできる設定は、以下のとおりです:

名前	このエレポーやトレー対応する入力/ ラメーターのナメの詳細名を入力します。マピングの実行時、このテキストボックスに入力された値がマピングに与えられる/ ラメーターの名前しております。 ですから、 スペースまたは特別な文字に許可されていません。			
<i>データ型</i>	デフォルトでは、全ての入力パラメーターは文字列データ型として扱われます。パラメーターが異なる型を持つ場合、リストからそれぞれの値を選択します。マンピングが実行された場合、MapForceは、入力パラメーターをここで選択されたデータ型にキャストします。			
入力は必須	有効化されている場合、この設定は、入力、ラメーターを必須しします(すなわち、、 ラメーターの値が提供 されない限り、マンピングを実行することしてきません)。			
	入力パラメーターのためのデフォルトの値を指定するには、このチェックボックスをクリアします(デフォルトの入力 値を作成するを参照)。			
値の指定	この設定は、デザイン時にマンピングを実行する場合のみ適用することができます。「プレビュー」タブをクトック することにより、コンポーネント内に直接マンピング入力として使用される値を入力することができます。			
値	この設定は、デザイン時にマンピングを実行する場合のみ適用することができます。「プレビュー」タブをクトック することにより、マンピング入力として MapForce で使用される値を入力するこよ、値の指定チェックボックス を選択し、必要な値を入力します。			
	メモ 値の指定 チェックボックスをクリックし横のボックスコ値を入力した場合、入力された値は、マメング をプレニーする際に、以前入力された(すなオタ、デザイン時の実行時の、デフォリトの値を上書き します。しかしなから、デザイン時の値はMapForce Server による実行おけまFlowForce Server へのデプロイ時に生成された、XSLT、XQuery、おけまプログラムコードコードに影響を与 えません。			

4.5.3 デフォルトの入力値を作成する

入力コンポーネントをマピングエノアに追加すると、default アイテムがコンポーネントの左側に表示されます。



単純型入力コンポーネント

デフォルトアイテムは任意のデフォルトの値のこの入力コンポーネントへの接続を以下のように有効化します。

1. 定数コンポーネントを追加し、(「挿入」メニューカジ定数」をクルクします。)入力コンポーネントのdefault アイテムコ接続します。

💶 🖱 Altova_Hierarchical"
🐮 InputFileName
😫 Altova_Hierarchical
👍 🗗 ファイル: <dynamic> 🚦</dynamic>
p 🖓 Altova 🖒

2. 入力コンポーネトをダブルクトックして、入力は必須チェックボックスのチェックをクリアします。デフォルトの入力の値を作成すると、この設定は重要しなり、マッピングの検証に関する警告を引き起こす可能性があります。

😢 入力の作成	×
名前(N): InputFileName	
データ型(D): string 🔹	2
□入力は必須(I)	
デザイン時実行	
(直(V):	
OK キャンセル	

- 3. 「OK」をクリックします。
- メモ 値の指定 チェックボックスをクトックし横のボックスミ値を入力した場合、入力された値は、マッピングをプレビューする際に、以前入 力された(すなわち、デザイン時の実行時のデフォットの値を上書きします。しかしなから、デザイン時の値はMapForce Server による実行ませるように、XQuery、ませるプログラムコードコードに 影響を与えません。

4.5.4 例: ファイル名をマッピングパラメーターとして使用する

このサンプルはランタイムに入力・ウメーターを取り込むマンピングの実行に必要なステップを説明します。マンピングデザインファイルは、以下のパスにあるサンプルを使用します: くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\FileNamesAsParameters.mfd.

このマピングはXML ファイルからデータを抽出し、ターゲオ XML ファイルに書き込みます。データはターゲオファイルコヨま変更されることなく書き込まれます。属性 PrimaryKey とName のみがマピングからの定数の一部と共に作成されます。マピングのメインの目的はマピングのランタイムで呼び出し元がマピングの、ラメーターとして入力ファイルの名前と出力ファイルの名前を指定できるようなることです。

これを達成するコよ、マピングコは2つの入力コンポーネントが存在します:InputFileName and OutputFileName。これらのコンポーネントはノースとターゲット XML ファイルの入力ファイル名(および出力ファイル名をそれぞれ)、与えます。この理由のため、これらのコンポーネントは、ファイル く動的>アイテムコ接続されています。ファイル/文字列(「File/String」)、ポタンをクトックして、マッピングにより提供された動的なファイル名を使用するを選択してコンポーネントをこのモードに切り替えることができます。



FileNamesAsParameters.mfd (MapForce Enterprise Edition)

InputFileName とOutputFileName コンポーネトのタイトルレーをダブルクトックすると、コンポーネントをビューすることができます。例 えば、<u>単純型入力コンポーネント設定</u>で説明されているとおり、入力パラメーターのデータ型を指定、おけよ、入力パラメーターの名前を変更 することができます。このサンプルでは、入力と出力パラメーターは以下のように構成されています。

- InputFileName 、 ラメーターは型 [string] で、 同じマピング内で定義されている定数により与えられているデフォルトの値を 持っています。 定数は型 [string] で、 その値は [Altova_ Hierarchical.xml] です。 このために、 他の値が、 ラメーターとして与え られないと想定し、 マピングが実行されると [Altova_ Hierarchical.xml] と呼ばれるファイルからデータを読み取らうと試みます。
- OutputFileName / デメーターは型 [string] で、同じマピング内で定義されている定数により与えられているデフォルトの値を 持っています。定数は型 [string] で値は [Altova_ Hierarchical_ output xml] ですこのため、このために、他の値が、デメータ ーとして与えられないと想定し、マピングは実行時に [Altova_ Hierarchical_ output xml] と呼ばれる XML 出力ファイルを 作成します。

以下のセグタンオマピングの実行方法と以下の変換言語でパラメーターを与える方法を説明しています

- RaptorXML Server を使用した<u>XSLT 2.0</u>
- MapForce Server を使用したビルドイン(MapForce Server 実行ファイル)
- <u>Java</u>
- <u>C#</u>
- <u>C++</u>

XSLT 2.0

XSLT 1.0、XSLT 2.0、おはXSLT 3.0 でコードを生成する場合、XSLTファイルに加え、DoTransform.bat バッチファイルが選択されたターゲナディレクトリで生成されます。DoTransform.bat によりRaptorXML Server を使用してマメングを実行できるようにないます。
RaptorXML Server を使用した自動化を参照してくたさい。

異なる入力(おけよ出力)ファイルを使用するためにDoTransform.bat ファイルを必要とする ラメーター かきまれるように以下のように 編集します:

- 1. 最初にXSLT コードを生成します。例えば、XSLT 2.0, を生成する場合「ファイル| コード 生成 | XSLT 2.0」を選択しま す。
- 2. <マイドキュメント>¥iltovaWapForce2021WapForceExamples¥からAltova_Hierarchical.xml ファイルを XSLT 2.0 コードが直接生成されるディレクトリにコピーします(このサンプルでは、c:\codegen\examples\xslt2\)。 InputFileName パラメーター(コスタム値を提供しない場合、前記の通り、マメピング)はこのファイルを読み取らうと試みます。
- 3. DoTransform.bat かりき* の前おけば後にカスタム入カノ・ラメーターを含むように編集します。ノラメーターの値は一重引用符 で閉じられていることに注意してください。使用することのできる入カノ・ラメーターはのrem(リマーク) セクション・コストされていま す。output.xml と呼ばれる出力ファイルを生成すると想定します。これを達成するけは DoTransform.bat ファイルを以下 のように変更します。

```
@echo off
RaptorXML xslt --xslt-version=2
    --input="MappingMapToAltova_Hierarchical.xslt"
    --param=OutputFileName:'output.xml' %* "MappingMapToAltova_Hierarchical.xslt"
rem --param=InputFileName:
rem --param=OutputFileName:
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

DoTransform.batファイルを実行すると、RaptorXML Server はAltova_Hierarchical.xmlを入力パラメーターとして使用して、変換を完了します。上のステップに従うと生成されたファイルの名前はoutput.xml になります。



MapForce Server 実行ファイル

カスタム入力パラメーターをMapForce Server 実行ファイルコ提供する方法

- 1. これをまだっていない場合 FileNamesAsParameters.mfd をくマイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレカトリから開きます。
- 2. [ファイル] メニューで [Map Force Server 実行ファイルにコンパイルする] をクリックします。 Map Force Server 実行 ファイルにマッピングをエッパルする も参照してくたさい。 プロンプトされると、mfx 実行ファイルをにの場合は c: \codegen\examples\mfx)コンピューターのディレクトリに実行ファイルを保存します。
- 3. <マイドキュメント> Altova Map Force 2021 Map Force Examples ¥ デルノトリから Altova_Hierarchical.xml ファイルを.mfx ファイルを保存した場所にコピーします。
- 4. MapForce Server を以下のコマドで実行します:

MapForceServer.exe run "C:\codegen\examples\mfx\FileNamesAsParameters.mfx"
 -p=InputFileName:"C:\codegen\examples\mfx\Altova_Hierarchical.xml"
 -p=OutputFileName:"C:\codegen\examples\mfx\OutputFile.xml"

上のMapForce Server コマドでは、, -p=InputFileName と-p=OutputFileName はマピングに対する入力パラメーターです。-OutputFileName の値とて全てのファイル名を使用することができます。しかしながら、the file name supplied in -InputFileName パラメーター内で提供されるファイル名は物理的なファイルとして存在する必要があります。それ以外の場合マピングは 失敗します。

メモ [MapForceServer.exe かわ部おけおト部コマイ、操作可能なプログラム、おけお シチファイルとて識別されていません」というエラーメッセージが表示されると、現在のデルノオールを MapForce Server 実行可能ファイルがインストールされている場所に変更してくたさい。マメピングを実行する度につ なを変更することを回避するために、オペレーティングシステムのPATH 環境変数に MapForce Server 実行可能ファイルがインストールされているディレクオールへの なを追加します (例えば、C:\Program Files (x86)\Altova\MapForceServer2021\bin)。

W ith MapForce Server を使用すると、マピングの実行はC++、C#、おゴはJava などの言語を呼び出す) MapForce Server API を呼び出しても行うことができます。このシナリオ ご関する詳細は MapForce Server ドキュメンテーション (<u>https://www.altova.com/ja/documentationr</u>)を参照してくたさい。

Java

カスタム入力、デメーターをJava .jar アプリケーションは提供する場合:

- 1. これを既に行っていない場合 FileNamesAsParameters.mfd サンプルをくマイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレケーから開きます。
- 2. [ファイル] メニューカショードの生成 | Java]をクトックします。
- 3. Java コードを実行可能 JAR ファイルニエン イルします。Eclipse を使用してこれを行う方法の例は、例 Java コードの生成 と実行を参照してくたさい。
- 4. Altova_Hierarchical.xml ファイルをくマイドキュメント> ¥Altova ¥MapForce2021 ¥MapForceExamples ¥から .jar ファイルが存在するデルノトリニピーします。InputFileName パラメーターにカスタム値を提供したい場合、前記の通り、 マピングはこのファイルを読み取ろうと試みます。
- 5. アプリケーションを以下のコマンドで実行する

java -jar Mapping.jar /OutputFileName "output.xml"

上のコマドでは、入力パラメーター/OutputFileName は生成される出力ファイルの名前を提供します。

メモ jar ファイルコ デメーターをパマする際にフイルドカードを使用する場合、ワイルドカードパテメーターを引用符で囲んでくたさい。例

java -jar Mapping.jar /InputFileName "altova-*.xml"

C#

カスタム入カパラメーターをMapForce により生成されたC# コマバラインアプリケーションに提供する方法:

- 1. これをまだっていない場合 FileNamesAsParameters.mfd をくマイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレクトリから開きます。
- 2. ファイルファイルメニューから[コードの生成 | C#] を選択しターゲナディンケーを選択します(このサンプルの場合、C: \codegen\examples\cs)。
- 3. ソレーションを Visual Studio で開きビルドします (Ctrl + Shift + B)。
- Altova_Hierarchical.xml ファイルをくマイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥から Mapping.exe が生成されたディレクトリニピーします(このサンプルではC: \codegen\examples\cs\Mapping\bin\Debug)。InputFileName / ラメーター(こカスタム)値を提供したは場合、前 記の通り、マメピング)はこのファイルを読み取らうと試みます。
- 5. コマドプロンプトウイドウを開き Mapping.exe かある場所にディノクトリを変更します。

cd C:\codegen\examples\cs\Mapping\bin\Debug

6. アプケーションを以下のコマドで実行する

Mapping.exe /OutputFileName output.xml

上のコマボでは、入力パラメーター/OutputFileName は生成される出力ファイルの名前を提供します。

C++

カスタム入力パラメーターをMapForceにより生成されたC++コマイドラインアプリケーションは提供する方法:

- 1. これをお行っていない場合 FileNamesAsParameters.mfd をくマイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレカトルら開きます。
- 2. ファイルファイルメニューから[コードの生成 | C++] を選択しターゲホディレケトリを選択します(このサンプルの場合、C: \codegen\examples\cpp)。
- 3. ソリューションを Visual Studio て開きビルドします (Ctrl + Shift + B)。
- Altova_Hierarchical.xml ファイルをくマイドキュメント> ¥Altova WapForce2021 WapForceExamples ¥から Mapping.exe が生成されたディレクトリニピーします(このサンプルではC: \codegen\examples\cpp\Mapping\Debug)。InputFileName / ラメーター(こカスタム値を提供しては、場合、前記の 通り、マノビングはこのファイルを読み取るうと試みます。
- 5. コマボプロンプ・ウイボウを開き Mapping.exe かある場所にディレクトリを変更します。

cd C:\codegen\examples\cpp\Mapping\Debug

6. アプケーションを以下のコマイで実行する

Mapping.exe /OutputFileName output.xml

上のコマドでは、入力パラメーター/OutputFileName は生成される出力ファイルの名前を提供します。

4.6 マッピングから文字列の値を返す

マピングから文字列を返すことが必要な場合単純型出力コンポーネントを使用します。マピングエアでは、単純型出力コンポーネントは、 アイテムとシーケンスの構造の代わりに文字列データ型を持つターゲナコンポーネントの役割をします。この結果、単純型出力コンポーネント (おけば追加して)ファイルベースのターゲナコンポーネントの代わりに作成することができます。.例:単純型出力コンポーネントを使用し て、素早く関数の出力をテストとパビューすることができます関数の出力(例:関数の出力をテストするを参照)。単純型出力コンポーネント トの主な目的は、しかしながら、MapForce Server APIを呼び出す際にファイルに書き込むことがス文字列に戻ることです。

単純型出力 コンポーネントをユーザー定義関数の出力パラメーターと混同しないように注意してください。(<u>ユーザー定義関数</u>を参照)。類 似点と相違点は、以下のとおりです。

出力コンポーネント	ユーザー定義関数出力パラメーター
「関数 出力の挿入」メニューから追加	「関数 出力の挿入」メニューから追加
文字列をデータ型とて持ちます。	単純型と複雑型のデータ型を持つことができます
マルング全体に適用することができます。	定義された方法で関数のコンテキストに適用することができます。

必要であれば、複数の単純型出力コンポーネトをマピングに追加することができます。単純型出力コンポーネトをファイルベースのター ゲトコンポーネトとデータベースと共に使用することができます。マピングが複数のターゲトコンポーネトを含む場合、特定のエンポーネ ントによ返されるデータをプレニーすることができます。コンポーネントタイトルドー内の「プレビュー」(
¹) ポタンをクトックすることにより、 マピングウィンドウの「出力」 タブをクトックします。

単純型出力 コンポーネントをMapForce 変換言語内で以下のように使用することができます。

言語	作動のしくみ
BUILT-IN (マピング変換のプレビュー 時)	ファイルベースのマピング出力をプレビューする同様、マピングウイドウの「出力」 タブをク リックすることにより、出力コンポーネントをプレビューすることができます。
BUILT-IN (MapForce Server 実 行ファイルの実行時)	コンパルされたMapForce Server 実行ファイルを実行すると(MapForce のマナング <u>をエッパル</u> を参照、マナング出力は、標準出力 ストリーム(stdout)内に返され、表示ま たはファイルヘダイレクトすることができます。例:MapForce server 実行ファイルの名前が MyMapping.mfx とます。以下の構文を使用して、マナング出力を output.txt ファイ ルコダイレクトし、全てのエラーを log.txt ファイルログイレクトします: MapForceServer.exe run MyMapping.mfx >output.txt 2>log.txt
XSLT 1.0, XSLT 2.0, XSLT 3.0	生成されたXSLT ファイル内で、マメング内で定義されている単純型出力コンポーネントは XSLT 変換の出力にないます。 RaptorXML Server を使用する場合、RaptorXML Server にマメング出力をファイル に書き込み、output パラメータに値とて、次するように命令することができます。 ファイルに出力を書き込む、DoTransform.bat ファイル内のoutput パラメーターに 追加ませば編集する場合。例 次のDoTransform.bat ファイルは Output.txt ファイ ルへのマメング出力に書き込むために編集されました(ハイライトされたテキスト参照してくたさ い。

	RaptorXML xsltxslt-version=2 input="MappingMapToResult1.xslt" <mark>output="Output.txt"</mark> %* "MappingMapToResult1.xslt"
	output ノラメーターカ定義されていたい場合、マンピングが実行される際、マンピング出 カカ標準出カストリーム(stdout)に書き込まれます。
C++、C#、Java	生成されたC++、C#、とJavaコード内で、マピング出力は、生成されたアプリケーションの標準出力に書き込まれます。
	マピングが複数のターゲトコンポーネントを含む場合、生成されたアプリケーションは、各ター ゲトコンポーネント標準出力を連結して、1つのまとかられた標準出力を返します。

逆のマピングを作成する場合、(メニューコマボ「ツール | 逆マピングの作成」を使用すると単純型出カコンポーネトは、単純型入カコンポーネトになります。

4.6.1 単純型出力コンポーネントの追加

マピングエアに出力コンポーネトを追加する

- 1. ウイドウがソユーザー定義関数ではよりインのマピングを表示していることを確認してくたさい。
- 2. 以下の一つを行います:
 - a. 「関数」メニューから「出力の挿入」をクトックします。
 - b. 「出力の挿入」 レールレーボタンをクリックします。
- 3. コンポーネトのナメの名前を入力します。
- 4. 「OK」をクルクします。

🕑 出力の作成			×
名前(N):	result		
データ型(D):	string		•
	ОК	キャンセル	

出力の作成ダイアログボックス

コンポーネント名を以下の方法で後に変更することができます
- コンポーネトを選択して、「コンポーネント」メニューから「プロノティ」をクリックします。
- コンポーネントへッダーをダブルクトックします。
- コポーネトへダーを右クリックして、「プロノティ」をクリックします。

4.6.2 例: 関数出力のプレビュー

この例は、単純型出力コンポーネートを使用して MapForce 関数により返された出力を確認する方法を説明します。関数に関する基本的理解と、MapForce 関数に関する全般的な知識があると、このサンプルをよく理解することができます。MapForce 関数の基礎知識がない場合、例を開始する前に関数の使用を参照してくたさい。

この例の目的は、マピングエアに複数の関数を追加することで、おき単純型出力コンポーネントを使用して出力をプレビューする方法を理解することです。特に、例はcore ライブラノ内で使用することのできる複数の関数をマピングエノアに追加する方法を説明しています。次に使用方法の概要が述べられています。

<u>string-length</u>	は、引数として与えられた文字列内の文字の数を返します。例 これを関数値 "Lorem ipsum" に やす ると、テキスト "Lorem ipsum" か特つ文字の数である結果は"11" です。
substring-after	は、引数として与えられたセッレーターの後に発生する文字列の一部を返します。例にため関数値 "Lorem ipsum"とスペース文字(" ") に みすると、結果は "ipsum" です。
substring-before	は、引数として与えられたセンレーターの前に発生する文字列の一部を返します。例:これを関数値 "Lorem ipsum"とスペース文字(" ") につぶすると、結果は"Lorem" です。

カスタムテキストの値に対して関数をテストするコまこのサンプルでは、Lorem ipsum ")以下のステップを行います

- 1. 値を持つ定数 "Lorem ipsum" マナングエアを追加します(メニューコマド「挿入 | 定数」を使用して)。定数はテストされる関数のそれぞれの入力/デメーター」ではます。
- 2. string-length、substring-after とsubstring-before 関数をcore ライブラ人 string 関数 セクションか らマングエアイボラッグして、マンピングエアパン追加します。
- 3. 空のスペース(″ ″) を値とて持つ定数を追加します。これは substring-after と substring-before 関数により必要とされるセ シーター・ デメーターとなります。
- 4. 単純型出力コンポーネトを(メニューコマト「関数 | 出力の挿入」を使用して)追加します。他の名前を与えることも可能ですが、このサンプルでは、Result1、Result2、および Result3、と名前が付けられています。
- 5. コンポーネートを下に表示されるように接続してくたさい。



単純型出力コンポーネントを使用して関数の出力をテストする

上のサンプルで示されるとおり、"Lorem ipsum"文字列はそれぞれの、string-length、substring-after、および substring-before 関数入力パラメーターとして振る舞います。更に、substring-after とsubstring-before 関数は、ス ペースの値を第2入カパラメーターとして取ります。Result1、Result2、およびResult3 コンポーネントは各関数の結果をプレビューす るけっかに使用することができます。

関数の出力をプレビューする方法:

• コポーネトタイトル・一内のプレビュー」(10) ボタンをクリックし、マピングウィボウの「 出力」 ダ港クリックします。

4.7 変数の使用

変数とは更なる処理のために中間マピングの結果を保管するために使用される特別なエンポーネントです。マピング上のデーダを一時的に 「記憶」し、処理する必要かある場合があります。例えば、ターゲナコンポーネントにエピーされる前に、フィルター、ませば関数を適用するな ど。

変数は、単純型(例えば、文字列、整数、ブール値、など)ませま複合型(ツノー構造)であることができます。

VAR	
Self compute-when	1
⊵ <mark>⊭≣value</mark>	Ş

単純型変数

変数の構造を示す XML スキーマを提供して複合型の変数を作成することができます。スキーマか要素をグロー・ ジルニ定義すると、変数構 造のルートノード こなるものを選択することができます。 変数に関連した インスタンス XML ファイルが存在しない場合、 変数のデータはマッピ ングランタイム こ計算されます。

VAR:		
200	compute-when	
¦>⊒ () t	ooks	P
¢ ¦⊖ ()	book	¢
¢	= id	Ь
5	() author	Ę
\$	() title	ķ
÷	() category	Þ
¢	() year	Ļ
l		

XML スキーマから作成された複合型変数

データベースから複合型の変数を作成することも可能です。データベースの場合は、変数構造のためのルートアイテムとして特定のデータ ベーステーブルを選択することができます。

VAR	
Compute-when	
	Ŗ
🖒 🔤 🕼 ApplD	¢
🖒 🖳 AppName	ķ
🗟 🔲 Description	ķ
ခု 🔤 🔲 Category	¢
🖒 🦾 📶 URL	Þ
L	_^

データベーステーブルから作成された複合型変数

上のイメージでは、各変数には、compute-whenとう名前のアイテムが存在することに注意してくたさい。このアイテムは接続することは 任意です。これにより、マノピング上でどのように変数の値が計算されるかを管理することができます、次を参照してくたさい、変数のコンテキスト とスコープの変更。

必要な場合、通常のエンポーネントと同様に、変数構造のアイテムを1つ以上のノース接続から受け入れられるように複製することができます (次を参照してくたさい、入力の複製)。これは、しかしなからデータベーステーブルから作成された変数には適用されません。

VAR	
Compute-when	_
¦> ₩ i value	[
(>₩İ value (2)	
>₩İ value (3)	

複製された入力を持つ単純型変数

変数に関して最も重要な点は、変数はシーケンスであり、シーケンスを作成するさめに使用されるとう点です。「シーケンス」とう用語は ここでは、ゼロおさは、それ以上のアイテムのリストを意味します(次も参照してください、マッピングのルールと戦略)。これにより、変数はマッピ ングの寿命内で複数のアイテムを処理することができるようになります。しかしなから、値を変数に与え、マッピングのその他の部分を同じに保つ ことも可能です(次を参照してください、変数のコンテキストとスコープの変更)。

ある程度までは、変数をチェーンマとングの中間コンポーネントと比較することができます(チェーンマンピングを参照してくたさい)。しかしなが ら、マメニング内の各段階で中間ファイルを作成する必要がない場合、変数は柔軟性があり便利です。次のテーブルの概要は変数とチェー ンマメニングの違いを示しています。

チェーンマッピング	変数
チェーンマンピングゴま、2つの独立したステップが含まれます。例え ば、A、B、とCの3つのエレポーネトがマンビングご存在すると ます。マンピングの実行にコセンの段階があります: A からB への マンピングの実行、および、B からC へのマンピングの実行。	マピングの実行中、変数は、コンテキストとスコープゴビル 評価されます。コンテキストとスコープを影響することはできます(<u>変数のコ</u> ンテキストとスコープの変更を参照してくたさい)。
マルングが実行されると、中間結果はおい部でファイルに保管されます。	マノピングが実行されると、中間の結果は内部で保管されます。変数の結果を含む外部ファイルは作成されません。
中間結果は ・ なたを使用して ルビューする ことができます。	マパングラタイムで計算される変数の結果をパビューすることは できません

メモ 変数はマピング変換言語がXSLT 1.0 に設定されていない場合サポートされません。

4.7.1 変数の追加

マピングに変数を追加するコよ以下に示されているようことくつかの方法があります。

メニュー、またはツール・コマドの使用

1. 「挿入」メニューから「変数」をクリックします。(「変数」 いール いってな を代わりにクリックします)。

🕑 変数の作成	
型 一型 逆 単純型(integ	er, string など)(S)
デ─久型(D):	string ▼
◎ 複合型(ツリー	-構造)(C)
構造(U):	選択(H) 編集
ルート(R):	
	☑ 構造ファイルパスをMFDファイルから相対的に保存(V)
	OK キャンセル

2. 挿入する変数の型を選択します(単純型 ませま複合型)。

"複合型"を選択した場合、追加ステップが存在します

3. 変数の構造を与えるノースを選択するために「選択」をクリックします(例えば、XML スキーマ、まけはデータベース)。

	— ×
挿入する既存の構造または新しい構造の型を選択してください:	
◎ 既存の構造をパラメーターとして挿入する(E):	
ファイル名 / 接続パス / 記述	^
	-
◎ 以下で選択された型の新規構造を挿入する(N):	
 	
回 EloyTayt構造	
OK Th	

4. プロンプトされると、構造のルートアイテムを指定します。XML スキーマの場合、ルート アイテムよ グロー・ ジルニ定義されている す べての要素であることができます。 データベースの場合は、ルートアイテムよすべてのテーブルであることができます。

♥ルートアイテムの選択	×
パラメーターのルート アイテムを選択してください:	
	^
⊞neensensensensensensensensensensensensen	
	-
4	
✓ 注釈を表示(A)	
型の表示(T)	OK キャンセル
	///

コンテキストメニューの使用

• コンポーネトの出力コネクタを右クトックします(このサンプルでは、「Article」) そして、「ソースノードから変数を作成する」を選択します。

😫 Articles]
[□□ ファイル : Articles.xml <u>ファイル/文字列</u>] □	P
¢ ₽ () Articles	¢
🖞 🗄 🖓 Article	
Number	全ての出力子要素の削除(0)
🗟 🕂 🗘 Name	いニフ ノードから亦物を作成(c)
Single Price	シースノードル・ウ友妖を1F/3(3)

これにより、同じソーススキーマを使用する複合型の変数を作成し、自動的にすべてのアイテムをすべてをエピーする接続により接続します。



- ターゲトコンポーネトの入力コネクタを右クトックして、「ターゲットノードのために変数を作成する」を選択します。これにより、 同じスキーマをターゲトとして使用し、複合型の変数を作成します。そして、すべてコピー接続を使用してすべてのアイテムを自動 的に接続します。
- フィルターコンポーネント (on-true/on-false)の出力コネクタを右クトックして、「ソースノードから変数を作成する」を選択します。これにより、ソーススキーマを使用して複合型コンポーネントを作成し、フィルター入力にコンクされているアイテムを中間コンポーネントのレーン要素として自動的に使用します。

4.7.2 変数のコンテキストとスコープの変更

各変数には、変数のスコープを管理することを許可する compute-when 入力アイテムが存在します。すなわち、マピングが実行される際 に変数の値がいへ どの頻度で計算されるかを管理することができます。この入力は、多くの場合接続する必要がありませんが、デフォルトのコ ンテキストを上書きする場合、ませまマピングの、フォーマンスを最適化するこよ、必要です。



「Compute-when」 アイテム

次のテキストでの、サブソレーは、ターゲトコンポーネント内のアイテムノノードセナとその子孫を意味します。例えば、<FirstName>と <LastName> 子要素を持つ、<Person> 要素。

変数の値は変数コンポーネトの出力サイドでデータが使用できることを意味します。

- 単純型の変数に関しては、コンポーネントプロ、ティ内で指定されているデータ型を持つ動的な値のシーケンスであることを意味します。
- 複合型の変数に関しては、それぞれが自身の子孫ノードを含む(コンポーネント プロ/ ライ内で指定されている型の)ルートノードの シーケンスであることを意味します。

動的な値(おさよノード)のシーケンスは、1つの要素、おさま要素を全く含またい場合があります。これは、変数の入力サイドに可か接続されているか、および、ソースとターゲトコンポーネント内の親アイテムの存在により異なります。

「Compute-when」が接続されていない場合(デフォルト)

compute-when 入力アイテムがソースエンポーネントの出力ノードに)接続されていたい場合、ターゲットサブソノー内で最初に使用される際に変数の値はコネクタには変数エンポーネントからターゲットコンポーネント内のノードに直接、おけま関数を使用して間接的に)計算されます。

実際の変数の値は、ソースとターゲトコンポーネトの親アイテム間の接続によ興なります。

このデフォルトの振る舞いは、正規ユーザー定義関数とWebサービス関数の呼び出しの複合型の出力と同じです。

変数の出力が複数の関連したになーゲトノードに接続されている場合、変数の値は、それぞれのアイテムのために個別に計算されます。これにより、各ケース内で異なる結果を生成することができます。これは異なる親の接続は、変数の値が評価されるコンテキストを影響するからです。

「Compute-when」が接続されている場合

ソースコンポーネントの出力コネクタを compute-when は接続すると、ソースアイテムか最初にターゲト・サブソノー内で使用される都度に変数が計算されます。

変数は実際には、compute-whenに接続されているアイテムの子アイテムのように振る舞います。これは、新規のアイテムがソースエンポー ネント内のシーケンスから読み込まれる都度、ランタイム変数が再評価されます。これは、MapForce内の接続を管理する一般的なルール に関連しています:各ソースアイテムのために、1つのターゲットアイテムが作成されます。compute-whenに関しては、各ソースアイテムの ために、変数の値が計算されます、次を参照してくたさい、マンピングのルールと戦略)。

[Compute-once]

必要であれば、ターゲトコンポーネトの前に」度変数の値を計算し、変数をマンピングの残りでグロー・シレな定数にすることを選択することが できます。これをおこなうコよ、compute-when アイテムを右クリックして、コンテキストメニューから「1度計算する」を選択します:

VAR		
P <u>■ value</u>	一度計算(O)	
	接続の削除(D)	►
	変数コンポーネントの作成(V)	→
	コンポーネント(C)	►

変数のスコープを compute-when=once に変更する場合、このような変数は1度のみ評価されるため、入力コネク タは、 compute-when アイテムから削除されます。

実際の関数の結果が評価される前に、ユーザー定義関数 compute-when=once 変数は関数が呼び出される都度評価されます。

親コンテキスト

親エンテキストを追加する必要かある場合かあります。例えば、マンピングか複数のフィルターを使用し、反復するために親ノードを追加する必要がある場合など。例<u>親エンテキストの変更</u>を参照してくたさい。

変数に親エンテキストを追加するコよルートノード(このサンプルでは「PersonList」)を右クリックします、そして、エンテキストメニューから、「親コンテキストを追加する」を選択します。これにより新規ノード、親コンテキストを既存の階層構造に追加します。



親コンテキストは、仮想の「親」ノードをコンポーネント内の階層構造に追加します。これにより、同じ、ませま異なるノースコンポーネント内で追加ノードを反復することができます。

4.7.3 例: データベースのテーブルの行の計算

このサンプルで説明されているマメピングは、<マイドキュメント>¥Itova¥MapForce2021¥MapForceExamples¥フォルダー内の DB_UserList.mfd です。このマメピングは、ユーザーのレコードを "Users" という名前のデータベーステーブルから抽出し、XML ファイル に書き込みます。データベースの列 "Username" には、個人の姓と名の両方が含まれています(例えば、"Vernon Callaby")。このマッ ピングの目的は以下のとおりです。

- 1. "Users" テーブル内の各レコードのために、XML ファイル内に新規のPerson 要素を作成する。
- データベースフィールド "Username" から抽出された値を XML ファイル内の個別のフィールド("First" と"Last")に分割する。
 各レコードのために、データベース内に存在するレコードの全体の番号に対して比較される連番を検索し(例えば、"4 件中の1件目"など) Details 要素にこの情報を書き込みます。



DB_UserList.mfd

上で説明されているとおり、最初の目的を達成するために、ソース "Users" テーブルとターゲナ XML ファイルの Person 要素の間に接続が描かれます。これにより、ソーステーブル内の各レコードのために、ターゲナト内に新規の Person 要素が作成されます。

フィールド "Username" の値が substring-before と substring-after 関数に与えられています。これらの2つの関数は、スペース文字("")の前後のテキストをそれぞれ抽出し、このマンピングの目的を達成します。

3つ目の目的を達成するながに、マピングは count 関数を使用します。集計関数の結果は、変数に マピングルに に結果が保管され、各個人の "Details" 要素 をターゲナ XML に書き込む際に使用することができます。効率性のために、データベース のレコードは1度のみ数えられるへきであり、変数のスコープは compute - when = once に設定されています(次を参照してくたさい) 変数 のコンテキスト とスコープの変更)。

4.7.4 例: ノードのフィルターと番号付け

このサンプル内で説明されているマピングのシーケンスは、〈マイドキュメント〉 ¥ltova ¥Map Force 2021 ¥Map Force Examples ¥ フォルダー内の Position In Filtered.mfd で使用することができます。

このマピングは、複数の個人の連絡先のデータを含むXML ファイルを読み取り、フィルターし、ターゲオ XML ファイルに書き込みます。このマピングの目的は、ソースXML ファイルから、「M」 ませよ、以降のアルファベットから始まる姓を持つ個人をフィルターします。 抽出された 連絡先には、番号がコガンれている必要があります。 番号は、ターゲオ XML ファイル内で各連絡先の一意の識別子としての役割を果たします。



PositionInFilteredSequence.mfd

上の目的を達成するコよ、次のコンポーネント型がマングに追加されまた

- フィルター(次を参照してくたさい)フィルターと条件)
- 複合型変数 (次を参照してくたさい) 変数の追加)
- 関数 greater と position (次を参照してけきい) 関数をマメングづらかする)
- 定数(定数を追加するコよ、メニューコマボ「挿入 | 定数」を選択します)。

変数は、ソースコンポーネントとして同じスキーマを使用します。変数を右クトックして、コンテキストメニューからプロパティを選択すると、この変数構造のためコレトトノードとしてノード BranchOffices/Office/Contact が選択されていることを確認してくたさい。

最初にソースエンポーネトのデータはフィルターコンされます。フィルターは変数にフィルターの条件を満たすレコードのみをパスします。具体的コよフィルターは最初の名前が「M」に等しい、おけま大きい値を持つContact ノードのみを取得するようご構成されています。この目的を達成するコよ、関数greater は各last アイテムを定数値「M」と比較します。

変数には、ソースコンポーネント(BranchOffices)のレートアイテムに接続されている compute-when 入力がみます。ランタイムで は、これは、ソースコンポーネント内のシーケンスから新規のアイテムか読み込まれる都度、変数を再評価するようにします。このアメピングで は、しかしなから、 compute-when アイテムの接続、おけば未接続に違いなみがません。この理由は、変数が、フィルターを使用して間接的 に)、 Contact ソースアイテムに接続されており、フィルター条件を満たす Contact のインスタンスの数だ」たけ算かになけれるからです。

position 関数は、変数の各反復のために現在のシーケンスの番号を返します。8件の連絡先がライルターの条件を満たしています。です から、マピングをプレビューして、出力を確認すると、1から8のIDが、ターゲナイコンポーネントのID要素に書き込まれていることが確認 することができます。

変数の必要性に関しては、すべてのレコードに番号付けをする必要性に応じるものです。フィルターの結果を直接ターゲトコンポーネントに 接続すると、Contact の各発生に番号を付けることができません。このマピング内の変数の目的は、ですから、Contact の各个ンスタン スを一時的にマピングに保管し、ターゲットに書き込まれる前に番号を付けることです。

4.7.5 例:記録のグループ化、および、サブグループ化

このサンプルで説明されているマピングは、〈マイドキュメント〉¥Altova¥MapForce2021¥MapForceExamples¥フォルダー内の DividePersonsByDepartmentIntoGroups.mfd です。

このマナングは、架空の会社の社員のレコードを含むXML ファイルを処理します。社内には2つのオフィスが存在します。「Nanonull, Inc.」と「Nanonull Partners, Inc」。各オフィスコは複数の部署が存在します(例えば、「IT」、「マーケティング」など)、おこ、各部署 には一名以上の社員が存在します。マナングの目的は、オフィスニ関わびよく各部署から3人までで構成されているグループを作成することで す。グループのサイズはデフォルトでは3名です。しかしながら、必要に口応じて変更することもできます。各グループは、フォーマッ ト^ベ、Department Name>_ GroupN[™]の名前を持つ個別のXML ファイルとして保存される必要があります(例えば、 Marketing_Group1.xml、Marketing_Group2.xml、など)。



DividePersonsByDepartmentIntoGroups.mfd

上て説明されているとおり、マンングの目的を達成するココよ、複合型変数と主に関数である他のコンポーネント型をマンングに追加します。ソースXML内のDepartment アイテムと同じ構造が変数ココな存在します。プロンテを表示するオーダン変数を右クトックすると、ソースコンポーネントと同じ XML スキーマか使用されていること、および、ルート要素として Department が存在していることに気が付きます。 重要な点は、各部署のコンテキスト内で変数が最初に計算され、次に各グループのコンテキスト内で計算されるように、変数には、2つのネストされた親コンテキストアイテムが存在することです(次も参照してくたさい、変数のコンテキストとスコープの変更)。 始めこ、マッピングは、各部署の名前を取得するためにすべての部署を反復します(これは各グループに対応するファイル名を後に作成するために必要とされます)。これは、group-by 関数を Department ソースアイテムに接続すること、部署名をグループ化のキーとして与えることに建成されます。

次に、各部署のエンテキスト内で、2番目のグループ化が発生します。具体的にコシン要とされるーのグループを作成するオンパ、マピングが group-into-blocks 関数を呼び出します。デフォルトの値が "3" である単純型のエレポーネントが各グループのサイズを提供します。デ フォルトの値は、定数により与えられます。グループのサイズを変更するオンパニ、このサンプルでは、必要に応じて定数を簡単に変更すること ができます。しかしながら、 "サイズ" 入力コンポーネントも変更することができ、マピングが生成されるコード、おけはMapForce Server により実行される場合、 グループのサイズは、マピングへの、ラメーターとして便利に提供することができます。 更に詳しい情報に関しては、次 を参照してくたさい、マピングにつ、ラメーターを与える

次に、変数の値はターゲオ PersonList XML コンポーネトにより与えられます。作成された各グループのためのファイル名は、concat 関数を使用して次の部分を連結することにお計算されます:

- 1. 各部署の名前
- 2. 文字列 "_ Group"
- 3. 現在のシーケンス内のグループの番号(例えば "1"はこの部署の最初のグループを指します)
- 4. 文字列 ["].xml"

この連結の結果は、変数のName アイテム内に保管され、ターゲトコンポーネントに動的なファイル名として与えられます。これにより、受信した値のからに新規のファイルが作成されます。このサンプルでは、変数は、8個のグループを計算するため、マンピングが実行されると3件の出力ファイルが作成されます。この技術に関する更に詳しい情報は、次を参照してくたさい、複数の入力おける出力ファイルを動的に処理

4.8 データの並べ替え

特定のノーキーをベースした入力データの並べ替えを行う場合、ソートコンポーネントを使用してください。ソートコンポーネントは XSLT2、XQuery、内蔵の実行エンジノニてサポートされます。変換言語が「Built-in」の場合、並べ替えコンポーネントを使用して、デー タベーステーブルのデータを並べ替えることができます。しかしながら、パフォーマンスを向上するコは、SQL-WHERE/ORDERコンポーネントを使用することに達成することができます。詳細に関しては、次を参照してくたさい、データベースデータのフィルターと並べ替え(SQL WHERE/ORDER)。

マッピングに並べ替えコンポーネントを追加するには、以下を行います

• 既存の接続を右クルクして、コンテキストメニューから Insert Sort: Nodes/Rows」を選択します。並べ替えコンポーネント を自動的に挿入し、ソースとターゲトコンポーネントに自動的に接続します。例えば、下のマンピングでは、並べ替えコンポーネント トが変数とXML コンポーネント間に挿入されます。(並べ替えるフィールドである)並べ替えキーのみを手動で接続します。



• 「挿入」メニューから「並べ替え」(おけよ「並べ替え」 「 替え」ンポーネトか挿入されます。



ソースコンポーネトに接続が構築されると、タイトルモ名は nodes/rows アイテムは接続されているアイテムの名前に変更されます。

並べ替えるアイテムを定義する

• 並べ替えコンポーネントのkey パラメーターを並べ替えるアイテム」接続します。例えば下のマメピングでは Person nodes/rows は フィールド Last に従い並べ替えられます。



並べ替えの順序の変更:

・ 並べ替えコレポーボト内の▲→Z アイコンをクリックすると、 Z→A に変更され、並べ替えの順序は逆の順序になったことを示します。

単純型アイテムにより構成される入力データの並べ替え

• アイテムを並べ替えてレポーネトのhodes/rowsとkey パラメーターは接続します。下のマピングでは単純型 first 要素が並べ替えられます。



言語特有のルールを使用して文字列を並べ替える

• 並べ替えコンポーネントのヘッダーをダブルクトックして、「並べ替え」プロ・ティダイアログボックスを開く。

😢 並べ替えのプロ/	パティ 🔀
文字列の並べ替え(□使用する照合順序を選択して下さい。
◎ Unicode コードポイ	イント照合順序 <mark>(U)</mark>
◎言語固有の照合	⊁順序 (T)
「ロケールーーーー	
言語(L):	•
国 (C):	•
	<u>ок</u> キャンセル

Unicode コードポイント 照合: この(デフォルト)オプションにより、コードポイントの値をベース した比較/並べ替えか行われます。コードポ イント値とは Unicode コンノーシアムによ採用された Universal Character Set (UCS) における絶対文字に割り当てられた整数の ことです。このオプションでは複数の言語やスクレプト間で並べ替えを行うことが可能しています。

言語固有の照合: このオプションにより、並べ替えを行う際にベースとなる言語や国を指定することができます。このオプションは内蔵の実行 エンジン(BUILTHN)が選択されている際にサポートされ、XSLTにおけるサポート」はコードの実行に使用されるエンジンに左右されます。

4.8.1 複数のキーを使用した並べ替え

マピングに並べ替えコレポーネントを追加した後、keyとう名前の並べ替えキーがデフォルトで作成されます。

	A⊈sort		
Ľ	nodes/rows	rooutth	l
Ľ	key A=>Z	result.	ſ

デフォルトの並べ替えコンポーネント

複数のキーを使用して並べ替える場合、並べ替えコンポーネントを以下のように調整します

- 「キーの追加」(•) アイコンをクリックして新しいキーを追加します(例えば、下のマンピング内の key2)。
- 「キーの削除」(図)アイコンをクルクしてキーを削除します。
- ● アイゴノ 接続をドロップして、キーを追加し、接続します。

複数のキーによる並べ替えを示すマンピングは、次の、やで検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\SortByMultipleKeys.mfd.



SortByMultipleKeys.mfd

上のマピングでは Person レコードはつの並べ替えキーには述べ替えられています:

- 1. Shares (個人か所有するシェア数)
- 2. Last(姓)
- 3. First (名)

並べ替えコンポーネント内の並べ替えキーの位置は、並べ替えの優先順序を決定します。例えば、上のマンピングでは、所有するシェア数に よし述べ替えられています。これは、最も高い優先順位を持つ並べ替えキーです。所有するシェア数か同じの場合も同様で、姓によし述べ替 えられます。最後に、同数のシェアを持つ同じ性を持つ個人は、名を考慮して並べ替えられます。

各キーの並べ替えは異なる順番であることができます。上のマンピングでは、キーShares は降順の並べ替え順序(Z-A)か与えられてしますが、その他2つのキーコは昇順の並べ替え順序(A-Z)か与えられてします。

4.8.2 変数を使用して並べ替える

希望する結果を達成するためにマピングに中間変数を追加する必要がある場合があります。このサンプルでは、XML ファイルから記録を抽出し、中間変数を使用して並べ替える方法について説明されています。このサンプルゴよ、次の・なで検索することのできるマピングのサンプ ルが存在します: 〈マイドキュメント〉 ¥ltova ¥lap Force2021 ¥lap ForceE xamples ¥ltova_ Hierarchical_ Sort.mfd。



Altova_Hierarchical_Sort.mfd

このマピングは、Altova_Hierarchical.xmlとら名前のノースXMLファイルからデーを読み取り、ターゲオ XMLファイルに書き 込みます。上に示されるとおり、ソースXML には架空の企業の情報が含まれています。企業にオフィスに分岐されています。オフィスは部署に 分岐されており、部署に社員に分岐されています。

ターゲット XML コンポーネト、PersonList、はPerson レコードのノストが含まれています。Details アイテムは、社員が属するオ フィスと部署の情報が保管されています。

ソースXMLから全ての社員を抽出し、姓をアルファベナ順に並べ替えることが目的です。また、オフィスと部署名は、Details アイテムに書き込まれる必要があります。

この目的を達成するために、このサンプルは、次のエレポーネトの型を使用します

- 1. concat 関数: このマピング内では、この関数は、フォーマトOffice (Department)で文字列を返します。オフィス名、部署名、かつ開始し終了する2つの定数を入力として取ります。関数をマパングに追加するた参照してくたさい。
- 2. 中間変数: 変数の役割は同じマピングコンテキストに個人に関連する全てのデータを運ぶことです。変数によりマピングかそれぞれの個人のエンテキストで、個人の部署とオンイスを探します。すなわち、変数は、個人か所属するオフィス名と部署名を「記憶」することを意味します。変数が不在の場合、コンテキストは、(同じ XML スキーマを使用して)正確ではなくマピングは希望しない結果を生成します。(マピングの実行方法の詳細に関しては、次を参照してくたさい: マピングルールと戦略)。変数は XML ファイルの構造を複製することに注意してくたさい。これにより、全てコピー接続を使用して、ターゲットに並べ替えの結果を接続することができます。次も参照してくたさい、変数の使用と全てコピー接続。
- 3. 実際の並べ替えを行う、並べ替えコンポーネント:Sort コンポーネントのキー入力は、姓により個人の記録を並べ替える変数の Last アイテムに接続されていることに注意してくたさい。

4.9 データのグループ分け

マルングがシード、または行をグループ分けする必要がある場合、次のMapForce内蔵関数を使用してこの目的を達成することができます。

- group-by
- group-adjacent
- group-into-blocks
- group-starting-with
- group-ending-with

マルング上でこれらの関数を使用する場合、ライブラリウィドウからドラッグしてマッピングエリアドドロップします。関数をマッピングに追加するも参照してくたさい。

メモ グループ関数は次の変換言語内で使用することができます: XSLT 2.0、XSLT 3.0、C++、C#、Java、Built-In。

次のセクションではグルーフ関数のすみに使用することのできる一般的なサンプルは挙じたれています。これらのサンプルは次のデモマ・ピングが 伴います: 〈マイドキュメント〉 Witova Wap Force 2021 Wap Force Examples Witorial W Grouping Functions.mfd。デモ マビングロは各関数のすみロコ つずつ与えるすみの複数の変換が含まれています。一度にプレビュー可能な出力はいつであるすみ、希望する 変換に適用するプレビュー 100 ボタンを「出力」 タブをクリックする前に、 にクリックしてくたさい。

group-by

group-by 関数は、指定するグループキーイン従いレコードのグループを作成します。

f _x group-by	
hodes/rows	groups 🗘
⊳key	key Þ

例えば、下で示される抽象的な変換では、グループキーは、「Department」です。トータルで3つの一意の部署が存在するため、 group-by を適用すると3つのグループが作成されます:

		1	Department	Name
		i	Administration	Verno
Department	Name		Administration	Frank
- Administration	Vernon Callaby			
Marketing	Susi Sanna			
Engineering	Michelle Butler	group-by()	Department	Name
Engineering	Fred Landis		Marketing	Susi S
Administration	Frank Further			
		1	Department	Name
		1	Engineering	Michel
		1	Engineering	Fred I

詳細は関しては group-by 関数を参照してくたさい。

group-adjacent

group-adjacent 関数はgroup-by 関数と同様、グループキーを引数としてグループ分けすることを必要とます。group-by とは、 異なり、この関数は次のキーが異なる都度新規のグループを作成します。2つの隣接するレコードが同じキーを持つ場合、これらのアイテムは 同じグループコ配置されます。

例えば、下記の抽象的な変換に示されるように、グループキーはDepartment」です。このダイアグラムの左側は入力データを示し、右側は グループ分けの後の出力データを示しています。次は、変換の実行中に発生します:

- 最初は最初のキー「Administration」は新規のグループを作成します。 •
- 次のキーは異なるため、2番目のグループである「Marketing」が作成されます。
- 3番目のキーもお生異なるための「Engineering」と呼ばれる他のグループが作成されます。 •
- 4番目のキーは3番目と同じキーです。このため、レコードは既存のグループに配置されます。
- 最後に、5番目のキーは番目のキーと異なるため、最後のグループを作成します。 •

下に説明されるとおり、「Michelle Butter」と「Fred Landis」は隣接する同じキーを持っために一緒にグループ分けされています。しか しなから、「Vernon Callaby」と「Frank Further」は同じキーを所有しているにもかわらず隣接していたいため、異なるグループに分 けれてます。

			Department	Name
			Administration	Vernon Callab
partment	Name			
ministration	Vernon Callaby		Department	Name
cketing	Susi Sanna		Marketing	Susi Sanna
gineering	Michelle Butler	group-adjacent()	, , , , , , , , , , , , , , , , , , ,	
gineering	Fred Landis			
ministration	Frank Further		Department	Name
			Engineering	Michelle Butl
			Engineering	Fred Landis

Department	Name	
Administration	Frank	Further

詳細に関しては、group-adjacent 関数を参照してくたさい。

group-into-blocks

group-into-blocks 関数は、N がblock-size 引数に与えられる値である、N アイテムを含む等価のグループを作成します。最 後のグループは、ソース内のアイテムの数に従い、N つのアイテム おけおりない数のアイテムを含むことができます。下のサンプルでは、 block-sizeは2です。総数でうつのアイテムが存在し、最後のアイテムを除き、それそれのグループはつのアイテムを含んでします。



詳細に関しては、group-into-blocks 関数を参照してたさい。

group-starting-with

group-starting-with 関数はブール式の条件を引数として取ります。関数はブール式の条件がtrueの場合、条件を満たすレコードから、新規のブループが作成されます。下のサンプルでは、「Key」が heading」と等価であることが条件です。最初と4番目のレコードのために、この条件がtrueの場合、2つのブループか結果として作成されます。

			Кеу	Value
Кеу	Value		heading	Intro
heading	Intro	N	line	A
line	A		line	в
line	в	group-starting-with()		
heading	Body	· · · · · · · · · · · · · · · · · · ·		
line	c		Кеу	Value
		1	heading	Body
			line	С

メモ 条件を満たす最初のレコードの前にしコードが存在する場合、追加のグループが作成されます。例えば、「line」レコードが最初の「heading」レコードは比多く存在する場合、これらは新しレヴループ内に配置されます。

詳細に関しては、group-starting-with 関数を参照してくたさい。

group-ending-with

group-ending-with 関数はブール式の条件を引数として取ります。関数はブール式の条件がtrue の場合、条件を満たすレコードが 最後になるように、新規のグループが作成されます。下のサンプルでは、「Key」が「trailing」と等価であることが条件です。3番目と5番 目のレコードのために、この条件がtrue の場合、2つのグループが結果として作成されます。

		Key	Value
Value	group-ending-with()	line	A
A		line	в
В		trailing	Total 1
Total 1	group-ending-with()		
C			
Total 2		Кеу	Value
		line	C
		trailing	Motol 2
	Value A B Total 1 C Total 2	Value A B Total 1 C Total 2	Value Key A line B group-ending-with() Total 1 Key C Total 2 Key line Ine trailing

メモ 条件を満たす最後のレコードの前にレコードが存在する場合、追加のグループが作成されます。例えば、「line」レコードが最後の「heading」レコードより比多く存在する場合、これらは新しいグループ内に配置されます。

詳細に関しては、group-ending-with 関数を参照してくたさい。

4.9.1 例: キー別にレコードをグループ分けする方法

このサンプルは、group-by 関数の手助けと共にグループ分けする方法、および、データを集計する方法を示しています。このサンプルコよ 次の、次で使用することのできるデモマンピングが付随しています: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\GroupTemperaturesByYear.mfd。このマンピングは、月ごとの気温を 含むXML ファイルわらデータを読み取ります:

```
<Temperatures>
  <data temp="-3.6" month="2006-01" />
  <data temp="-0.7" month="2006-02" />
  <data temp="7.5" month="2006-03" />
  <data temp="12.4" month="2006-04" />
  <data temp="16.2" month="2006-05" />
  <data temp="19" month="2006-06" />
   <data temp="22.7" month="2006-07" />
  <data temp="23.2" month="2006-08" />
  <data temp="18.7" month="2006-09" />
  <data temp="11.2" month="2006-10" />
  <data temp="9.1" month="2006-11" />
  <data temp="0.8" month="2006-12" />
  <data temp="-3.2" month="2007-01" />
  <data temp="-0.3" month="2007-02" />
  <data temp="6.5" month="2007-03" />
  <data temp="10.6" month="2007-04" />
  <data temp="19" month="2007-05" />
  <data temp="20.3" month="2007-06" />
  <data temp="22.3" month="2007-07" />
  <data temp="20.7" month="2007-08" />
  <data temp="19.2" month="2007-09" />
  <data temp="12.9" month="2007-10" />
  <data temp="8.1" month="2007-11" />
   <data temp="1.9" month="2007-12" />
</Temperatures>
```

このマピングのビジネスの条件は2段階で構成されています

- 1. 各年の気温をグループ分けします。
- 2. 各年度の最低、最高、および平均機能を検索します。

最初の目的を達成するコよ、マピングはgroup-by 関数を呼び出します。2番目のゴールを達成するコよ min、max および avg 集計関数を呼び出します。これらすべての関数はMapForce ビルトイン関数で、ライブラリウィンドウからマピング ゴ・ラッグして追加することができます。
関数をマンピング ご追加するを参照してください。



GroupTemperaturesByYear.mfd

MapForce がマピングを実行する方法は(そして読み取りを開始する」かに奨励されるアプローチ)ターゲトコンポーネントの一番上のア イテムを確認することです。このサンプルでは、YearlyStats アイテムがgroup-by 関数により返される各グループのけっかに作成されます。 group-by 関数はkey 入力に持続が作成されると、最初の引数として、すべてのdata アイテムをノースとグループから取ります。条件 か気温を年度ごとごグループ分けすることであり、年度が最初に取得される必要かあります。これを達成するには、<u>substring-before</u> 関数は、各 data 要素のmonth 属性から年度の部分を抽出します。具体的には、monthの値を引数としてとり、の最初の substrの発生の前の部分に返します。上記の通り、このサンプルでは、substr は点線の文字のセットであるため、値「2006-01」を 与えられると、関数は「2006」を返します。

最後に、MinimumTemp, MaximumTemp、とAverageTempの値は、これらのアイテムを対応する集計関数に接続することに よ取得することができます: min、max、とavg。全ての3つて関数は、ソースコンポーネントから読み取られた入力として温度のシーケンス を取ります。

これらの関数は、各グループのエレテキストと既に作業しているためparent-context 引数を必要としません。すなわち、作業する各集計 関数のためのエレテキストを提供する data から YearlyStats への親コネケンヨンの存在します。

マピングの出力をルビューするコよ Output ダブをクリックしてください。ソースファイルの読み取りて取得された年数とグループの数量が 一致することに注意してください。例えば、

```
<Temperatures>

<YearlyStats Year="2006">

<MinimumTemp>-3.6</MinimumTemp>

<AverageTemp>11.375</AverageTemp>

</YearlyStats>

<YearlyStats Year="2007">

<MinimumTemp>-3.2</MinimumTemp>

<AverageTemp>11.5</AverageTemp>

</YearlyStats>

</Temperatures>
```

メモ 簡単化のために、上記のロードリストには、デモマノピングで使用されている実際の入力と出力よりた少ないデータを含んでいます。

4.10 フィルターと条件

データをフィルターする場合、おけは植を条件につして取得する場合、以下のうち1つのコンポーネント型を使用することができます:

- フィルター: Nodes/Rows (축)
- SQL WHERE/ORDER (🚟)
- If-Else 条件(๋ ๋ ๋ ๋ : `)

「挿入」メニューから、ませま「コンポーネントの挿入」、ツールレーからこれらのコンポーネントをマンピングに追加することができます。上記のそれぞれのコンポーネントには特定の振る舞いと必要条件があることに注意してくたさい。差異は下のセグションで説明されています。

ノードと行をフィルターする

XML ノード おけまCSV 行を含むデータをフィルターする場合、Nodes/Rows をフィルター コンポーネントを使用します。この Nodes/Rows をフィルター コンポーネントにより、 true おけまfalse 条件をベースに、大きなデータセントからノードのサブセンを抽出 することができます。 マピングエルアでのこの構造は以下のように表されます:

😴 filter			
>node/row	on-true(;		
>bool	on-false(;		

上の構造では、は接続されている条件は、bool に接続された node/row が、on-true おけまのn-false 出力は接続されているかを 決定します。具体的には、条件がtrue の場合、node/row は、on-true 出力に以れてかされます。一方、条件がfalse の場合、 node/row は、on-false 出力に以れてかされます。

マピングがフィルターの条件を満たすアイテムのみを消費する必要かある場合、on-false 出力を未接続のままこできます。フィルターの条件 を満たされ、アイテムを処理する必要かある場合、このようなアイテムがダイレクトされる、on-false 出力をターゲナトに接続します。フィルタ ーの条件が満たされ例外を追加する場合は、on-false 出力への接続は必須です(以下を参照してください、例外の追加)。

手順を追ってのマピングのサンプルは、例:ノードのフィルターを参照してくたさい。.

データベースのデータをフィルター

Nodes/Rows をフィルターコンポーネントは、MapForce により、他のコンポーネント構造からデータをフィルターすることがでます。しか しながら、データベースからデータをフィルターする場合、SQL WHERE/ORDER コンポーネントを代わりに、使用することが奨励されます。 SQL WHERE/ORDER コンポーネントは、データベースと作業し、Nodes/Rows をフィルターコンポーネントより上良し v ウォーマンス を与えるためこ最適化されています。

SqI-where/order

このようジョンポーネトに関する詳細に関しては SQLWHERE / ORDER コパーネトを参照してください。

値を条件付きで返す

(ノードおけばてはない)単一の値を条件付きで取得する必要がある場合、If-Else 条件を使用します。If-Else 条件は、ノードおけま 行をフィルターするけよ、適していません。Nodes/Rows をフィルター・エンポーネントとは異なり、If-Else 条件は、(文字列おけは整数 などの)単純型を返します。ですから、If-Else 条件は、単純型を条件付きで処理するシナリオのみに適しています。例えば、各月の平均 気温のリストがあると仮定します。

<temperat< th=""><th colspan="6"><temperatures></temperatures></th></temperat<>	<temperatures></temperatures>					
<data< th=""><th>temp="19.2"</th><th>month="2010-06" /></th></data<>	temp="19.2"	month="2010-06" />				
<data< th=""><th>temp="22.3"</th><th>month="2010-07" /></th></data<>	temp="22.3"	month="2010-07" />				
<data< th=""><th>temp="19.5"</th><th>month="2010-08" /></th></data<>	temp="19.5"	month="2010-08" />				
<data< th=""><th>temp="14.2"</th><th>month="2010-09" /></th></data<>	temp="14.2"	month="2010-09" />				
<data< th=""><th>temp="7.8" r</th><th>nonth="2010-10" /></th></data<>	temp="7.8" r	nonth="2010-10" />				
<data< th=""><th>temp="6.9" r</th><th>nonth="2010-11" /></th></data<>	temp="6.9" r	nonth="2010-11" />				
<data< th=""><th>temp="-1.0"</th><th>month="2010-12" /></th></data<>	temp="-1.0"	month="2010-12" />				
<th>tures></th> <th></th>	tures>					

If-Else 条件 により、リスト内の各アイテムのために気温が摂氏20度を超える場合は、値 "高" を、気温が摂氏5度以下の場合は、値 "低" を返すように設定することができます。

マピング上でIf-Else 条件の構造は、以下のよう」ないます:

<u>f/</u> if-else	
(Xool	
value-true	result[
value-false	

bool は接続された条件がtrue の場合、value-true は接続された result としての出力になり、条件がfalse の場合、valuefalse は接続された result としての出力にないます。 result のデータ型を事前に知ることはできず、これは値のデータ型、により異なり、 value-true おけよ value-false. は接続された値により異なります。 重要な点は、常に(文字列および整数などの)。単純型であるとい うことです。 入力値をノードおけば行などの複合型に接続することは If-Else 条件によりサポートされていません。

If-Else 条件は拡張することが可能です。これは、追加(◎)ボタンをクリックすることにより、コンポーネトに複数の条件を追加できることを 意味します。以前に追加された条件を削除するこは、ボタンを削除(◎)をクリックします。この機能により、条件がtrueの場合は、複数 の条件をチェックして、各条件のために異なる値を返すことができます。

	∬ ₈ if-else	
Ľ	>bool1 🛛	
ļ	value-true1	
Ľ	>bool2 🛛	
ļ	value-true2	result[:
Ľ	>bool3 🛛 🛛	
ļ	value-true3	
ļ	otherwise	

展開されたIf-Else 条件 は上から下へと評価されます(最初の条件が最初にチェックされ、2番目が次にチェックされます)。条件の全てが 真ではない場合に値を返す場合は、otherwise に接続してください。

手順を追ってのマピングのサンプルは、例値を条件付きで返すを参照してください。

4.10.1 例: ノードのフィルター

この例は、true/false条件を基はしてノードをフィルターする方法について説明しています。Filter: Nodes/Rows(🗟)コンポーネント かこの目的を達成するために使用されています。この例で説明されている技術は、XML のみではなく、CSV おけよテキストなどの他の コンポーネント型などとでも作業することができます。データベースの場合、フィルターを使用することはできますが、パフォーマンスのためには、代 われこSQLWHERE/ORDER コンポーネートを使用することが奨励されます(以下を参照してくたさい <u>SQLWHERE / ORDER コ</u> ンポーネト)。

この例で説明されているマピングは以下のいて検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\MarketingExpenses.mfd。



上に示されるように、このマピングは、経費の方向k ("ExpReport")のデータを含むノースXML からデータを読み取り、ターゲナ XML ("Marketing Expenses")に書き込みます。ターゲナとノースの間には複数の他のエンポーネントが存在します。最も関連性の高しエンポーネントは、このトピックの主題を表す expense-item フィルター()です。

このマピングのゴールは、マーケティング部署に属する経費アイテムのみをフィルターすることです。この目的を達成するオメリニ、フィルターコン ポーネト がマピングに追加されました。(フィルターを追加するコよ「挿入」メニューをクリックして、「Filter: Nodes/Rows」をクリック します)。

各経費がマーケティング部署に属するかを識別するオンタニ このマンピングは ソース内の "expto" 属性の値を確認します。この属性は経費 がマーケティングの経費である場合、値 "Marketing" を有します。 例えば、下にリストされるコード では、最初と3番目のアイテムは、マーケ ティングに属し、2番目のアイテムは開発部署に、4番目は販売部署に属することが示されています:

```
<expense-item type="Meal" expto="Marketing">
     <Date>2003-01-01</Date>
     <expense>122.11</expense>
  </expense-item>
  <expense-item type="Lodging" expto="Development">
     <Date>2003-01-02
     <expense>122.12</expense>
  </expense-item>
  <expense-item type="Lodging" expto="Marketing">
     <Date>2003-01-02</Date>
     <expense>299.45</expense>
  </expense-item>
  <expense-item type="Entertainment" expto="Sales">
     <Date>2003-01-02</Date>
      <expense>13.22</expense>
  </expense-item>
. . .
```

マピングが実行される前のXML 入力

マピングエアで、フィルターのnode/row入力が、ソースエンポーネント内のexpense-itemノードに接続されています。これにより、フィ ルターエンポーネントが処理されなければならないノードのノストを取得することを保証します。

フィルタレグが発生する条件を追加するコよ MapForce core ライブラカらequal 関数 を追加しました(詳細に関しては、以下を 参照してくたさい、<u>関数をマレングに追加する</u>)。equal 関数は、"type"属性を値 "Marketing" を持つ定数と比較します(定数を追 加するコよ 「挿入」 メニューをクトックして、定数をクトックします)。

条件を満たすアイテムのみをフィルターする必要があるため、フィルターのon-true 出力のみをターゲナコンポーネントに接続します。

マピングの結果をルビューする準備が整うと、「出力」 タブをクリックします。MapForce は、フィルターのbool 入力に接続されている条件である、それそれの経費アイテムノードを評価します。条件がtrue の場合、経費アイテムノードはターゲナトに ひされます。それ以外の場合は、無視されます。結果、条件を満たす経費アイテムのみが出力内に表示されます。

マッピングの実行後のXML 出力

4.10.2 例: 条件付で値を返す

この例では、true/false 条件を基はして、コンポーネトから単純型の値を返す方法を説明しています。「If-Else 条件」(in) かこの 目的を達成するかっかっ使用されています。If-Else 条件 とついターコンポーネントを混同していようご注意してください。「If-Else 条件」 は、(文字列、整数などの)単純な値を条件付きて処理する場合のかい適しています。ノードなどの複合値をついターする場合、代わりに ついターを使用してください(以下を参照してください) 例:ノードのフィルター)。

この例で説明されるマピングは以下の、次で検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\ClassifyTemperatures.mfd。



このマピングは、気温データを含むノースXML(「Temperatures」)からデータを読み取り、データを同じスキーマに準ずるターゲオ XML に書き込みます。ターゲオとノースの間には複数の他のエンポーネントが存在します。そのうちの1つが、このトピックの主題である。赤で) イライ トされている) if-else 条件です。

このマッピングのゴールは、ターゲット内の各気温の記録に短い説明を追加することです。具体的には、摂氏20度以上の気温の説明は、 "high"と表示され、摂氏5度以下の気温は、「low"と表示されるように設定します。上記以外の場合は、説明が表示されません。

この目的を達成するために、条件付き処理が必要とされます。ですから、If Else 条件 がマピングに追加されました。(If Else 条件を追加するコよ「挿入」メニューをクリックして、If Else 条件をクリックします)。このマピングでは、以下の2つの条件を受け入れるために(の ボタンを使用して)If Else 条件が展開されています: bool1 と bool2。

条件自身は、MapForce core ライブラから追加されたgreaterとless 関数によ追加されます。詳細に関しては、<u>関数をマピン</u> <u>
グロ追加する</u>を参照してくたさい。関数は、「upper」と「lower」と呼ばれる2つの入力エンポーネントより与えられた値を評価します、入力 エンポーネントを追加するコよ、「挿入」メニューをクリックして、「入力の挿入」をクリックします。入力エンポーネントに関する詳細にていて は、マンピングロン ラメーターを提供するを参照してくたさい

greater とless 関数は、true おけまfalse のどちらかを返します。関数の結果は、ターゲト・インスタンスに何か書き込まれるかを決定 します。具体的はよ、ソース内の"temp" 属性の値が、20 よれ大き、場合、定数値 "high" がif-else 条件に いこされます。ソース内の "temp" 属性の値が5 よれ小さ、場合、定数値 "low" は、if-else 条件に いこされます。 otherwise 入力が接続されていませんの で、上記の条件が満たされたい、場合は、result 出力コネクターにいてされる値はおりません。

最後に、result 出カコネクタは(各気温の記録に)度ずつ与えられる)この値をターゲル内の"desc" 属性に与えます。 マピングの結果をプビューする準備が整うと「出力」 タブをクリックします。 気温が20度は高い、 おけお度は低い場合、結果 XML 出力は、 "desc" 属性を含むことに注意してくたさい。

• • •

```
<data temp="-3.6" month="2006-01" desc="low"/>
<data temp="-0.7" month="2006-02" desc="low"/>
<data temp="7.5" month="2006-03"/>
<data temp="12.4" month="2006-04"/>
<data temp="16.2" month="2006-05"/>
<data temp="19" month="2006-06"/>
<data temp="22.7" month="2006-06"/>
<data temp="22.7" month="2006-07" desc="high"/>
<data temp="23.2" month="2006-08" desc="high"/>
...
```

マッピングの実行後のXML 出力

4.11 データのジョイン

(例えば、最初の構造内のフィールドA が2番目の構造内のフィールドBと同じ値を持つなど)の条件をベース 2つの構造からのデータを 結合する必要かある場合がおます。このようなマッピングの条件には、ジョイノエレポーネトを使用することができます。

ジョインエンポーネトは、カスタムに定義付けられた条件をベースして2つの構造をジョインすることを有効化するMapForce ユンポーネン 上です。条件を満たすアイテムのアノシエーション(ジョインされたセナ)を返します。ジョインは、(ID などの)共通のフィールドを共有する2つ の構造からのデータを結合する際にとても役に立ちます。

例えば、下のマンピングでは、中間コンポーネトは「ジョイン」コンポーネトです。このマンピングでは、2つのXML構造(個人のJストと 住所のJスト)がショインされます。このサンプルの目的は、各個人の情報をターゲト、XML ファイルに書き込むことです。。FirstName と LastName フィールドは、ジョインするキーとしての役割を果たします。すなわち、(Person の下の)FirstName とLastName の値が (Address の下の)FirstName とLastName と同じ値の場合、同じ個人に属する住所の詳細は「ジョイン」されます。ジョインした構造からのアイテムは、次のターゲト(この場合、XML ファイル)にマップすることができます。ジョインの条件は、ジョインコンポーネントプロパティ内で「ジョインの条件の定義」(『ショー)、ポマをクトックして定義されます。このサンプルコはマンピングのサンプルが存在し、例、XML構造のジョイン内で詳しく説明されています。



JoinPeopleInfo.mfd

上記のとおり、ソース構造とジョクノコンポーネントは、マンピングを優先順位の低い「全てコピー」 接続を使用して接続されています。 一般的には コンテキストが関連する場合、このような接続は自動的に MapForce により作成されます(詳細に関しては、次を参照してくたさい、全てコピー接続)。

(上のマピングのような)個別のエレポーネト、おけま同じエレポーネトトに属するエレポーネトから2つの構造をショインすることができます。 (例えば、XML構造とデータベーステーブルなどの)異なる型の構造をショインすることもできます。データベースに関連したショインの詳細に 関しては、次を参照してくたさい、データベースのデータのジョイン(SQL JOIN)。

ジョインコンポーネントを追加する方法:

1. マノピング変換言語をBUILTHNに設定します(これを行うココージッール、サイタンをクリックします、お子ば出力 | ビルトイン 実行エンジン] メニューコマンドを使用します)。 2. [挿入] メニューから[ジョイン] をクリックします。おけま[ジョイン] № ツール デーボタンをクリックします。ジョインエンポーネントが マッピングに表示されます。デフォルト では 200構造からデータを受け入れるけな、 200 nodes/rows 入力が存在します。必要 であれば入力の追加](□) ボタンをクリックして、ジョイノニ新し、入力を追加することができます。 3つ以上の構造をジョインするを 参照してくたさい。



- 3. ジョインコンポーネントのnodes/rowsアイテムどうへこれる構造を選択します。
- 4. ジョイン(おけま複数の条件)のナメの条件を追加してください。これを行うけま、ジョインロメポーネントを右クトックしプロンティを 選択します。ジョインの条件は、関数のブール値の結果をジョインエンポーネントのcondition アイテムは接続することにより、マッ ビングから直接追加することができます。データベースがジョインされる特定の場合、ジョインの条件は自動的にMapForce により 作成されます。詳細に関しては、ジョインの条件の追加を参照してください。

灹

- ジョインコンポーネントは、BUILT-IN 実行エンジン内でのみサポートされています。C#、C++、おけま Java 内でのコードの生成はサポートされていません。
- 構造がサポートされていない、おゴまジョインのナゴの入力ソースによりサポートされて以内場合、MapForceマピング上直接と ントを表示するか、おゴまマピングを検証する際に、メッセージウィンドウ内にといたを表示します。(マノビングの検証参を照してくだ さい)。
- ジョインコンポーネントをインラインユーザー定義関数の結果おけま入カノラメーターは接続したいでください。このような接続が存在すると、マッピングの検証中に検証エラーが発生します。
- 使用することのできるデータベースエンポーネント(テーブルオピューなどの)を直接ジョインエンポーネントに接続すると、SQL モード ^{SQL} ポタンカ自動的にジョインエンポーネントの右上に表示されます。有効化されると このパタノよ ジョイン操作に適用することのできる特別な SQL 機能を提供します(<u>SQL モード内のジョイン</u>を参照してくたさい)。
- ジョインされたアイテムの出力を他のジョインコンポーネントに接続することはできません。しかしなから、ジョインの結果の一部を他のジョインは接続することは可能です。

他のコンポーネント型と比較したジョインコンポーネント

ー部の場合、同じ結果を得るために、ジョインエパーネトの代わりに複合型変数、おけよフィルターを使用することができます。次を参照してください、変数の使用とフィルターと条件)。したしたがら、他のエレポーネント型とは異なり、ジョインエパーネントは、同じエンテキストに与えられるデーダを確認することができるため、マメピングのエジッグをわかり易くします。おこ、SQLモードがジョインエパーネント上で有効化されている場合、マメピングの、プォーマンスが大幅に向上します(これはデータベースジョインにも適用されます、次を参照してください、データベ <u>ーステーブルのジョイン</u>)。

親コンテキストの追加

特別な場合は、特定のマメピングの結果を達成するために、ジョインエンポーネントは接続されているデータのためていキからび親エンテキスト」 と呼ばれる、マメピングのエンテキストを明示的に提供することができます。親エンテキストを追加するには、ジョインエンポーネントのジョインされた アイテムをクリックし、コンテキストメニューから、親コンテキストの追加」を選択します。ジョインエンポーネントは、必要とするソースアイテムに接続することのできる追加 parent-context 入力を含む外観に変更します。詳細に関しては、以下を参照してくたさい。例親エンテキスト トの変更。

4.11.1 ジョインの条件の追加

条件に従し2つの構造のアイテムを組み合わせることによりジョイノは作動します。このため、ジョイノコは常に条件が必要にないます。下に示されるようにジョインの条件を追加するコは複数の方法があります。

メモ SQL モード内でデータベーステーブルがショインされると、MapForce は、テーブル間で検出されオット部キーレーションシップをベー スコして、ジョインの条件を自動的に作成します。自動的なジョインの条件が起こるコは、データベーステーブルがMapForce コン ポーネント上で親子関係内に存在する必要があります(すなオム、1つのテーブルが、親」まりまず 子」で、他のテーブルがセンポーネ ント上に存在する必要があります)。例 SQL モード内のテーブルのジョインを参照してくたさい。

アプローチ1: コンポーネント プロノティからジョインの条件を追加する

- マピング上で、2つ以上の構造(おは、データベーステーブル)がショインエンポーネントに接続されている必要があります。このサン プルで説明されているショインエンポーネントは、フォルダーくマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial、内で使用することのできる JoinPeopleInfo.mfd マッ ピングの一部です。詳細に関しては、例、XML構造のジョインを参照してくがさい。
- 2. ジョインエンポーネントで、「ジョインの条件の定義」(¹) ポタンをクリックします(おまよ エンポーネント の ッグーを右クリック し、エンテキスト メニューから プロハティ を選択します)。
- 3. 左の構造からアイテムを選択し、その他を右の構造から選択します(これは、このペアの比較がtrueを返すと、左と右の構造が ジョインされます)。

ジジョインの条件の定義			×
以下の条件と任意で関連する条件が true の場合ジョイン	が構成されま	ます:	条件の追加(A)
Person/FirstName AND () Person/LastName	=	 Address/FirstName Address/LastName 	X A
条件を定義するには、各構造からアイテムを選択して比喇	交します。		Ŧ
構造 1:		構造 2:	- 2 5
O FirstName O LastName		Address	
Control Environme Phone		C Address C FirstName C LastName C City C Street C Number C Number	
4	-	4	
✓ 注釈を表示する(A) ■型を表示する(T)			OK キャンセル

複数の条件を追加する必要がある場合、をクトックして、「条件の追加」新しいアイテムのペアを選択します。例えば、上のイメージは内では、2つのジョインの条件が定義されています:

- 1. 構造1内のFirstName は構造2内のFirstName と等価である必要かがます。
- 2. 構造1内のLastName は構造2内のLastName と等価である必要がおます。

ジョクの条件を削除するコよ横の「削除」 X ボタンをクルクします。

灹

- 複数のジョインの条件が存在する場合、2つの構造がショインされるためには、全ての条件が満たされている必要があります。すな わち、複数の条件は理論的な AND オペレーションによりジョインされていることを意味します。また、マノビングから追加された任意 の条件も含んています(次を参照してくたさい、下のアプローチ2)。
- 2つ以上の構造がショインエンポーネントに接続されている場合、これらの追加構造は、「構造 2」の下のドロップダウノストに表示されます。このような追加構造をドロップダウノストから選択すると、左側のペイノニ、ショインエンポーネント前の全ての構造が表示

されます。この方法で、任意の複数の構造間のジョインの条件を定義することができます。例に関しては、以下を参照してくたさい 例: CSV レポートを複数のテーブルから作成する。

各構造内のアイテムのデータ型をビューするけっかコよ「型の表示」チェックボックスを選択します。基トなるスキーマ(おけまデータ ベース)内に情報が存在することが前提で、「注釈の表示」オプションはアイテムの追加情報を表示します。両方のチェックボッ クスが選択されていると、レイアナトは、注釈と型の両方を表示するけっか「調整されます。例:

😢 Define Join Condition		×
Join structures when all of the following conditions and the op	tionally connected condition are true:	Add Condition
O Person/FirstName	= () Address/FirstName	X ^
AND Y Person/LastName To define a condition, select an item from each structure to co	= () Address/LastName	×
Person	∧ Address	✓ + +
type restriction of xs:anyType [1] ann. C FirstName type xs:string ann. C LastName type xs:string ann. C Email type xs:string ann. C Phone type xs:int ann.	 Address type restriction of xs:anyType [1∞] ann. FirstName type xs:string ann. LastName type xs:string ann. City type xs:string ann. Street type xs:string ann. Number type xs:string ann. 	^
< >	K	>
Show annotations	OK	Cancel

アプローチ2:マピングからジョインの条件を追加する

• マピング上で、ブール値を生成するコンポーネントを追加し、condition アイテムの入力にブールの出力を接続します。例え ば equal 関数は 一部のマピングアイテムの値を比較し、ジョインコンポーネントのcondition アイテムの入力として、ブー ル値の結果を提供します。



メモ ジョインエンポーネント プロ・ティ(アプローチ1)で条件が定義されていない場合、ジョインエンポーネントの condition アイテム か接続されている必要があります(アプローチ2)。

アプローチ3: 複合型のアプローチ

同じマピング内で、コンポーネト プロ・ティ(アプローチ1)内のジョインの条件を定義し、マピング(アプローチ2)からの1つと結合する ことができます。しかしながら、SQL モードを使用して、データベーステーブルをジョインする場合は、条件はアプローチ1 を使用して定義される 必要があります。 <u>SQL モード内のジョイ</u>を参照してください。

4.11.2 3つ以上の構造をジョインする

メニューコマイ「挿入 | ジョイン」を使用して、ジョインコンポーネントをマンピングに追加する場合、2つの構造をデフォルトで受け入れることができます(すなオカ、コンポーネントには2つのnodes/rows入力が含まれています)。

	🕅 join
	📮 🕪 joined
[ီ modes/rows
[nodes/rows
[* * condition

2つの構造をジョインする場合、入力を追加(回)ボタンをクリックし、必要な数だけ nodes/rowsを作成します。 nodes/rows入力を削除(図)ボタンをクリックします。 ジョイノオッケムとと2つの構造を必要とします、 ですから図ボタンは、2つの入力が存在する場合のみ使用することができます。


ジョイノニ複数の入力が存在する場合、ジョインの条件は、ジョインする各入力を考慮する必要がみます。以下を参照してくたさい、<u>ジョイン</u> <u>の条件を追加する</u>。複数のデータベーステーブルをジョインするサンプルのステップ・イステップの例に関しては、以下を参照してくたさい<u>例</u> CSV レポートを複数のテーブルから作成する。

4.11.3 例: XML 構造のジョイン

このサンプルは、ジョインコンポーネントを使用して、2つのXML構造からのデーダを結合する方法について説明しています。このサンプルは、 次の なしあるマメピングサンプルが存在します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\JoinPeopleInfo.mfd.

このマッピングの目的は、2つのソースXML ファイルから個人の情報(名、姓、住所、電子メール、と電話番号)を収集し、単一のター ゲオ XML ファイルに書き込むことです。

最初のXML ファイルは、各個人の名と姓、電子メールと電話番号を、下のサンプルロードで示されているとおり保管します(XML 宣言、 名前空間、とレコードの一部は簡素化のために省略されています):

```
<People>

            <firstName>Marquita</FirstName>
            <lastName>Bailey</lastName>
            <lastName>Bailey</lastName>
```

People.xml

2 番目のXML ファイルは、各個人の名と姓、および電子メールアドレスの詳細を保管します。:

```
</Address>
<Address>
<Address>
<FirstName>Totie</FirstName>
<LastName>Rea</LastName>
<City>Roseford</City>
<Street>Evergreen Lane</Street>
<Number>34</Number>
</Address>
</Addresse>
```

Addresses.xml

マピングの目的は、<Address> 情報を持つ最初のファイルからの<Person> 情報と2 番目のファイルからの情報を姓が一致するか比較します。特に、最初のファイル内の各 <Person> と2 番目のファイル内の各 <Address> で、 FirstName とLastName が比較されます。両方の値が同じの場合、対応する <Person> と<Address> レコードが同じ個人を参照している場合、ジョインが実行されます。ターゲナ XML 構造 は以下のよう ことはます。

```
<PeopleInfo>
   <Row>
      <FirstName>Marquita</FirstName>
      <LastName>Bailey</LastName>
      <City>Bridgedell</City>
      <Street>Olive Street</Street>
      <Number>4</Number>
      <Email>m.bailey@nanonull.com</Email>
      <Phone>555323698</Phone>
   </Row>
   < Row>
      <FirstName>Totie</FirstName>
      <LastName>Rea</LastName>
      <City>Roseford</City>
      <Street>Evergreen Lane</Street>
      <Number>34</Number>
      <Email>t.rea@nanonull.com</Email>
      <Phone>555598653</Phone>
   </Row>
</PeopleInfo>
```

PeopleInfo.xml

このマピングの目的は、ジョインエンポーネントをマピングに追加することで簡単に達成することができます。他のエンポーネント型を使用して、同じ結果を達成することもできることに注意してくたさい。しかしなから、下のステップでは、このサンプルの注題である、ジョインエンポーネントを使用してサンプルが説明されています。

必要とされるマンピングを作成するコよ、以下のステップに従います。

ステップ 1: ソース XML ファイルをマピングに追加する

- 1. 「挿入」メニューからXML スキーマ/ファイル をクリックします。次のソースファイルを参照します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\People.xml。
- 2. Addresses.xml(2番目のノースファイルのために上のステップを繰り返します。

ステップ 2: ターゲナ スキーマファイルをマピングに追加する

「挿入」メニューカらXML スキーマ/ファイルをクックします。<マイドキュメント
>\Altova\MapForce2021\MapForceExamples\Tutorial\PeopleInfo.xsd(ターゲナ XSD スキーマファイル)を
参照しますXML ファイルを与えるようにプロンプトされると、「スキップ」をクックします。ルート要素を選択するようにプロンプトされると、ルート要素として People Info を選択します。

ステップ 3: ジョインコンポーネントを追加する

1. 「挿入」メニューから「ジョイン」(おけよ「ジョイン」 🕺 ツール トーボタン をクリックします。この時点でマンピングは以下のよう しています (下のマンピングと同じ表示にするけよ、エンポーネントをドラッグしてサイズを調整する必要があります):



ジョインエンポーネントの構造に注目してくたさい、比較するこの構造につ場合、Person とAddress 構造)を接続するこつのnodes/rows アイテムが存在します。

2. ジョインレポーネトの最初のnodes/rows アイテムこPerson から接続を描きます。同様に2番目のnodes/rows ア イテムこAddressを接続します。



- 3. 既に説明されているとおり、FirstNameとLastName値が両方の構造内で等価の場合のみジョインは実行されます。この条件を定義するために、「ジョインの条件の定義」を引がたをクリックしてくたさい。
- 4. 最初のジョインの条件(構造1の下のFirstNameと構造2の下のFirstName)を定義するアイテムののでを選択します。
- 5. 「条件の追加」をクリックして、上記のステップをLastName のために繰り返します。

🕑 ジョインの条件の定義		
以下の条件と任意で関連する条件が true の場合ジョ	ョインが構成されます:	条件の追加(A)
AND Person/FirstName	= () Address/FirstName = () Address/LastName	X A
		Ŧ
条件を定義するには、各構造からアイテムを選択して、 構造 1:	比較します。 構造 2:	
	Address	▼
 () FirstName () LastName () Email () Phone 	Address FirstName LastName City Street Number	
4	•	-
 ✓ 注釈を表示する(A) □ 型を表示する(T) 		ОК キャンセル

マピングの一部では、1つの比較から構成される条件のみでジョインを実行することを可能な場合があります。しかしなから、このサンプルでは、2つの比較が作成されることが重要です:

- 1) 構造1 内のFirstName = 構造 2 内のFirstName。
- 2) 構造1 内のLastName = 構造 2 内のLast Name。

複数の条件が定義されている場合、ジョインが実行されるコは全ての条件が満たされている必要があります。このため、このサンプルでは、(期待される結果である)ジョインは両方の比較の条件が満たされている場合のみ実行されます。それ以外の場合、1つの比較の条件のみが定義されている場合、ジョインは同じ名と姓を持つ個人に対して実行される可能性があります。

ステップ 4: ジョインコンポーネントをターゲナ スキーマニマップする

2つの構造はジョインされ、ジョインした構造のどのアイテムがターゲトにマップされるかを定義することができます。これを行うには、ターゲトコンポーネントへジョインされた構造のアイテムからの接続を下に示されるように作成します。joined とRow 間の接続の目的は次のとおりです:ジョインの条件が満たされると、ターゲント内に新規のRow アイテムを作成する。



出力 ダブをクトックしてマピング出力をパビューすることができます。期待通り、各レコード(<Row>)には、2つの異なるノースからジョインされた住所の詳細が含まれています。

Value-Map の使用 4.12

Value-Map コンポーネントにより ルックアップを使用して値を他の値に置き換えることができます。 このようコンポーネントは1度に1つの値 を処理します。このためこマンピング上につの入力とつの結果が存在します。



Value-Map はアイテムを置き換えるために2つのセナト内の個別のアイテムをマップする場合役に立ちます。例えば、数値(1、2、3、4、5、 6 および7) で表示された曜日をそれぞれの曜日(「月曜」、「火曜」 など)と置き換えます。同様に月の名前(「1月」、「2月」、「3月」な ど)を各月の数値表記(1、2、3, etc). に置き換えます。マナングのランタイムでは、一致する値はカスタムルックアップテーブルン従い置 き換えられます。両方のセル内の値は異なる型であることができますが、各セルは同じデータ型の値を保管する必要があります。

Value-Map コンポーネトは最初のセナ内の各値が番目のセナ内の単一の値に対応する単純なルックアップに適しています。ルックアッ プテーブル内で値が見つからない場合、カスタム値、おけは空の値と置き換える、おけはそのままがなすることができます。更に複雑な条件をべ ースした値をルックアップおけよフィルターする場合、ンポーネトのフィルターを代わりて使用します。

重要な点は、コードを生成ませるMapForce Server 実行可能ファイルをマナピングからてレッイルする場合、ルックアップテーブルデータは 生成されたコード、ませまプァイル内に埋め込まれる点です。この結果、ルックアップテーブルを直接マッピング上で定義することは、データが頻 繁に変更されず、(例えば、数百エトリよりも少ないなど)データの総量が大きくない場合のみ良い選択でしょう。ルックアップデータか定期 的に変更する場合、マピング、および生成されたコードを定期的に管理することは難しく、ルックアップデータをテキスト、XML、データベース、 おけまExcel とて管理する方法のまうが簡単です。

ルックアップテーブルかきも、場合、ルックアップテーブルによりマッピングの実行が遅くなる場合がかます。この場合、代われに SQL-Where を使用するデータベースコンポーネントを使用することが奨励されます。ポータビリティを考慮すると、SQLite データベースは良い候補です。 他方、MapForce Server おけはMapForce Server Advanced Edition を使用してマピングを実行しルックアップテーブルの ウォ 一マノスを向上することもできます。

Value-Map の作成方法

Value-Map コンポーネントをマピングに追加するコは次の、ずれかの操作を実行してくたさい

- シールドードなをクリックします。 「Value-Map の挿入」
- 「挿入」メニューカら「Value-Map」をクルクします。
- 接続を右クルクし「Value-Map の挿入」をコンテキストメニューから選択します。

この操作は新規のValue-Map コンパーネントをマンングに追加します。ルックアップテーブルにアイテムのペアを追加することから開始しま す。これを行うコムエンポーネントのタイトルレーをダブルクリック、または右クリックし、「プロパティ」をコンテキストメニューから選択します。

🙁 V	alue-Map プロパティ	X
特定	の値を他に割り当てる Value-Map テーブル(V):	
冒	X 🖻 🛍	×
	input 🥖	result 🥖
	integer 💌	string 👤
	1	Sunday
	2	Monday
	3	Tuesday
	4	Wednesday
	5	Thursday
	6	Friday
	7	Saturday
*>	(新規エントリ)	
	✓ それ以外の場合	④ 值: incorrect date
		O オリジナルの入力値
		OK(O) キャンセル

マピングのランタイムで、MapForce は Value-Map の入力に各値が到達するかをチェックします。ルックアップテーブル内の 左側の列 に一致する値が存在する場合、元の入力値を右側の列からの値と置き換えます。それ以外の場合、任意で以下のオプションの一つに戻る ように構成することができます。

- 置換の値。下のサンプル内では、置換の値はデキスト「incorrect date」です。置換の値をテキストを入力せずに空に設定することしてきます。
- 元の入力の値。これはテーブル内で一致が存在したい場合、元の入力値がマンピングシンやされることを意味します。

それ以外の場合の条件を構成したい場合、Value-Map は一致か見つかられ、場合空のノードを返します。この場合、ターゲトコンポーネントには何も、なされず、出力には空のフィーリドが含まれます。このようは場合が発生することを回避するために、それ以外の場合の条件を構成、ませよ、substitute-missing関数を使用します。

空の置換値の設定と、それ以外の場合の条件を設定しない事は異なります。最初のケースは、出力でフィーリドは生成されますが、空の値 が存在します。後者のケースは値を囲むフィーリド(ますはXML 要素)が作成されません。詳細に関しては、<u>例 職位の置換</u>を参照してく たさい。

Value-Map の設定方法

ルックアップテーブル内で、必要な数の値のペアを定義することができます。値を手動で入力する、封まし、テキスト、CSV、封ましExcel ファイルからテーブルのデーダをコピーして貼り付けることができます。一般的なブラウザーを使用して HTML ページからテーブルをコピーして貼り 付けることができます。 <u>DB クエレペイン</u> 内のデータベースグリッドからデーダを張り付けることができます。 テキストファイルからデーダをコピーする とフィールドはダブ文字で区切られる必要があります。 更に、MapForce は多くの場合コンマ、おままたミコロンイことり区切られてい るテキスト を認識することができます。 ルックアップテーブルを作成する際に以下の点を考慮してくたさい

- 1. 左の列内のすべてのアイテムは一意である必要があります。それ以外の場合、具体的に一致するアイテムを決定することは不可能 です。
- 2. 同じ列に属するアイテムは同じデータ型である必要があります。 ルッグアップテーブル内の各列の上のドロップダウンリストからデータ 型を選択することができます。 ブーリ型を変換する必要がある場合、テキスト「true」 おさは「false」を文字通り入力してくたさい、このシナリオの説明は、例 曜日の置換を参照してくたさい。

MapForce がルックアップテーブル内で無効なデータを検知すると、ピンク色でッイライトされたエラーメッセージカ表示されます。例

Value-Map コンポーネントに外部ソースからデータをインポートする方法:

- 1. (例えば、Excelなどの)ソースプログラム内の注目するセルを選択します。これは単一のデータ列、ませおつの隣接した列である ことができます。
- 2. 外部プログラムの「コピー」コマンドを使用してクリップボードにデータをコピーします。
- 3. Value-Map エノポーネト上で、データを張り付ける行の前をクリックします。
- 4. Value-Map エンポーネント上の「クリップボードからテーブルを張り付ける」 🗟 ボタンをクトック、 おけば Ctrl+V」 おけま 「Shift+Insert」を押します。
- メモ 「クリップボードからテーブルを張り付ける」ボタノはノースからデータがユピーされている場合のみ有効化されています(すなわち、 クリップボード上にデータが存在します)。

クトップボードデータニ複数の列が含まれる場合、最初の2列の行のでーナーのみがレックアップテーブルに挿入されます。以降の行は無視されます。既存の値のトップの単一行からデータを張い打ける場合、コンテキストメニューが表示され、新規の行としてクトップボード上のデータが挿入されるか、ますは既存の行が上書きされるかか問われます。このナーダ、テーブル内の既存の値を上書きする必要がある場合、クトップボード には「お福栖ではなく1行のみのデータが含まれていることを確認してください。

既存の行の前に行を手動で挿入するコよ、最初に注目する行をクリックし、「挿入」 目 ポターをクリックします。

既存の行き削除するコよを側こマウスのおとを押したまま行をドラッグして(上ませよ下の新しい位置に移動します。

他の箇所で後に張り付けるためのことーまけまかい取りを行うには、行を選択し「コピー」 (1) ボタン(まけま「切り取り」 (1) ボタンをそれ それ)クリンクします。必ずしも連続していない、複数の行をことーまけまかい取ることもできます。複数の行を選択するには、行を選択しなから 「Ctrl」キーを押し続けます。切り取られた、まけまことーされたテキストは両方の列からの値を含むことに注意してくたさい。1つの行からの値 のみを切り取る、まけまことーすることとできません。

行を削除するコは「削除」 X ボタンをクルクします。

左と右の列をスワップするコな「スワップ」 一部、たちかりいたます。

Value-Map パラメーターの名前を変更する方法

デフォルトではValue-Map コンポーネントの入力パテメータガ入力」と呼ばれ、出力パテメーターガ結果」と呼ばれます。マメピングをお簡 潔にするオーダに、それそれの名前の横の「編集」 🖉 ポタンをクリックしてこれらの、テメーターの名前を任意で変更することができます。以下 は、カスタムパテメーター名を持つ Value-Map のサンプルです:

value-map P day-as-number day-as-text P

Value-Map をプレビューする方法

Value-Map の作成後、コンポーネントのタイトルバーにマンズをかざすことによりマンピングから直接実装をプレビューすることができます:



例:曜日の置換 4.12.1

このサンプルは整数の値を曜日の名前(1 = 日曜日、2 = 月曜日など)と置き換える Value-Map を説明しています。このサンプルは以 下の以で見つけることのできるマピングが存在します、 くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\Expense-valmap.mfd。



Expense-valmap.mfd

このマピングはノースファイル内のDate アイテムから曜日を抽出し、テキスト内で数値に変換し、ターゲトコンポーネトのWeekday ア イテムに書き込みます。具体的には、以下が行われます

- weekday 関数はノースファイル内の Date アイテムから曜日の数値を抽出します。この関数の結果は1から7の範囲の整数です。
- 最初のValue-Map コポーネトは整数を曜日(1 = 日曜日、2 = 月曜日 などに変換します。コポーネトは1-7 範囲外の無効な整数が発生するとテキスト「incorrect date」(無効な日付)を返します。

🙁 v	alue-Map プロパティ	×
特定 冒	の値を他に割り当てる Value-Map テーブル(V):	×
	input 🥖	result 🥖
	integer 💌	string 💌
	1	Sunday
	2	Monday
	3	Tuesday
	4	Wednesday
	5	Thursday
	6	Friday
	7	Saturday
*>	(新規エントリ)	
	✓ それ以外の場合	④ 值: incorrect date
		O オリジナルの入力値
		OK(0) キャンセル

 曜日ゴTuesday」が含まれている場合、テキスト「Prepare Financial Reports」はターゲトコポーネント内でNotes アイテムに書き込まれます。これはブール型 true おけはfalse 値を2番目のValue-Map コパーネントにつなする contains 関数を使用して達成することができます。2番目のValue-Map はは次の構成が存在します。

ν	/alue-Map プロパティ		
特定	の値を他に割り当てる Value-Map	テーブル (V) :	
冒	አ 🖻 🛱		×
	input	🥖 result	1
	boolean	▼ string	•
	true	Prepare financial repo	rts !
*>	(新規エントリ)		
	✓ それ以外の場合	◉ 值:	
		O オリジナルの入力値	

上で説明されている Value-Map は以下のように理解されるへきです:

 ブール値 true か発生すると、テキスト「--Prepare financial reports --!」に変換されます。それ以外の場合、テキスト 「-」か返されます。 最初の列のデータ型が「boolean」に設定されていることに注意してください。これは入力ブールの植 true か認識されていることを保証しています。

4.12.2 例:職位の置換

このサンプルはValue-Map コンポーネントを使用して XML ファイル内の特定の要素の値を置き換える方法(すなオカ定義済みの ルックアップテーブルを使用する方法)を示しています。

このサンプルで必要とされる XML ファイルは以下の 次で見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\MFCompany.xml。他のデータ以外に、企業の従業員と職位 などの清報を保管します。例:

```
<Person>
  <First>Michelle</First>
  <Last>Butler</Last>
  <Title>Software Engineer</Title>
</Person>
  <First>Lui</First>
   <Last>King</Last>
   <Title>Support Engineer</Title>
</Person>
  <First>Steve</First>
   <Last>Meier</Last>
   <Title>Office Manager</Title>
</Person>
</Person>
```

XML ファイル内の職位の一部を置き換える必要があると仮定します。具体的には職位「Software Engineer」は「Code Magician」と置き換えられます。また、職位「Support Engineer」は「Support Magician」と置き換えられます。その他の職位は すべて変更されません。

この目的を達成するために、「XML スキーマノファイルの挿入」 で、ツール デーポターをクリック、おけば「挿入 | XML スキーマノファイル」 メニューコマンドを実行してマッピングエレアイファイルを追加します。 次に、マッピング上のXML コンポーネトをエピーして貼り付け、以下

に示される通りは接続を作成します。自動的に不必要な接続の作成を防くために目「子要素の自動的な接続の切り替え」ツール バーオプノョンを最初にオオに切り替える必要がある場合があります。



First、Last、およびTitle 要素に変更を加えることなく作成されたマンピングはPerson 要素をターゲオファイルにピーします。

必要とされる職位を置き換えるために、Value-Map コンポーネントを追加します。2つのTitle 要素間の接続を右クルクし、コンテキスト メニューから「Value-Map の選択」を選択します。Value-Map プロノティを下に表示される通りセットアップします。

Υ Va	alue-Map プロパティ		Х
特定() 1日	D値を他に割り当てる Value-Map テーブル(V):	X	Ī
	λb	▲果 ノ	
	string 📃 💌	string 🔹 💌	1
	Support Engineer	Support Magician	
*>	(新規エントリ)		
	□それ以外の場合	④ 值:	
		◎ オリジナルの入力値	

上記のセオアップごは、「Software Engineer」は「Code Magician」、と置き換えらえます。そして、「Support Engineer」は 「Support Magician」と置き換えらえます。それ以外の場合条件はお指定されていません。このため、Value-Map は職位が 「Software Engineer」と「Support Engineer」、以外の場合は空のノードを返します。この結果、出力ダをクトックし、マノピング をプレニーすると、Person 要素の一部はよ不足している Title が存在します。例

<person></person>
<pre><first>Vernon</first></pre>
<last>Callaby</last>
<person></person>
<first>Frank</first>
<last>Further</last>

```
<Person>

<First>Michelle</First>

<Last>Butler</Last>

<Title>Code Magician</Title>

</Person>
```

上記の通り、空のノードは生成される出力内で不足するエトリを引き起こします。このため、上記のXML フラグメト内では、ルックアップ テーブル内で職位が存在するためMichelle Butler の職位のみが置き換えられています。ここまでご作成されている構成はオリジナルの必要条件を満たしていません。正しいセントアップは以下の通りです:

シ Value-Map プロパラ		;
特定の値を他に割り当て	> Value-Map テーブル(V):	
E X 🖻 🛱		×
እታ	_∕ 結果	1
string	✓ string	▼.
Software Engine	Code Magician	
Support Engine	Support Magician	
*> (新規エントリ)		
✓ それ以外の場合	〇 値:	
	● オリジナルの入力値	

上記の構成を使用すると、マンピングのランタイムコ以下が発生します

- 「Software Engineer」は「Code Magician」と置き換えらえます。
- 「Support Engineer」は「Support Magician」と置き換えらえます。
- ルックアップテーブルでオドジナルのタイトルが見つからな場合、Value-Map は変換しないままタイトルを返します。

説明のためこ、「Software Engineer」と「Support Engineer」以外の職位を「N/A」などのカスタム値に変えることができます。これを達成するコよ、Value-Map プロノティを以下に示される通り設定します。

۷ 🕲	/alue-Map プロパティ			>
特定	の値を他に割り当てる Value-Ma	p テーブル <mark>(V)</mark> :		
	Х 🗈 💼			\times
	入力		結果	1
	string	_	string	•
	Software Engineer		Code Magician	
	Support Engineer		Support Magician	
*>	(新規エントリ)			
	✓ それ以外の場合		 ● 值: N/A 	
			O オリジナルの入力値	

マピングをプレビューする場合、各職位が出力に存在しますが、一致しない項目には「N/A」値が与えられます。例

<person></person>		
<first>Vernon</first>		

```
<Last>Callaby</Last>
<Title>N/A</Title>
</Person>
<Person>
<First>Frank</First>
<Last>Further</Last>
<Title>N/A</Title>
</Person>
<First>Michelle</First>
<Last>Butler</Last>
<Title>Code Magician</Title>
</Person>
```

Value-Map サンプルは終了です。上記のロジックを適用することにより、他のマメピング内で希望する結果を得ることができます。

4.13 例外の追加

例外は、フィルターによ版された条件が発生すると、マピングの処理を停止し、エラーを表示する特別なエレポーネトの型です。 true/false 条件をチェックするフィルターをマピングが含む場合、例外を追加することができます(次を参照: フィルターと条件)。 スタム化されたしきい値よりマピングアイテムの一部の値が大き、場合、例外をスローするなどが挙げられます。

例外をマッピングに追加する

- 1. 「挿入」メニューから「例外」をクルクします。
- 2. 「例外の挿入」(1)、ハールトーボタンをクリックします。
- 3. 例外 throw 入力をフィルターのon-true おけよon-false 出力に接続します。
- 4. この例外がマローされると、任意で、例外のerror-text入力を、エラーのテキストを与える他の通常、定数であるコンポーネントに接続します。
- メモ フィルターのon-true とon-false 出力が双方接続されている必要があります。具体的には、これら出力の内のしつが、例外に (中間関数おとは、コンポーネントを使用することなく)直接接続されていないけいばよりません。

マピングロ例外が発生すると、以下のような通知が送信されます

• MapForce 内でメルセージウイドウはエラーと別外テキストを表示します。 "Expense limit exceeded")。



マピング言語がXSLT 2.0 おけはXQuery の場合、「実行に失敗しました」 エラーがシャレージケイドウ内で表示され、対応するXSLT2 おけはXQuery タガ開かれます。エラーの生じたライノはシャセージケイドウ内で、イライトされます。

 MapForce Server からマピングを実行する場合は、MapForce 内で定義されたカスタムの例外テキストに続きエラー「例 外カシローされました」が返されます。

c:\codegen\mfx\ExpenseLimit>"C:\Program Files (x86)\Altova\MapForceServer2016\bi n\MapForceServer.exe" run ExpenseLimit.mfx Exception was thrown! Expense limit exceeded!

 生成されたC#、C++、おさまJavaコードからマピングを実行すると、エラー「USER EXCEPTION」が、内で定義された カスタム例外テキストの前に返されます。

C:\codegen\cs\ExpenseLimit\Mapping\bin\Debug>Mapping.exe Mapping Application USER EXCEPTION: Expense limit exceeded!

4.13.1 例: "より大きい" 条件の例外

この例では、「よりも大きい」条件が発生すると例外を発生するマンピングの作成方法を説明します。このマンピングのオメのサンプルマンピング は以下にすります: <マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥ExpenseLimit.mfd。



このマピングは、ソースXML インスタンス内のexpense アイテムが200 以上の値を有すると、例外をスローします。値 "200" は、定数 により与えられます。1ess 関数 は、2つの値を比較するためご使用されます。 expense の値が200 より小さ、場合、その親である expense-item をフィルターン マされ、例外はスローされません。それ以外の場合、カスタムテキスト "Expense limit exceed" と共 に、例外がマローされます。

上に示されるように、例外は、 🤍 アイコンにお識別され、 次の2つのアイテムは構成されます: throw とerror-text。 throw アイテムは、 フィルターの on-false おっぱ on-true 出力に接続されている必要があります。 error-text は、 例外のカスタムテキストを与える定数に接続されています。

重要な点は、フィルターの両方の出力は接続されています。それ以外の場合、例外は、スローされません。この特別な例では、on-false 出力は、例外に接続されていますが、on-true 出力は、ターゲナコンポーネントに接続されています。

4.13.2 例: ノードが存在しない場合の例外

この例では、ソースXMLスキーマが存在しない場合例外をスローする方法を説明します。説明を簡素化するために、この例は、同じXML スキーマをソースとターゲトコンポーネトで使用します。

ソーススキーマをマッピングに追加する

- 1. 「挿入」メニューカら「XML スキーマ/ファイル」をクリックし、<マイドキュメント >\Altova\MapForce2021\MapForceExamples\BookList.xsd を参照します。
- 2. インスタンスファイルを提供するようにプロンプトされると、「スキップ」をクリックします。
- 3. スキーマルト要素を選択するようにプロンプトされると、BookListをルト要素として選択します。

ターゲットスキーマを追加するコま、同じステップに従います。そして、対応するコマンド(ませま、対応するソールレーボタン)を「挿入」メニュ 一から使用し、以下を追加します:

- Filter: Nodes/Rows コパーネト (以下も参照してくたさい: フィルターと条件)
- テキスト "No year defined!" を伴う定数
- 例外

最後に、ライブラリウイボウから、exists 関数をマピングエリアイボラッグして、下に表示されるように接続を描きます。



XML スキーマニ従うと、書籍の名前以外のBook 要素の全ての属性は、任意です。ですから、"Year" 要素は、有効な XML インスタ ンス内に存在する、もしくは、しない場合があります。このマメピングのゴールは、各書籍のために "Year" 要素が存在する箇所で XML イン スタンスを正確に処理することです。それ以外の場合、、マメピングは、例外をスローします。

マッピングの例外の成功をテストする

- 1. ソースエンポーネントのヘッダーをダブルクトックします。「入力 XML ファイル」の横から、次のファイルを参照します: <マイドキュ メント> ¥Altova WapForce2021 WapForceExamples BookList.xml。
- 2. 「出力」ボタンをクリックしてマッピングを実行します。

例外をテストする

- 1. 同じデルケリ内に、BookListInvalid.xmlと呼ばれるBookList.xml ファイルのコピーを作成します。
- 2. Book 要素から"Year" 要素を削除するようこ修正します。
- 3. ソースコンポーネントのヘッダーをダブルクトックします。「入力 XML ファイル」の横から、BookListInvalid.xml ファイルを参照します。
- 4. 「出力」ボタンをクリックしてマッピングを実行します。

マパングの作動のしくみい注目します。

接続Aは、ターゲオーインスタンスがソースーインスタンス内の各書籍のために作成されていることを保証します。接続B、C、D、Eは "Title"、"Year"、"Price"、と"Author" が各書籍のためにノースからターゲットにこと一されルにとを保証します。

接続Fはexists関数が"Year"属性の存在をチェックするようなトリガーします。接続Gは関数の結果(true おけまfalse)を フィルターコンプします。結果がtrueの場合、"Year"属性は存在し、書籍はフィルターコンプされ、フィルターご書籍はいでされ、後に、接続Hを介してターゲットについてされます。

ソースコンポーネントのYear 出力にフィルターが直接接続されていないことに注意してください。これを行うた場合、フィルターは自身の存在によりYear をフィルターするので、意味が無く、例外は決してスローされません。

ルールゴ従い、フィルターのon-false おけなon-true 出力へ例外が接続されている必要があるため、Iか存在します。

最後に、接続 K かりスタムエラーテキストを定数から例外コンポーネントに マレます。

4.14 文字列の解析とシリアル化

文字列の解析とメアル化は、文字列からデーダを解析、おオオデーダを文字列にメアル化するエレポーネントを構成する高度なマピング 技術です。この技術は、ファイルからデーダを読み込む、おオオファイルにデーダを書き込む)代替として使用することができます。文字列を解 析する、おオオデーダを文字列にシノアル化するMapForce エレポーネントは、以下のようなシナレオにとても役しさきます。例

- XMLなどの構造をデータベースフィールドに挿入する必要がある場合。
- データベースフィールド内に保管されたXML プラグメントをスタンドアロンのXML ファイルニ変換する必要がある場合。
- レガシーデータをテキスト(例)単一のデータベースフィールド内の固定値コンテンツとして保存しており、このデータを並べ替え可能 なフィールドベースの構造にするために変換する場合。

文字列の解析とノアル化以下のMapForce コノポーネト型で使用することができます:

- テキスト (CSV、固定値フィールド テキスト)
- XML スキーマファイル

上記のエレポーネト型のために、文字列の解析とノアル化はBUILTHN ターゲト言語でサポートされています。更に、JSON への文字列の解析、おけは文字列からのJSON シリアル化はBUILTHN、C#、おけは Java でサポートされています。

4.14.1 コンポーネントの解析/シリアル化について

MapForce内のコンポーネントの解析/シリアル化コンポーネントはソースもしくはターゲトコンポーネントではなし、ハイブリンドのコンポーネントです。マンピングデザイン内での役割を考えると、このようニコンポーネントはノースとターゲトコンポーネントの間に置かれることが適切です。

文字列の解析ためのコンポーネントの「文字列の解析/シノアル化」のためコよ、構造を持つ文字列を他のフォーマトに変換する必要かあ ります(例:データベース内に文字列として保管されているXML)。ソース文字列からデータを解析するコよ、コンポーネントの「解析/シノアル 化"は、ソース文字列がMapForce構造に変換されることを意味します。ですから、文字列として保管されたノースXMLの要素お台北属 性へのアクセスを取得することができます。

ソース(文字列構造)	 コンポーネントが文字列を解析		
	文字列		
	構造	[ターゲット(全ての構造)
		┺┤	
		1	
		14	

ジェネリックな "文字列の解析" ユンポーネント

上の図は、MapForce エノポーネトカ文字列を解析する全般的な構造を示しています。エノポーネトの "文字列の解析/シリアル化" は、マノピングのノースとターゲトの間に配置されています。このエレポーネトは、文字列の構造の一部をトップ文字列ノードに接続されてい る単一のMapForce コネクタを入力とて受け入れません。出力構造は、MapForce によりサポートされるデータターゲットとなることができます。

コンポーネトから文字列にデータをシリアル化すると、逆の効果が発生します。特に、MapForce コンポーネトの全体の構造が、必要であれば更に操作することが可能な文字列構造にないます。例:これによりXML ファイル(おけはXML ファグメント)から、データベースフィールドおけはExcel スプレメジントの単一のセルに書き込むことが可能にないます。



ジェネリックな "文字列ニシリアル化" コンポーネント

上の図は、MapForceでのジェネトックなコンポーネトの文字列ミンアル化"を示しています。このコンポーネトは、標準 MapForce コネクタを使用して、MapForceによりサポートされるデータノースを入力として受け入れます。出力構造は、コンポーネトのトップ文字列 からターゲトコンポーネトアイテムコ描かれた単一のMapForceコネクタを使用して、更に、なすることのできる文字列、例、スプレイドシー トセル)です。例、文字列にシリアル化(XML からデータベースへ)を参照してくたさい。

文字列の解析ませまシリアル化のためにことポーネートをマピングウイドウからいつでも指定することができます。これを行うには、ルートノード横の「ファイル/文字列」(File/String))がシをクリックして、希望するオプションを選択します。



コンポーネントモードの変更

メモ 「文字列の解析/シリアル化"コンポーネントは、文字列からデータを読み込み、同時に文字列に書き込むことはできません。ですから、ルートノードは入力コネクタ、まとは出力コネクタの、両方ではなくどちらかを持つことができます。両方のオペレーションに同じコンポーネントを使用しようとすると、エラーが生成されます。

コポーネントを文字列の解析ませよシノアル化に指定する場合、コポーネントの外観の変更は以下の通りです

- コンポーネトはタイトル内に「解析」ませばシリアル化」のプレフィックスを与えられます。
- タイトル・一は関数コンポーネントに類似した黄色の背景色が与えられます。

- トップのノードは文字列: プレフィックスから始まり、 2 アイコンで識別されます。
- コンポーネトが文字列を解析する場合、ルートノードからの出力コネクタは有効ではなく、使用することはできません。
- コンポーネトが文字列にシリアル化される場合、ルートノードへの入力コネクタは、有効ではなく使用することはできません。

コンポーネトが「文字列の解析/シリアル化」モード内にある場合、ファイルベースのモードと同様の法要で設定を変更することができます(ヨ ンポーネトト設定を変更するを参照)。コンポーネトが「解析」ませま「シリアル化」モードの場合、全てのコンポーネント設定を使用できない場合があります。

4.14.2 例: 文字列にシリアル化 (XML からデータベースへ)

このサンプルは、デーダを文字列にシリアル化するマメピングデザインの作成に必要なステップを説明します。このサンプルコまサンプルファイルが 存在します。このサンプルを開始する前に、サンプルファイルを確認するコよ、ファイルを以下の「なで検索することができます: くマイドキュメン ト>¥Altova¥MapForce2021¥MapForceExamples¥Tutorial¥SerializeToString.mfd。

複数の<Person> 要素には構成される XML ファイル(関連したスキーマ) かあると仮定します。各 <Person> 要素は、それぞれの 姓、名、役職、内線番号、電子メールアドレスを表します:

```
<Person>

<First>Joe</First>

<Last>Firstbread</Last>

<Title>Marketing Manager Europe</Title>

<PhoneExt>621</PhoneExt>

<Email>j.firstbread@nanonull.com</Email>

</Person>
```

目的は XML ファイルから各 <Person> 要素を抽出し、リテラルに(XML タグも含め) SQLite データベースのPEOPLE 新規データ ベースの記録として挿入することです。PEOPLE テーブルは 2 つの列のみを含んでします: ID とPERSON。 定義は以下の通りです:

CREATE TABLE PEOPLE (ID INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, PERSON TEXT);

マピングが実行された後、PEOPLE テーブルは、XML ファイル内のXPerson> 要素の行数と同じ数の行を持つようになります。

このゴールを達成するコよ、以下を行います

1. マメピングエアソースXML コンポーネントに(「挿入 | XML スキーマ/ファイル」メニューコマンドを使用して)追加します。サン プルファイルは以下で検索することが出来ます。 くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\MFCompany.xml。

- 2. (コピー貼いけ)を使用して) XML コパーネトを複製します。
- 3. 複製された XML コポーネントで、File/String をクリックして、「XML を文字列にシリアル化する」を選択します。

■ MFCompany □□ファイル: MFCompany.xml ファイル/文字列	ļ	MFCompany D7イル: MFCompany.xml	771	(ル/文字列)
Company Root element		Company Root element	~	コンポーネント設定からファイル名を使用する(F)
a ⊕ O Person		OPerson		マッピングにより与えられた動的なファイル名を使用する(D)
				XML に文字列を解析する(P)
				文字列にXMLをシリアル化する(S)

4. コンテキストメニューから、複製されたコンポーネトを右クルクして、「ルート要素の変更」を選択します。ルート要素を 〈Person〉 「変更します。

🕑 ルート要素の選択	—
名前 Company Person	名前空間 URI http://my-company.com/namesp http://my-company.com/namesp
	- - - - - - - - - - - - - -

全般的に、ルート要素をXML スキーマ内で(ローカルではない)グロー・ い宣言を持つ要素に変更することができます。スキーマ内でグロー・ いして定義されていない要素は ベルート要素の選択 グダイアログボックス こしえいされません。

5. コンポーネントをダブルクトックし、「XML 宣言の書き込み」チェックボックスをクリアします。これにより、XML 宣言が答 <Person> 要素のために書き込まれることを回避します。

· · · · · · · · · · · · · · · · · · ·
スキーマ ファイル(F) MFCompany.xsd 参照(W) 編集(T)
ターゲット 名前空間のプレフィックス(P):
☑ スキーマ/DTD参照を追加(スキーマの絶対パスを使用する場合はフィールドを空に)(A):
 ■ XML 宣言の書き込み(X) ✓ 値をターゲットの型にキャスト (無効な出力を書き込むリスクがある時に数値またはデータ値のフォーマッティングの保持を無効化)(G) ✓ 整形して出力(Y)
✓ 全ファイルパスを MFD ファイルへの相対パスで保存(S) OK キャンセル //

6. マピングエレアニターゲト SQLite データベースエンポーネントを以下の やから追加します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\\dbserialize.db。(「挿入 | データベース」メニューコ マドを使用して データベースエンポーネントを挿入します。 データベースへ接続する も参照)。データベースオブジェクトを挿入する ようにプロンプトされると、PEOPLE テーブルを選択します。

ि データベース オブジェクトを挿入	×
テーブル、ビュー、ストアドプロシージャを選択するか、独自の SOL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SOL SELECT ステートメント はローカルの mfd ファイルに保存されます。	
▼ ▼ ◆ SELECT ステートメントの追加/編集 リレーションの追加/編集 (0) レコードセット構造の追加/編集	
■ 図 ■ deserialize 日 図 ● main 日 図 ● テーブル 日 図 ● ユーザーテーブル − 図 団 PEOPLE	
_ プレビューを表示(P) >> OK	キャンセル

7. コンポーネントを下て表示されるように、リンクします。マンピングの左側では、<Person>要素は、コンポーネントはシノアルビコン ポーネントにマップされます。マンピングの右側では、PEOPLEデーダベーステーブルのシノアルビされた文字列の値がPERSON列 に挿入されます。最後に、<Person>からPEOPLEテーブルン描かれたコネクタは、MapForceに当てはまる各 <Person>要素のよりに新し、記録を作成するように命令します。



- 8. データベースエレポーボトのA: In ボタンをクリックし、 MapForce にマッピング変換が実行される都度、以下のアクションを行うように命令します:
 - a. 「全てのレコードを削除」オプションを選択します。マッピングランタイムでは、これは既存のレコードを新規のたのか挿入される 前にデータベースから削除します。
 - b. ID 列の横の「DB-生成された」オプシンを選択します。これはレコードのID がデータベースにお生成されることを確保 します。「DB-生成された」オプションは列がこのオプションをサポートする場合のみ表示されます。識別、ませま自動インクリ メトフィールドではない場合、max+1 オプションを代われて使用することができます。このオプションは、その列内で既存の最 高値が可であるかを確認し、次に使用することのできる1 つずつ増やされた整数を入力します。

ビデータベーステーブルのアクション - PEOPLE							
最初のレコードの前に実行する SQL ステートメント ② なし(N) ③ 全てのレコードを削除(L) ② 全ての子テーブルにある全てのレコードも削除 ③ カスタム SQL(C):							
各レコードに対して実行するアクション 以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て真であれば、指定されたアクションが実行されます。							
入力データ アクション 全て挿入 アクションを追加(A) ID max() + 1 アクションを追加(A) ID マップされた値 アクションを挿入(I)							
 ✓ ✓ アクションを削除(D) ■トランザクションを使用(T) 							
○ バルク転送を使用 (MapForce Server のみ)(B) バッチ サイズ: 1000 レコード バルク転送はデータベース接続によりサポートされていません。							
OK キャンセル							

ID 列のために選択されて、サニmax()+1 アクショント注意してくたさい。これによりデータベース内の既存のID 値の最大値 をMapForce に分析するように命令します。

データを文字列にシリアル化するマンピングデザインが作成されました。「出力」タブをクリックすると、プレビューSQL クエリは、XML ファイル内の各 <Person> 要素 ためご個別の記録がデータベース 挿入されることを示します。

4.15 ノード名のマッピング

MapForce を使用して、マピングを作成すると、多くの目的はノースから値を読み取り、ターゲメトロ値を書き込むことです。しかしなから、ソ ースからだけではなく、ノード名から値を取得することを希望する場合がおります。例えば、ソースXML から値ではなく、要素おけは属性の 名前を読み取り、ターゲメ XML内で(名前では無く)要素おけは属性の値に変換することを希望する場合かおります。

以下の例を考慮してくたさい、XML ファイルが製品のリストを含む場合、各製品は次のフォーマナを有します:

```
<product>
    <id>l</id>
    <color>red</color>
    <size>10</size>
</product>
```

目的は、各製品の情報を名前と直のペアニ変換することです。例:

```
<product>
    <attribute name="id" value="1" />
    <attribute name="color" value="red" />
    <attribute name="size" value="10" />
</product>
```

このようなシナリオコよ、マピングからのノード名へのアクセスが必要にないます。このトピックの主題であるノード名への動的なアクセスにより、このようなデータ変換を行うことができます。

メモ node-name とstatic-node-name コアライブラリ 関数を使用して、上記の変換を行うことができます。しかしなから、この 場合は、ソースから期待する正確な要素名が必要であり、ターゲルに手動で要素を接続する必要があります。また、これらの関 数は、十分ではない、場合かあります。例えば、名前別につれいター、ませるグループ分けを行う場合、また、ノードマノピングからデー タ型を操作する場合などか挙げられます。

ノード名への動的なアクセスは、ノード名を読み取る場合だけではよく、書き込む必要がある場合にも使用することができます。標準マンピングでは、基となるエンポーネントのスキーマからくる、ターゲット内の属性おけは要素の名前は、マンピングの実行前に既知ではよい、新規の属性おけは要素を作成することができます。具体的には、属性おけは要素の名前が、MapForceによりサポートされるソースからマンピング自身により与えられます。

ノードの子要素、おけば属性への動的なアクセスを可能にするには、ノードが実際に子要素おけば属性を持つ必要があり、XML ルートノードであってにないません。

動的なノード名は、次のコンポーネント型に、おけま型からマパングをする際にサポートされています。

- XML
- CSV/FLF*

* MapForce Professional おはEnterprise Edition が必要です。

メモ CSV/FLF の場合、CSV/FLF 構造には、ノードが存在しないため、動的なアクセスは、ノードの代わりにフィールドへのアクセ スを意味します。 マピングのターゲトがCSV おさまFLF(固定長フィールド)ファイルである場合、フィールドは、コンポーネト設定内で定義されて いる必要があります(名前、順序、おさなーゲトフィールドの数量を変更することはできません)。 XML とは異なりテキストファイルの フォーマトは固定されており、実際のフィールドの値のみを操作することができますが、フィールド名、数量、順序を操作することはできま せん。

ノード名への動的なアクセスは次のマンピング言語によりサポートされています: BUILT-IN、XSLT 2.0、XSLT 3.0、XQuery、C#、C++、Java。*

*これらの言語はMapForce Professional おけまEnterprise Edition を必要とます。

ノードへの動的なアクセスに関する情報は、以下を参照してくたさい、ノード名へのアクセスを取得する。順序を追ったマッピングの例に関しては、例、要素名を属性値にマップするを参照してくたさい。

4.15.1 ノード名へのアクセスを取得する

XML コンポーネント(おけよ CSV/FLF コンポーネント内のフィールド)内のノードに子ノードが存在する場合、マンピングの名前と各子ノー ドの値をマンピングで直接取得することができます。このテクニックは、「動的なノード名」へのアクセスと呼ばれます。「動的」 とは、処理がランタ イム中に素早く行われることを意味し、マンピングが実行される前に既知の静的なスキーマ情報をベースに行われます。このトピックは、ノード 名への動的なアクセスの有効化の方法と、その使用方法について説明します。

ソースからデータを読み取る場合、「動的なノード名」とは以下を行うことができることを意味します

- ノードの全ての子ノード(おけま属性)のリストをシーケンスとして取得します。MapForceでは、 ベシーケンズ は、 ゼロのリスト、またはおーゲトに接続することのできるアイテムであり、 ソース内にアイテムが存在すると、 ターゲトに同じ数のアイテムを作成します。ですから、 例えば、 ノードがノース内に5つの属性を持つ場合、 ターゲト内に属性に対応する5つの新規の要素を作成します。
- (標準マピングが行う)子ノードの値を読み取るだけではなく、名前も読み込みます。

ターゲトにデーダを書き込む場合、「動的なノード名」は、以下を行うことができることを意味します。

コンポーネント設定(しわかる「静的」名前)におり与えられた名前とは対照的な、マンピング(しわかる「動的」名前)により与えられる名前を使用して、新ししレードを作成します。

動的なノード名を説明するために、このトピックは次のXMLスキーマを使用しています:〈マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Products.xsd。このスキーマは、サンプルインスタンスドキュメト Products.xmlを伴っています。スキーマとインスタンスファイルをマンピングエノアに追加するには、「挿入| XML スキーマ/ファイル」メニュ ーコマンドを選択して、〈マイドキュメント〉>\Altova \MapForce2021\MapForceExamples \Tutorial\Products.xmlを参照 してくたさい、ルート要素を選択するように問われると、productsを選択します。

product ノードのための動的なノード名を有効化するコム、右クトックし、次のコンテキストメニューコマンドの「つを選択します:

- ノードの属性にアクセスする場合、「動的な名前を持つ属性を表示」
- ノードの子要素にアクセスする場合、「動的な名前を持つ子要素を表示」



図.1 (子要素のための)動的なノード名の有効化

メモ 上のコマドは、子ノードを持つノードのみに対して使用することができます。また、コマドはフォノードには使用することができません。

ノードを動的なモードに切り替えると、下に表示されるダイアログボックスに類似したダイアログボックスが表示されます。このトピックの目的のために、オプションを下に表示されるように設定します。これらのオプションは<u>特定の型のノードにアクセスする</u>内で説明されています。

🕑 動的に名前の付けられた子の設定	×						
これらの設定を使用してジェネリックな構造を使用する子要素全体を構成します。							
動的に名前の付けられたアイテムのコンテンツにアクセスするための型を選択してください:							
 □ □ □ □ R xs:anyType ✓ □ □ R text() 							
-							
ここで型を選択すると、構造がマッピングためだけに使用できるようになります。 ランタイムでは実際の型はチェックされません。							
■ノード名を固定し、要素をフィルター、または、作成するためにテストノードの名前を表示し	.ます。						
▼ 親要素のスキーマ子要素を表示する							
OK キャンセル							

図2 「動的に名前がつけられた子の設定」ダイアログボックス

図3はproductノードのためて動的なノード名が有効化されている場合、どのようことポーネトか表示されるかを示しています。コポーネトの外観が大幅に変更されていることに注意してくたとい。



図3 (要素のためご 動的なノード名が有効化されている場合

コンポーネントを標準モードに切り替えるには、product ノードを右クリックして、コンテキストメニューからオプション「動的な名前を持つ 子要素を表示する」を無効化します。

下のイメージは、ノードの属性への動的なアクセスが有効化された時、同じエンポーネントかどのように表示されるかを示しています。 product 要素を右クリックして、エンポーネントが取得され、コンテキストメニューから「動的な名前を持つ属性を表示する」を選択します。



図4 (属性のために)動的なノード名が有効化されている場合

コンポーネントを標準モードに切り替えるコよ、product ノードを右クトクして、コンテキストメニューから、「動的な名前を持つ属性を表示する」オプションを無効化します。

図3と図4に示されているようにコンポーネトの外観はノード(この場合は、product)が動的なノード名」なモードに切り替えられると変更されます。新しい外観は次のアグランを取る可能性を広います。

- すべてのノードの子要素おけば属性のノストを読み取り、おけば書き込みます。これらは、element() おけはattribute() アイテムによりそれぞれ提供されます。
- 各子要素おけ調性の名前を読み取り、おけま書き込むます。名前は、node-name()とlocal-name()アイテムことり 提供されます。

- 要素の場合、各子要素の値を特有のデータ型として読み取る、おけは書き込みます。この値は、型キャストノード(この場合は xs:string アイテム)により与えられます。要素のみか型キャストノードを持つことができることに注意してくたさい。属性は、常に、 「文字列」型として処理されます。
- 名前別に子要素をグループ化します。例に関しては次を参照してくたさい例名前別にノードをグループ化ませる。

「動的なノード名」モード内で作業することのできるノード型は、下て説明されています。

element()

ターゲトコポーネトと比較すると、このノードはノースコポーネト内で異なる振る舞いをします。ソースコポーネト内で、ノードの子要素をシーケンスとして提供します。図3 el ement()は、全てのproductの子要素のJスト(シーケンス)を与えています。例えば次の XMLから作成されナシーケンスは、3つのアイテムを含みます(これは、product 3つのつの子要素が存在するからです):

```
<product>
<id>1</id>
<color>red</color>
<size>10</size>
</product>
```

シーケンス内の各アイテムの実際の名前と型は、node-name()ノードと型キャストノードによりそれそれ与えられています上記を理解するいる、ソースからのデータXMLをターゲルXMLに次のようご変換する必要かあると仮定してくたさい



図6 シーケンス内の各アイテムの実際の名前と型は、node-name()ノードと型キャストノードによりそれぞれ与えられています上記を理解するコよソースからのデータXMLをターゲルXMLに次のようご変換する必要があると仮定してください

マピングの目的を達成するマピングは以下のようてはますマピング



図 7 (MapForce 内で) XML 要素名 を属性値 にマルングする

node-name()とtext()はシーケンス内の各アイテムの植と実際の名前を与えますがelement()のここでの役割は productの子要素のシーケンスを提供します。このマッピングはチュートリアルのサンプルが付随しており、以下で更に詳しく説明されています。 ます:<u>例:要素名を属性値にマップする</u>。

ターゲトコンポーネント内では element() は ソース内の各アイテムのマンピングの基本ルールの列外であるとおり自身では何も作成しませんが、1つのターゲットアイテムを作成します。実際の要素は(node-name()の値を使用して)型キャストノード ど自身の名前を使用して)名前テストノードにより作成されます。

attribute()

図4内で表示されているようここのアイテムはマンピングのランタイムでノードの全ての属性へのアクセスを有効化します。ソースエンポーネント内で、接続されたノースノードの属性をシーケンスとして提供します。例えば、次のXML内で、シーケンスは、(product には2つの属性が存在するため2つのアイテムを含みます:

product id="1" color="red" />

attribute() ノードは、シーケンス内の各属性の値のみを文字列の型とて与えます。各属性の名前はnode-name() ノードにより 与えられます。

ターゲトコンポーネント内では このノードは 接続されナシーケンスを処理し、シーケンス内の各アイテムのナカン属性の植を作成します。属性の名前は、node-name() により与えられます。例えば、ソースからのデータXMLをターゲナ XMLを以下のようこ処理すると想定します。

1 2 3 4 5 6 7	<pre><?xml version="1.0" encoding="UTF-8"?> </pre> <pre>cyroducts xsi:noNamespaceSchemaLocation="// </pre> <pre><pre>cyroduct> </pre> <pre><attribute name="id" value="1"></attribute> </pre> <pre><attribute name="color" value="red"></attribute> </pre> <pre><attribute name="size" value="10"></attribute></pre></pre>		
7 8 9 10 11 12 13	<pre><pre><pre>c<pre>product> <attribute name="color" value="blue"></attribute> <attribute name="size" value="20"></attribute> </pre></pre></pre></pre>	1 2 3 4 5 6	<pre><?xml version="1.0" encoding="UTF-8"?> <pre><products 1"="" color="red" size="10" xsi:nonamespaceschemalocation="//\ <product id="></products> <product color="blue" id="2" size="20"></product> <product color="green" id="3" size="30"></product> </pre></pre>
14 15 16 17 18	<attribute name="id" value="3"></attribute> <attribute name="color" value="green"></attribute> <attribute name="size" value="30"></attribute>		

図8 属性の値を属性名にマッピングする(必須)

この目的を達成するマパングは以下のようこなります



図9 (MapForce 内で)属性の値を属性名にマピングする

メモ この変換はノードの属性へのアクセスを有効化せずご行うことができます。ここでは、どのようこattribute()がターゲナンポーネント内で動作するかを表示しています。

このマピングを再作成する場合は、〈マイドキュメント〉¥Altova¥MapForce2021¥MapForceExamples¥Tutorial¥フォルダー 内のConvertProducts.mfd マピングと同じ XML コンポーネントを使用しています。マピングとして使用します。唯一の違いよター ゲトがノースコントリノースカダーゲナトロンることのみです。ソースコンポーネントのナメの入力データコよ、属性の値を実際に含む XML イン スタンスが必要についます。例

上記のコードリスティングでは、名前空間とスキーマ宣言が、ここでは簡単にするなっと当時されていることに注意してくたさい。

node-name()

ソースエポーネト内では node-name() は element()の子要素の名前、おけattribute()の要素の名前をそれそれ提供します。デフォルトでは、提供される名前は xs:QNameの型です。名前を文字列として取得するコよ local-name() ノードを (図3を参照してくたさい)、おけよ 関数 QName-as-string を使用します。

ターゲナコポーネナ内ではnode-name()はelement()おはattribute()内に含まれる各要素おは属性の名前を書き込みます。

local-name()

このノードは node-name () と同様の作動をしますが、異なる点は xs:QName の代われにxs:string が使用されていることです。

型キャストノード

ソースエレポーネント内では、型キャストノードはelement()内の各子要素の値を提供しますこのノードの名前と構造は、「動的」名前がつたれた子の設定」、ダイアログボックス(図 2)から選択された型により異ないます。

ノードの型を変更する場合は、「選択の変更」(🔚) ポタンを使用して、スキーマンイルドカード (xs:any) を含む、使用できる型から 希望する型を選択します。詳細に関しては、次を参照してくたさい、特定の型のノードにアクセスする。

ターゲナコンポーネント内で、型キャストノードは、element()内に含まれる各子要素の植特有のデータ型として書き込みます。希望するデータ型は「選択の変更」()) ポタンをクリックすることには、選択することができます

名前テストノード

ソースエンポーネト内では、名前テストノードは、ソースインスタンスから名前別に子要素をグループ化、おオオフィルターする方法が提供されています。正確な型を使用してマンピングが、インスタンスデータにアクセスしていることを保証するオンガに子要素を名前別にフィルターする必要かある場合があります。(次を参照してくたさい、特定の型のノードへのアクセス)。例えば、例名前別にノードをグループ化おオオフィルターするを参照してくたさい。

一般的には、名前テストノードは、値とサブ構造の読み取りと書き込みをする通常の要素ノードとまま同様の作動をします。しかしなから、動 的なアクセスが有効化されていると、マメピングのセマンティクスが異なるため、制限が発生します。例えば、2つの名前テストノードを連結するこ とってきません。

ターゲナ側では、名前テストノードは、接続されているノースシーケンスにアイテムが存在するため、出力内と同じ数量の要素を作成します。 これらの名前は、node-name()にマップされている値を上書きします。

必要であれば、エレポーネトから名前テストノードを非表示にすることができます。これを行うには、element()ノードの横の「選択の 変更」(
同)ボタンをクリックします。そして、「動的な名前を持つ子の設定」ダイアログボックスから、「名前テストノードの表示」チェックボックスのチェックを解除します。

4.15.2 特定の型のノードへのアクセス

前のセクションで説明されているとおり、ノード名へのアクセスの取得は、ノードを右クリックして「動的な名前を持つ子要素を表示する」コンテキストメニューコマンドを選択し、ノードの全ての子要素にアクセスすることができます。特定の型のノードにより値をアクセスすることはできます。すが、マメピングのランタイムでは、node-name()ノードから各子要素の名前にアクセスすることができます。下のイメージでは、型キャストノードは、text()ノードの横つかます。



子要素のデータ型は、マピングのランタイム前はおかりません。おこ、各子要素にとり異なる可能性があります。例えば、XML インスタン スファイル内のproduct ノードは、型xs:integerの子要素idと型xs:stringの子要素 sizeを持つ可能性があります。 特定の型のノードコンテンメにアクセスするはよ下に表示されるダイアログボックスがシードの子要素への動的なアクセスを有効化するたびに 開かれます。element()ノードの横の「選択の変更」())がタンをクリックすることにより、このダイアログボックスを後で開くことができます。
● 動的に名前の付けられた子の設定 ● ■	×
これらの設定を使用してジェネリックな構造を使用する子要素全体を構成します。	
動的に名前の付けられたアイテムのコンテンツにアクセスするための型を選択してください:	
□ • · · · · · · · · · · · · · · · · · ·	
ここで型を選択すると、構造かマッピンクためだけに使用できるようになります。 ランタイムでは実際の型はチェックされません。	
■ノード名を固定し、要素をフィルター、または、作成するためにテストノードの名前を表示しま	す。
▼ 親要素のスキーマ子要素を表示する	
ок キャンセル	
	-//

「動的に名前がつけられた子の設定」ダイアログボックス

マピングのランタイムに各子要素のコンテンソンアクセスするコよしくつかのオプションの初ます

- 1. コンテンンを文字列としてアクセスするはよ、上のダイアログボックスでtext() チェックボックスを選択します。この場合、ダイアログボックスか閉じられると、text() ノードがコンポーネント上に作成されます。このオプションは、コンテンツが、xs:int、xs:string などの単純型の場合、適切です。例 要素名を属性値にマップするで詳しく説明されています。text() ノードは、現在のノード の子ノードがデキストを含むことができる場合表示されることに注意してくたさい。
- 2. スキーマー:許可されている特定の複合型としてコンテンソニアクセスすることができます。カスタム複合型が、選択されているのために ノードグロー・ジルニ許可されていると、上のダイアログボックスで使用することが可能にない、横のチェックボックスを選択することができ ます。上のイメージでは、グロー・ジルニ定義されている複合型に存在せず、この選択には使用することができません。
- 3. コンテンンを型としてアクセスする。これは、高度のマンピングのシナリオで役に立ちます(次を参照してくたさい、「更に深い構成にア クセスする」を参照してくたさい。これを行うコは、xs:anyTypeの横のチェックボックスを選択してくたさい。

マピングランタイムでは、MapForce(型キャストノードを介して) には、実際のインスタンスノード型に関する情報がないことに注意 してくたさい。ですから、マピングは、正確な型を使用してノードコンテンソニアクセスする必要があります。例えば、ソースXML インスタ ンスか多種の複合型の子ノードを持つことを希望する場合、以下を行います。

a)型キャストノードを一致させる必要が有る複合型に設定します(上のJストの2番目のアイテムを参照してくたさい)。

b) 一致する必要のあるインスタンスのみから読み込むためにつイルターを追加します。このテクニックは、次で説明されています: <u>例 名</u>前別にノードをグループ化ませる

更に深い構成にアクセスする

スキーマワイルドカードの選択により、ノードをノードの直下の子よりもスキーマ内の更に深しレベルでアクセスすることができます。高度なマピングのシナリオではとても役に立ちます。マピングは、XML ノードの直下の子しかアクセスすることができないため、例<u>要素名を属性値にマップする</u>等の簡単なマピングでは、このテケニックを必要としません。しかしなから、動的に更に深い構造にアクセスする必要かある場合、 例えば「孫」にアクセスが必要な場合、以下の方法でアクセスすることが可能です。

- 1. 新規マピングを作成する。
- 挿入メニューからXML スキーマ/ファイルを挿入する」をクリックし、XML インスタンスファイルを参照します。(子の例では、 くマ イドキュメント> Altova Map Force 2021 Map Force Examples ¥Tutorial ¥ フォルダーからの Articles.xml ファイル です)。
- 3. Articles ノードを右クトックして、「動的な名前を持つ子要素を表示する」コンテキストコマンドを選択します。
- 4. 「動的に名前がコたわた子の設定」ダイアログボックスからxs:anyTypeを選択します。
- 5. xs:anyType ノードを右クリックして、「動的な名前を持つ子要素を表示する」コンテキストコマンドに対して選択します。
- 6. 「動的に名前がコナられた子の設定」ダイアログボックスからtext()を選択します。

뎥	Articles	
ē	File: Articles.xml File/String	ار
ц\$ ¹ -6	Articles	ե
¢	🕀 🛠 element() 🔚	ել
¢	node-name()	ել
¢	local-name()	ել
¢		ել
	= xs:anyAttribute	
¢	Abc text()	ել
¢	🕀 🛠 element() 🔚	ել
¢	"""node-name()	ել
¢	local-name()	ել
¢	Abc text()	ել
	🔿 xs:any 🔚	
¢	🕀 🔿 Article	ել
¢	() Number	ել
¢	Name	ել
¢	SinglePrice	ել
		1

上記のエレポーネントで、2つのelement()ノードが存在すること注目してくたさい。2番目のelement()ノードは、Articles.xml *へスタンス*内の<Articles>ノードの孫に動的なアクセスを与えます。

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Articles.xsd">
```

<article></article>
<number>1</number>
<name>T-Shirt</name>
<singleprice>25</singleprice>
<article></article>
<number>2</number>
<name>Socks</name>
<pre><singleprice>2.30</singleprice></pre>
<article></article>
<number>3</number>
<name>Pants</name>
<singleprice>34</singleprice>
<article></article>
<number>4</number>
<name>Jacket</name>
<pre><singleprice>57.50</singleprice></pre>

Articles.xml

例えば「孫」要素名(Number、Name、SinglePrice)を取得するコよ2番目のelement()ノードの下のlocal-name()ノ ードからターゲオノードに接続を描きます。同様に、「孫」要素の値(1、T-Shirt、25)を取得するコよ text()ノードから接続を描 きます。

この別は適用することはできませんが、実際のシナリオでは、更に深いレベルにアクセスするために、動的なノード名を後のxs:anyType ノードに対して有効化することができます。

以下の点に注意してくたさい

- <u>IVE</u> ポタンを使用して、生成された型を現在のスキーマから選択し、異なるノード内に表示することができます。この方法は、生成されたスキーマの型から、おけはスキーマの型へマップする必要がある場合のみに役に立ちます(次を参照してくたさい、生成された XML スキーマ型)。
- element()ノードの横の「選択の変更」()) ボタイよ このトピックで説明されている「動的に名前の付けられた子の設定」ダイアログボックスを開きます。
- xs:anyAttribute 属性の横の「選択の変更」(3)) がなくこと、スキーマ内でグロー・ジルご定義された属性を選択することができます。同様に、xs:any 要素の横の「選択の変更」(3)) にとしスキーマ内でグロー・ジルご定義されている要素を選択することができます。これにより、スキーマフイルドカードへ、おけはスキーマフイルドカードからのマッピングと同じように動作します(次も参照してください) ワイルドカード xs:any / xs:anyAttribute)。このオプションを使用する場合、選択された属性おこには要素がスキーマログルに存在できることを確認してください。

4.15.3 例: 要素名を属性値にマップする

この例は、XML ドキュメントから、要素名をターゲット、XML ドキュメント内の属性の値にマップする方法について説明しています。この例に は、付随するサンプルマメピングがあり、次の、マを使用して検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ConvertProducts.mfd。

例を理解するために、XML ファイルは製品のノストを含むと想定しましょう。各製品は次のフォーマナを有します:

<product>

```
<id>l</id>
<color>red</color>
<size>10</size>
</product>
```

目的は、各製品の情報を名前と値ペアに変換することです。例:

```
<product>
    <attribute name="id" value="1" />
    <attribute name="color" value="red" />
    <attribute name="size" value="10" />
</product>
```

上記のようなデータマッピングを行うさま この例でも使用されていますが、グノード名への動的なアクセズと呼ばれる MapForce 機能を使用 しています。マッピングを実行する際、「動的」 とま、(値だけではなく)ノード名も読み取ることができ、これらの名前を値として使用することを 意味します。 必要とするマッピングをしくつかのシンプルなステップを以下に示されるように作成することができます。

ステップ 1: ソース XML コンポーネントをマンング に追加する

 「挿入」メニューから、「XML スキーマ/ファイル」をクリックして、〈マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Products.xml を参照します。このXML ファイルは、 同じフォルダー内にある Products.xsd スキーマを指します。

ステップ 2: ターゲナ XML コンポーネントをマピングに追加する

 「挿入」メニューから「XML スキーマ/ファイル」をクリックして、次のファイルを参照します、くマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ProductValuePairs.xsd。インスタンスファイルを与 えるように問われると、「スキップ」をクリックします。ルト要素を選択するようにプロンプトされると、ルト要素として、 productsを選択します。

この時点では、マピングは、以下のようてなります。

ステップ 3: 子ノードへの動的なアクセスを有効化する

- 1. products ノードを右クトックして、コンテキストメニューから、「動的な名前を持つ子要素を表示する」を選択します。
- 2. 開かれるダイアログボックス内から、text()を型として選択します。他のオプションをそのままします。

● 動的に名前の付けられた子の設定 ●	×
これらの設定を使用してジェネリックな構造を使用する子要素全体を構成します。	
動的に名前の付けられたアイテムのコンテンツにアクセスするための型を選択してください:	
□ end productType □ end xs:anyType ☑ end text()	
ここで型を選択すると、構造かマッピンクためだけに使用できるようになります。 ランタイムでは実際の型はチェックされません。	
■ノード名を固定し、要素をフィルター、または、作成するためにテストノードの名前を表示しま	す。
✓ 親要素のスキーマ子要素を表示する	
OK キャンセル	

text()ノードがノースコンポーネントに追加されていることに注意してくたさい。このソードは、それぞれの子アイテムをマンピングに与えます(この場合は「id」、「color」、「size」の値です)。





ステップ 4: マピング 接続を描く

マノピング 接続 A、B、C、D 下に表示されるとおり、描きます。任意で、各接続上かをダブルクトックし、テキスト「A」、「B」「C」、および「D」をそれそれ、詳細ボックスに入力します。



ConvertProducts.mfd

上で説明されているマピングでは、接続 A は、ソース内の各製品のために、ターゲナト内に製品を作成します。これまでは、これはノード名を指され、標準的な MapForce 接続でした、しかしなから、接続 B は、product の各対応する子要素のために、新規の要素を attribute と呼ばれるターゲナト内に作成します。

接続 B は、マピング内でとても重要です。この接続の目的である product の子要素のシーケンスを、ソースからターゲトに運びます。実際の名前おける植を運ばれため、以下として理解される必要が決ます。ソース element() にN 子要素が存在する場合、ターゲト内のアイテムのN インスタンズを作成します。この特定の場合は、ソース内の product には、3 つの子要素 (id、 color とsize)が存在します。これは、ターゲト内の各 product がattribute とう名前の3つの子要素を持つことを意味します。

この例で説明されていませんが、同じルールが、attribute()の子要素をマップするために使用されます。ソース attribute()アイテムが、N子属性を持つ場合、ターゲメト内のそのアイテムのNインスタンスを作成します。

次に、接続Cは、productの各子要素の実際の名前をターゲオにピーします(文字通り、「id」、「color」、と「size」)。

最後に、接続 D は、製品の各子要素の値を文字列の型として、ターゲナトにコピーします。

マルングの出力をルビューするコよ「出力」ダブをクリックして、生成されたXMLを確認してください。マルングの意図とする目的である 通り、出力は、データが名前と値のペアとして保管されている、複数の製品を含んています。

xml version="1.0" encoding="UTF-8"?	
<pre><pre>products xsi:noNamespaceSchemaLocation="ProductValuePairs.xsd"</pre></pre>	•
<pre>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></pre>	
<pre><product></product></pre>	
<attribute name="id" value="1"></attribute>	
<attribute name="color" value="red"></attribute>	
<attribute name="size" value="10"></attribute>	
<pre><product></product></pre>	
<attribute name="id" value="2"></attribute>	
<attribute name="color" value="blue"></attribute>	
<attribute name="size" value="20"></attribute>	
<pre><product></product></pre>	
<attribute name="id" value="3"></attribute>	
<attribute name="color" value="green"></attribute>	
<attribute name="size" value="30"></attribute>	

生成されたマッピング出力

4.15.4 例:名前別にノードをグループ化またはフィルターする

この例は XML プロ・ティノスト(ませまXML プロ・ティノスト)からキー値ペアを読み取りCSV ファイルへ書き込むマメピングをデザインする 方法を説明します。(XML プロ・ティノストは macOS とiOS オブジェクトの情報を XML フォーマナ で保存する方法を示します。詳細 に関しては、次を参照してください

<u>https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/PropertyLists/UnderstandXMLPlist/UnderstandXMLPlist/UnderstandXMLPlist.html</u>。) このサンプルは、マングサンプルが半い、次の、なで検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ReadPropertyList.mfd。

下のコードリストはノースXMLファイルを表しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist SYSTEM "https://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
     <key>First Name</key>
     <string>William</string>
     <key>Last Name</key>
     <string>Shakespeare</string>
     <key>Birthdate</key>
     <integer>1564</integer>
     <key>Profession</key>
     <string>Playwright</string>
     <key>Lines</key>
     <array>
         <string>It is a tale told by an idiot,</string>
         <string>Full of sound and fury, signifying nothing.</string>
      </arrav>
   </dict>
</plist>
```

マピングの目的は、プロ・ディリストファイル内の<dict>ノードの下で検出された特定のキー値ペアからCSV ファイル内に新し、ラインを 作成することです。具体的には、マピングは<key> - <string> ペアのみをフィルターしなくてはないません。他のキー値ペア(例えば <key> - <integer>)は、無視されます。CSV ファイル内で、リストは、プロ・ティの値からコンマで区切られているプロ・ティ名を保存しな くてはないません。すなわち、出力は以下のよう」ないます:

First Name,William Last Name,Shakespeare Profession,Playwright

この目的を達成するコよ、マピングは、dict ノードのすべての子ノードへの「動的」なアクセスを使用します。第21こ、XML ファイルから取 得されたマメピングはキー値ペアをグループ化するためにgroup-starting-with 関数を使用します。最後に、マメピングは、フィルターを使用 してノード名が「文字列」であるノードをフィルターします。

下のステップは必要とされるマッピングの作成方法を示しています。

ステップ 1: ソース XML コンポーネントをマッピングに追加する

 「挿入」メニューからXML スキーマ/ファイル」をクトックします。次のファイルを参照します: <マイドキュメント
 >\Altova\MapForce2021\MapForceExamples\Tutorial\plist.xml。このXML ファイルは、同じフォルダー内の plist.dtd スキーマを指します。

ステップ 2: マピング マーゲナ CSV コンポーネントを追加する

- 1. 「挿入」メニューから「テキストファイル」をクリックします。プロンプトされると、「標準 CSVのための単純処理を使用する..」 オプションを選択します。
- 2. 「フィールドの追加」をクリックして、CSV フィールドをエレポーネンに追加します。ト
- 3. 各フィールド名をダブルクトックして、最初のフィールド名として「Key」を入力し、「Value」 を2番目のフィールドの名前として入力 します。「Key」 フィールドは、プロ・ティ名を保管し、「Value」フィールドはプロ・ティの値を保管します。 CSV エンポーネントの詳 細口乳しては、次を参照してください、 <u>CSV とテキストファイル</u>。

Key		Value	T
string	•	string 🗾	
]
•		4	
フィールドを追加(<u>A</u>)	フィールドを挿入(N) フィールドを肖		

ステップ 3: フィルター と関数を追加する

- 1. equal, exists とgroup-starting-with 関数をライブラリウィドウからドラッグしマピングリボロップします。関数に関する一般的情報は次を参照してください: <u>関数</u>。
- 2. フィルターを追加するは、「挿入」メニューから Filter: Nords/Rows」をクリックします。フィルターに関する一般的情報は次を参照してくたさい、フィルターと条件。
- 3. 「挿入」メニューカジ定数」をクリックし、テキスト「string」を入力します。
- 4. ソースンポーネト内で、dictノードを右クトクし、コンテキストメニューから「動的な名前を持つ子要素を表示する」を選択します。「動的に名前がコルトれた子の設定」ダイアログボックスでは、「ノード名を固定し、要素をフィルター、または作成するために名前テストノードを表示する」のチェックボックスが選択されていることを確認してくたさい。

● 動的に名前の付けられた子の設定 ■ ■	3
これらの設定を使用してジェネリックな構造を使用する子要素全体を構成します。	
動的に名前の付けられたアイテムのコンテンツにアクセスするための型を選択してください:	
□	
□ 唱 date □ 唱 dict	
□ el false □ el integer	
□ PB toxt()	
-	
ここで型を選択すると、構造がマッピングためだけに使用できるようになります。 ランタイムでは実際の型はチェックされません。	
☑ノード名を固定し、要素をフィルター、または、作成するためにテストノードの名前を表示します。	
■ 親要素のスキーマ子要素を表示する	
OK キャンセル	
	11.

5. 下に表示されるように接続を描きます。



ReadPropertyList.mfd

マピングの説明

ソースコンポーネオト上の要素()アイテムは dict ノードの全ての子を group-starting-with 関数へのシーケンスとして提供します。 す。group-starting-with 関数は 名前 key を持つノードが発生すると新しし グループを作成します。 exists 関数は この条 件をチェックして、結果をブール値 true false としてグループ関数に返します。

各グループのために、フィルターは equal 関数を使用して現在のノードが 文字列」と等しいかをチェックします。名前自身は、文字列としてノードの名前を与えるLocal-name() から読み込まれます。

ターゲトコンポーネントへの接続にコお以下の役割があります

- フィルターの条件が true の場合のみ、新しい行がターゲナ CSV 内に作成されます。
- Key (プロ/ディ名)は、ソース内のkey要素の値から得られます。
- Value (プロ、ティ値)は string 名前テストノードから得られます。

4.16 StyleVision を使用してマッピングの出力のスタイルを指定する方法

Altova StyleVision が使用中のエピューターにインストールされていると仮定して、マピングのターゲトコンポーネトがXML である箇所は、マピングの出力をHTML、RTF、PDF とW ord 2007+ ドキュメントとしてプレビューし、保存することができます。StyleVisionのEnterprise エディションの使用されている場合、プレビュー内でチャートを表示することができます。

マンプが、これらのフォーマトのプレビューをサポートする場合、出力 タブの横に使用することのできる追加タブが表示されます。例えば マッピング DB クエリ 出力 100 HTML 100 RTF 100 PDF 100 Word 2007+

StyleVision プレビュータブ (MapForce Enterprise Edition)

以下に注意してくたさい

- Style Vision Professional がインストールされていると、HTML とRTF 出力をパビューすることができます。Style Vision Enterprise を使用すると、HTML、RTF、PDF とWord 2007+ 出力をパビューすることができます。
- マナングの出力をPDF とてルビューするコよ Java、Acrobat Reader、とFOP (Formatting Objects Processor) バージョン 0.93 おけは1.0. FOP が StyleVision をインストールはオプションが選択されていな 限り、 StyleVision と共にインストールされて、る必要があります。
- MapForce の64ビットエディションの場合、W ord 2007+ とRTF プレビューは、埋め込まれていないアプリケーションとして開かれます。
- マピングがソースとターゲト(パススルーコンポーネント)としての役割を果たすコンポーネントを含む場合、コンポーネントの「プレビュー」」 ビュー」、ポタン

 が有効化されている場合のみ、Style Vision でのプレビューが可能にないます。このようなマピングの詳細に関しては、次を参照してくたさい、チェーンマピング。

マピングからデータをこのようこプレビューするコよ、以下の必要条件が存在します

- Altova Style Vision カロピューターに スタイアロン、おける Mission Kit の一部とて、インストールされている必要 かあります。
- ターゲトコポーネトはよ、関連付けられたStyleVision Power Stylesheet(SPS)ファイルが必要です。StyleVision を使用して、スタイルシートを作成、おけは編集することができます。MapForce で直接スタイルシートを編集、おけま変更すること はできませんが、StyleVision内のMapForceか引くことはできます。スタイルシートの準備ができると、下に示されるようこタ ーゲナ MapForce コポーネナトに割り当てることができます。

ターゲナコンポーネントにStyle Vision パワースタイルシートを割り当てる

- 1. Style Vision内で、必要とするスタイルシートファイルを作成します。MapForce コンポーネントと同じ XML スキーマがソースと、 て使用されていることを確認してください。
- 2. MapForce 内で、ターゲオ XML コパーネトを右クリックし、「プロパティ」を選択します。
- 3. コンポーネント設定ダイアログボックス内で、「StyleVision Power スタイルシートファイル」の横から、作成済みのスタイルシートファイルを選択します。

StyleVision Power Stylesheet ファイル					
CompletePO.sps	参照	編集			

メモ Style Vision Power スタイルシートファイルへの マは、絶対おける相対的であることができます。次を参照してくたさい、相対的 おしい絶対パマの使用。 Style Vision により生成された出力を保存するStyle Vision により生成されたHTML、PDF、RTF、おけるW ord 2007+ 出力を、他のマルングの保存と同様に保存することができます。具体的には、マルングのプレビューを行うには、以下の内の「つをおこなって くたさい

- 「生成された出力を保存する」(1) ツールレーボタンをクリックします。
- 出力 メニューから「出力ファイルを保存する」をクリックします。

Altova 製品のッシケージを使用して、HTML、PDF、RTF、Word 2007+ ファイルを自動'生成する

マルングが同じ、おけまなるマシン、おけまプラナフォームで)HTML、PDF、RTF、およびWord 2007+ファイルを自動的に生成する場合、(MapForce とStyleVisionの機能性を拡張する個別にライセンスされるサーバー製品である)MapForce Serverと StyleVision Serverを使用して行うことができます。このシナリオの場合、各アプリケーションが以下の特別な役割を果たします:

- MapForce (例えば、データベースからXML への)データ変換の入力と出力を定義するマッピング (.mfd ファイル)のデザインを行うことかできます。
- MapForceServer マピング(.mfx ファイル)をコマイドライン おうはAPI から(同じ、おうは異なるオペレーティングシステム 上で実行可能なマピングを実行します
- Style Vision マンピングの出力をHTML、PDF、RTF、Word 2007+に出力するためこ必要な、スタイルシート (.sps ファイル) をデザインすることができます。
- Style Vision Server マルングの出力をターゲオフォーマナに変換する.sps スタイルシートを作成します。これは、コマド ライン、おけよAPI(同じ、おけは異なるオペレーティングシステム)から行うことができます。
- (個別のライセンズを必要とする)Style Vision Server とMapForce Server は、任意でFlowForce Server の管理下で 作動します。このシナリオの場合、MapForce マナングとStyle Vision 変換は、予定通り、実行、トリガー、おけは需要に応じて実行することができ、完全に自動化することができます。

4.16.1 StyleVision スタイルシートを使用したマッピングの例

MapForce サンプルフォルダー(<マイドキュメント>¥AltovaWapForce2021WapForceExamples》内のマピングの多くは ターゲトコンポーネトに割り当てられている StyleVision Power Stylesheets (.sps ファイル) を含んでいます。この場合、マピングは を StyleVision プレビュータブ含んでいます。

マッピング | DB クエリ |出力 | 🛛 HTML | 🗿 RTF | 🗿 PDF | 🗿 Word 2007+

StyleVision プレビュータブ (MapForce Enterprise Edition)

次の 次で見つけることのできる CompletePO.mfd はスタイルシートを使用したマンピングの一例です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\CompletePO.mfd。このマンとングは、XML フォーマナで注文を表示しま す。ターゲナコンポーネントを右クトックして、プロパティを選択すると、.sps ファイルは割り当てられていることに気が付きます。

Stylevision Power Stylesneet JP1 JV		
CompletePO.sps	参照	編集

HTML フォーマトで表示される出力データを確認するために出力 タブをクリックします。

						Fax +1 ((321) 555 5155 - 9
						of	fice@nanonull.com
							www.nanonull.com
							www.manonun.com
Purcha	ase Orde	er Numbe	r: PO				
	TO:			Mrs.	Mr. Te	d Little	
			Long Way				
			Los-Angeles	3			
			CA 34424				
			Our Custome	er Identi	fier:	ID-3	
Order	Date:						
Shippi	ng Date:				_		
Item	Quantity	Name	Unit Price (\$)	Total (\$)			
3	5	Pants	34	170			
1	17	T-Shirt	25	425			
				595			
Other	Common	te.					
other	commen						
Authorized Signature Date							

CompletePO.mfd のHTML パビュー(MapForce Enterprise edition)

4.17 マッピングドキュメンテーションの生成

ドキュメンテーションの生成コマイドにより、マピングに関する詳細なドキュメンテーションをHTML、Microsoft Word (.doc)、RTF書 式で生成することができます。Style Vision がインストールされているとPDF書式でドキュメントを生成することもできます。

ょ

- MS W ord フォーマナにてドキュメンテーションの生成を行うには MS W ord (バージョン 2000 以降)がインストールされ ている必要があります
- PDF 書式でドキュメトを生成、おけま生成されナギャメトのデザインをカスタマイズする場合 Style Vision がインストー ルされている必要があります。

デフォル・では、キュメンテーションは含まれるコンポーネント、表示される、なの深さ、および他の設定を構成することのできる固定されたデザインで生成されます。StyleVision がインストールされている場合、含まれているStyleVision Power Stylesheets (SPS) ファイルの追加機能を活用、おけよStyleVision 内で自身のデザインを作成することができます。

マッピングドキュメンテーションの生成方法

1. 「ファイル」メニューから「ドキュメントの生成」をクリックします。これにより「ドキュメントの生成」ダイアログボックスか開かれます。

ドキュメントの生成	×
ドキュメント デザイン ● HTML、Word、または RTF フォーマットにおける ドキュメン 〇 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: %AltovaMapForceDoc%¥Funct	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 が必要) ionsCallGraph.sps
出力書式 HTML Microsoft Word RTF PDF(上記参照) 	パスの長さ制限 ☑ 入力アイテム(I) 3 ☑ 接続(C) 3 ☑ 出力アイテム(I) 3 ・
 □出力を複数のファイルに分割 ▽生成後に結果ファイルを表示 含める対象 ▽入力コンポーネント ▽変数コンポーネント ▽出力コンポーネント ▽すっ定義関数 ▽定数 	詳細 ☑注釈(A) ☑接続されたノード(N) ☑データ型(D) ☑接続されていないノード(U) ☑ライブラリ名(L)
すべて選択 チェックをすべて解除	すべて選択 チェックをすべて解除 OK キャンセル

2. 必要とされる設定を選択し「OK」をクトックします。

構成可能な設定は下に説明されています。

ドキュメントのデザイン

- 「固定デザインを使用」を選択して内蔵のドキュメンテーションテンプレートを使用します。
- 「ユーザー定義デザインを使用」を選択して Style Vision 内で作成された定義済み Style Vision Power Stylesheet を使用します。…、Documents、Altova、MapForce2021、Documentation、MapForce、フォルダー内で SPS ファイルを使用することができます。詳細に関しては、定義済み Style Vision パワースタイルシートを参照してくたさい。
- 「参照」をクリックして、定義済みのSPS ファイルを参照します。
- 「編集」をクルクして Style Vision をローンチし、 Style Vision ウィドウ内で選択された SPSを開きます。

出力書式

- 次の出力書式の一つを選択します: HTML、Microsoft W ord、RTF、おけはPDF. Microsoft W ord ドキュメトは固定 されたデザインを使用する場合.doc ファイル拡張子と共に作成され、Style Vision SPS を使用して生成する場合.docx ファイル拡張子と共に作成されます。ます。PDF 出力書式はStyle Vision を必要とし、Style Vision SPS を選択した場合 のみ使用することができます。
- 入力、おゴオ出力コンポーネトなどの個別のコンポーネト別に1つのファイルを与えるなど、複数のドキュメンテーションファイルを 生成する場合「出力を複数のファイルに分割」を選択します。固定されたデザインを使用する場合、複数のドキュメント間のシ クが自動的に生成されます。
- 「生成後に結果ファイルを表示」オプションが選択されている場合、MapForce は生成されたファイルを適用できる場合デフォ ルトのブラウザー、ませるエアプリケーション内で開きます。

パマの長さの制限

入力、おけ出力接続のために表示される、アの長さの最大値を定義するためにてれらのオプランを使用します。例えば、デフォルトの長さが3の場合、アイテム・マイおりてなります:.../ShortPO/LineItems/LineItem。

含まれる内容

生成されたギュメト内に含まれる特定のコンポーネントをここで選択します。

詳細

生成されたキュメンテーション内での詳細のレベルをカスタマイズするオプションを使用します。「ライブラリ名」オプションは巻数のために "core" プレフィックスを挿入します。

4.17.1 定義済みの StyleVision パワースタイルシート

When StyleVision がコンピューターにインストールされると、ビルトインの固定されたデザインの代わりに定義済みのStyleVision Power Stylesheet (SPS) ファイルをテンプレートとして選択してマンピングドキュメンテーションを生成することができます。次の定義済みのSPS スタイルシートを使用することができます:

- FunctionCallGraph.sps メインマングとユーザー定義関数の呼び出しグランを表示します。
- FunctionsUsedBy.sps -マピング内で直接、おけ間接的に使用された関数を表示します。
- ImpactAnalysis.sps ソースおびターゲホノードとターゲホノードまでの多種の関数を使用したすべてのルトをリストします。
- OverallDocumentation.sps 全てのノード、コネクタ、関数、およびターゲットノードを表示します。このテンプレートは最大の詳細を出力し、ビルトインの固定デザイン出力に合致しています。

以下に示されるようボキュメトを生成する前に必要とするスタイルシートを選択することができます。ファイルは... \MapForce2021\Documentation\MapForceフォルダー内で見つけることができます。

ドキュメントの生成						Х
ドキュメント デザイン ・ HTML、Word、または RTF フォーマットにおける ドキュメントに対して固定デザインを使用 ・ HTML、Word、RTF、または PDF フォーマットに対してユーザー定義デザインを使用(StyleVision が必要) SPS デザインを選択: %AltovaMapForceDoc%¥FunctionsCallGraph.sps v 参照 編集						
出力書式 ● HTML	%AltovaMapForceDoc%¥Funct %AltovaMapForceDoc%¥Funct %AltovaMapForceDoc%¥Impac %AltovaMapForceDoc%¥Overa	ionsCallGraph.sps ionsUsedBy.sps ctAnalysis.sps allDocumentation.sps				
○Microsoft Word ○RTF ○PDF(上記参照)		 ☑ 接続(C) ☑ 出力アイテム(T) 	3	÷		

下のサンプルはこれらのスタイルシートにより生成される出力を示しています。サンプルはMapForceを使用してインストールされたデモマンピングの1つ(PersonListByBranchOffice.mfd)により生成されています。これらのサンプルはHTML出力を示していますが、他のフォーマトのレイアナトもこれに類似しています。SPSファイルの作成、およびカスタム化に関しての清報は、<u>カスタムスタイルシート</u>を参照してくたさい。

スタイルシート "FunctionCallGraph.sps"

PersonListByBranchOffice.mfd \times +	—		×
$\leftarrow \rightarrow$ C \textcircled{a} (i) file:///C:/Users/alto \bigtriangledown \swarrow	lii\	»	≡
This report shows call graphs of the main mapping a defined functions.	and all us	ser-	^
Main mapping core.equal			
core.filter <u>user.LookupPerson</u>			
core.filter <u>user.EqualAnd</u>			
core.logical-and <u>user.Person2Details</u>			
core.concat			
user.EqualAnd core.equal core logical-and			
			~

スタイルシート "FunctionsUsedBy.sps"

PersonListByBrand	chOffice.mfd \times +	-	- 0	×
\leftrightarrow > C \textcircled{a}	i file:///C:/Users/alto	⊌ ☆	\ »	≡
This report lists a another function changes in user-o functions can be Library core	all functions and their di s. This is especially impo lefined functions in order affected.	rect and indire rtant for plan r to see what o	ect use in ning ther	~
Function	Directly used by	Indirectly us	ed by	
core.concat	user.Person2Details	Main mappin user.Lookupl	lg Person	
core.equal	Main mapping user.EqualAnd	user.LookupI	Person	
core.filter	Main mapping user.LookupPerson			
core.logical-and	user.EqualAnd	Main mappin user.Lookupl	erson	
				*

スタイルシート "ImpactAnalysis.sps"

PersonListByBranchOffice.mf	id × +	- 0	
	ile:///C:/Users/alto	⊍ ☆ II\ ≫	
This report lists every input and output node connection independently and is perfect for further impact analysis with modelling tools.			
Input Node	Functions	Output Node	
OfficeName	core.equal, core.filter	PersonList	
OfficeName	user.LookupPerson	PersonList/Person/Details	
BranchOffices/Office	core.filter	PersonList	
	1 61.	D T :	
BranchOffices/Office/Name	core.equal, core.filter	PersonList	
BranchOffices/Office/Name BranchOffices/Office/Contact	core.equal, core.filter	PersonList PersonList/Person	
BranchOffices/Office/Name BranchOffices/Office/Contact /Office/Contact/first	core.equal, core.filter	PersonList PersonList/Person PersonList/Person/First	
BranchOffices/Office/Name BranchOffices/Office/Contact /Office/Contact/first /Office/Contact/first	user.LookupPerson	PersonList PersonList/Person PersonList/Person/First PersonList/Person/Details	
BranchOffices/Office/Name BranchOffices/Office/Contact /Office/Contact/first /Office/Contact/first /Office/Contact/last	user.LookupPerson	PersonList PersonList/Person PersonList/Person/First PersonList/Person/Details PersonList/Person/Last	

Stylesheet "OverallDocumentation.sps"

PersonListByBranchO	ffice.mfd × +	:	×
\leftrightarrow > C $\textcircled{1}$	i file:///C:/Users/altova	✓ ··· ☑ ☆ >>	≡
Mapping PersonListByBranchOffice.mfd			^
Input OfficeName			
Nodes	Connections		
OfficeName Type: string Default: core.constant("Nanonull, Inc.")	<u>core.equal => a result =></u> <u>core.filter => bool on-true =></u>	PersonList Type: restriction of xs:anyType [01] Annotation: List of Persons	
	user.LookupPerson => Office Name result =>	PersonList/Person/Details Type: xs:string [01]	
Input BranchOffices (Branc	hOffices.xsd)		1
Nodes	Connections		
File: BranchOffices.xml Type: string			
BranchOffices Type: restriction of xs:anyType [01]			
BranchOffices/Name Type: restriction of xs:string			~

4.17.2 カスタムスタイルシート

ビルとインデザイノコルえ、生成されたマンピングのドキュメンテーションのためにStyleVision(<u>https://www.altova.com/ja/stylevision</u>) を使用してカスタムスタイルシートを作成することができます。フォントと他のスタイルを調整して定義済みのスタイルシートを変更することもできます。例えば、

カスタムデザインとはStyleVision Power StyleSheet(SPS)です。マピングドキュメンテーションの生成にSPSを使用する利点はボ キュメントのデザインを完全に管理できる点です。

カスタムSPS ファイルを作成するコよ 以下を必要とます:

 生成されたMapForceドキュメンテーションの構造を提供するXML スキーマ。このスキーマは MapForceDocumentation.xsd と呼ばれ MapForce インストーフレッケージュ 搭載されています。… \Documents\Altova\MapForce2021\Documentation\MapForce フォルダー内に保管されています。 MapForceDocumentation.xsd には上のフォルダー内にDocumentation.xsd ファイル かきまれていることに注意して くたさい。 カスタムデザインをテストしプレビューするデータ。次のXML ファイルをサンプルデータとて使用することができます:...
 \Documents\Altova\MapForce2021\Documentation\MapForce\SampleData\PersonListByBranch Office.xml.

上記のファイルはStyleVision内のデザイン概要ウィンドウ内で参照される必要があります。例

Design Overview X			
∇	Sources		
6	\$XML (main)		0
	Schema	MapForceDocumentation.xsd	0
	Working XML	PersonListByBranchOffice.xml	•
	Template XML		0
	XML Signature	Disabled	0
	add new Source	<u>2</u>	
∇	Modules		
	add new Module	<u>2</u>	
∇	CSS Files		
	add new CSS Fi	le	
∇	Parameters	;	
	add new Param	eter	
∇	XSLT Files		
	add new XSLT file		

StyleVision 内ではスキーマソレーウィンドウからノードをデザインエリアトドラックし、プロノティを割り当ててデザインを作成することができます。



リンクおよびイメージなどの追加コンポーネントをSPS デザインに追加することもできます。特定のフォーマトをデザイン内でルビューするには、次のダブをクリックしてくたさ、、HTML、RTF、PDF、おけよWord 2007+。詳細に関してはStyleVision ドキュメンテーションを参照してくたさ、(https://www.altova.com/ja/documentation)。

4.18 マッピングのルールと戦略

ー般的には、MapForce はデーダを直観的にマップしますが、出力に多すぎる、まけよりなすぎるアイテムか含まれているシチュエーションが存在します。このチャプターは性格ではない接続やマメングコンテキストが理由でマメングが希望しない出力を生成するシチュエーションを回避するナメの手助にを意図としています。

マピングのルール

有効であるためコまマピングココンなくとも1つのノースエンポーネトといなくとも1つのターゲトコンポーネトか含まれている必要があります。ソースエンポーネトはデータを通常はファイルませまデータベースから読み取ります。ターゲトコンポーネトは通常ファイルませまデータベ ースでデータを書き込みます。上記が正しくない個所でマピングを保存しようとすると、メッセージウィンドウィエラーか表示されます:「マピング はいなくとも2つの接続された構造を必要とします。例えば、スキーマませまデータベース構造を必要とします。」

データマピングを作成するコよマピング接続をノースとターゲトコンポーネト間に描きます。

描かれたマメピング接続全てはマメピングアルゴリズムを作成しています。マメピングラインタイムでは、MapForce はアルゴリズムを評価し、それに基づきデータを処理します。マメピングアルゴリズムの統一性と効率性は接続により異なります。。マメピングレベル、ユンポーネントレベル、お台は接続レベルで設定の一部を調整することもできすが、基本的にはマッピング接続かデータの処理方法を決定します。

接続を作成する際に以下のレールが存在することに注意してくたさい。

 ソースアイテムから接続を描く場合、マルングはノースファイルまけはデータベースからのアイテムと関連したデータを読み取ります。デ ータは、1、おけは複数の発生が存在する場合かあります(すなわち、シーケンスである場合もあります。例えば、マルングか書 籍を含むXML ファイルからデータを読み込む場合、ソースXML ファイルはし、1、おけは複数のbook 要素を含む可能性かあり ます。下のマルングでは、ソース(インスタンス)ファイルはは複数のbook 要素、おけは要素が含まれない場合かありますが、 book アイテムはマルングコンポーネント上で一度のみ表示されます。



- 2. ターゲトアイテムは装続を描く場合、マピングはその種類のインスタンスデータを生成します。ソースアイテムが単純なコンテンツを 含む場合(例えば、文字列ませば整数)およびターゲトアイテムが単純なコンテンツを受け入れる場合、MapForce はコンテン ッをターゲトアイテムコピーし、必要な場合は、データ型を変換します。受信するノースデータに近いせれ、1、複数の値を生成 することができます。次の点を参照してくたさい。
- 3. ソース内の各(インスタンス)アイテムのナムコンの(インスタンス)アイテムがターゲホ内で作成されます。MapForce内の一般的なマッピングルールです。上のマッピングを例として、ソース XML が 3つの book 要素を含む場合、3つの publication 要素がターゲホ側で作成されます。特別なケースシーケンスが、くつか存在することに注意してください。を参照してくたさい。
- 4. 各接続は現在のマメピングコンテキストを作成します。コンテキストはどのデータが現在のターゲトノードのかめにどのデータが使用できるかを決定します。コンテキストは、ですから、子のノースアイテムが実際にノースからターゲトコンポーネントにエピーされるかを決定します。接続を作成、ませば削除する場合、現在のコンテキストが不注意に変更される可能性があり、マメピングの出力に

影響を与える場合があります。例えば、同じマピングが不必要にデータベースませよ、Webサービスを呼び出す場合があります。このエレセルは下のマピングエレテキストで詳細に説明されています。

4.18.1 シーケンス

以前に記述されている通り一般的なマンピングルールボソース内の各アイテムのために、ターゲット内につつ作成する」です。ここでは、「アイテム」は以下のこつを意味します。

- 入力ファイルませまデータベースの単一のインスタンスノード
- 入力ファイルませまデータベースのゼロから複数のインスタンスノードのシーケンス

マンピングの実行中、シーケンスがターゲオアイテムに達成すると、ソースノードの数と同じ数量のターゲオを生成するルールを作成します。いくつかの例外がこのルールには存在します

- ターゲルアイテムがXMLルト要素の場合、1度のみ作成されます。シーケンスを接続すると、結果はスキーマイン対して有効でな し場合がおります。ルト要素の属性も接続されている場合 XML のシリアル化はマンピングのランタイムで失敗します。このため ルト XML 要素にシーケンスを接続することを回避してくたさい。
- ターゲトがつの値のみ受け入れる場合、1度のみ作成されます。アイテムのサンプルはつの値にのみより受け入れられます: XML 属性、データベースフィールド、単純型出力コンポーネント。例えば、下のマメピングはgenerate-sequence 関数を使用して3つの整数(1, 2, 3)のシーケンスを生成します。しかしなから、ターゲトは単一の値を受け入れる単純型出力コンポーネントのナメ出力は1個の整数を含みます。他の2つの値は無視されます。



 ソーススキーマか特定のアイテムが度のみ発生するが、インスタンスファイルコは多数のアイテムが存在する場合、MapForceは (スキーマイン従うと必ず存在する必要がある)ソースから最初のアイテムを抽出し、ターゲット内で1個のアイテムを作成します。この 振る舞しを無効化するコは、チェックボックスをコンポーネット設定から解除します。この振る舞しを無効化するコは、チェックボックス 「min/maxOccurs をベースに入力処理最適化するを有効化する」コンポーネット設定内でクリアしてください。この振る舞 しを無効化するコは、コンポーネット設定からクリア、おコは、XML コンポーネット設定を参照してくたさい。

シーケンスが空の場合、ターゲナ側には何も生成されません。例えば、ターゲナがXMLドキュメントで、ソースシーケンスが空の場合、 XML 要素はターゲナ内に作成されません。

関数も同様な方法で作動します:入力としてシーケンスを取得すると、シーケンス内のアイテムの数量だ1.1呼び出し(その数の出力を1作成します。

関数が空のシーケンスを入力として取得すると、空の結果を返し、この結果出力は生成されません。

しかしなから、関数の一部のカテゴリでは、デザインのために入力として空のシーケンスを受けると空の値を返します。

- exists, not-exists, substitute-missing
- is-null, is-not-null, substitute-null (これらの3つの関数は前の3つとエイリアスです)
- 集計関数(sum, count, など)
- (インラインではなく)正規関数であるユーザー定義関数

空の値を置き換えるコよ add the substitute-missing 関数をマピングに追加し、空の値を代替地と置き換えます。代われ、同じ結果を<u>デフォルトとノード関数</u>を使用してえることもできます。

関数には複数の入力が存在する場合があります。シーケンスが各入力に接続されている場合、通常希望される出力ではありませんが、全ての入力の連結された結果を出力することができます。これを回避するために、1つのシーケンスのみを複数の、ラメーターを持つ関数に接続してください。他のすべての、ラメーターは親まけよ地のエレポーネントからの「単純型」アイテムは接続されている必要があります。

4.18.2 マッピングコンテキスト

マルングエポーネントは深さの多くのノベルを含む階層的な構造です。他方、マルングは複数のノースとエパーネントおよび関数、フィルター、値マップなどの中間エレポーネントを持ちます。これは複数の関連性の内エンポーネントが接続されているとマルングアルゴリズムに複雑性を追加します。マルングをこのポジンヨンで実行するコよ、現在のエレテキストがそれぞれの接続で確立される必要があります。

現在のエレテキストは処理済みの各接続と共に変更されるため、複数の「現在のエレテキスト」がマンピングの実行の期間に構築されていると言うことかできます。

MapForce はターゲナルールアイテム(ノード)空開始する現在のエンテキストを常に構築します。これはマンピングの実行が実際に開始する箇所です。ターゲナルールアイテムへの接続は、関数ませま地の中間コンポーネントを含む直接ませま間接的に接続されている全てのノースアイテムともレースパックされます。関数により生成された全てのノースアイテムと結果は現在のエンテキストに追加されます。

ターゲトノードの処理後、MapForce は階層的に作動します。具体的はよ、全てのターゲトコンポーネントのマップされたアイテムが階層的に処理されます。具体的はよ、上から下にターゲトコンポーネントが処理されます。新規の各アイテムのオーダに、親コンテキストのすべてのアイテムを含む新規のコンテキストが作成されます。このオーダに、ターゲトコンポーネント内のマップされたすべての兄弟アイテムはそれぞれに対して独立していますが、親アイテムのノースデータにアクセスすることができます。

上記が実際にどのように作動するかサンプルマンピング、PersonListByBranchOffice.mfdをベースに確認してみましょう。 <マイド キュメント > ¥ltova Wap Force2021 Wap ForceExamples ¥ ディレクトリ内でこのマンピングを見つけることができます。



上のマピングではソースとターゲトコンポーネトの両方がXMLです。ソースXMLファイルコおつのOffice要素が含まれています。

前述の通り、いつも通り、マピングの実行はなーゲナル・トノード(このサンプルではPersonList)から開始されます。(フィルターと関数 を介して)接続をノースアイテム・ホレースバックすることで、ソースアイテムがOfficeであることが結論付けることができます。(他の接続バス は入力パラメーターへ接続されており、その目的に関しては以下で説明されています)。 Office とPersonList の間にオかりやすい 接続が存在する場合、一般的なマピングルールゴ従い、ソースファイル内のOffice アイテム の数だけ、マピングは PersonList インスタンスアイテムを作成します。しかしながら、フィルターが存在するためこれが発生しない 場合があ ります。フィルターはフィルターの bool 入力に接続されているブール値の条件を満たすデータをターゲメコンポーネントに提供します。 equal 関数は、オフィス名が「Nanonull, Inc.」に等しい場合 true を返します。ソース XML ファイル内のオフィス名が「つしか存在しな しため、この条件は一度のみ満たされることができます。

結果的にOfficeとPersonListの間尾の接続はターゲオドキュメナ全体のために単一のオフィスを定義します。これは PersonListアイテムのすべての子孫が現在のコンテキストの他のオフィスではなく "Nanonull, Inc." オフィスのデータにアクセス可能なこと を意味します。

次はContact とPerson 間の接続です。一般的なマピングルールに従うと、各ソースContact のナガコ つのターゲナ Person を 作成します。各反復で、この接続は新規の現在のエンテキストを作成します。このナガ、子接続(first to First, last to Last)はデータ をノースから各 Person のエンテキスト内でターゲトアイテム」提供します。

Contact とPerson 間の接続をそのままこすると、マピングによ、複数のFirst、Last とDetails ノードを持ついつのPerson が作成されます。このような場合、MapForce はシッセージウイドウカに警告と提案を表示します。例えば

Messages	×
T V VA DOG 498 X	
PersonListByBranchOffice.mfd: Mapping validation successful - 0 error(s), 1 warning(s)	
🕝 🗇 д <a>Person has no input connection and will be generated only once.	
You can try to connect \triangleright <u>Contact</u> with \triangleright <u>Person</u> to resolve.	
🕞 🛄 The output component 🦉 <u>PersonList</u> has no output file name set. A default file name will be used.	
PersonListByBranchOffice.mfd: Execution successful - 0 error(s), 0 warning(s)	
	-

最後に、マピングはコーザー定義関数 LookupPerson が含まれています。 Contact とPerson 間の親接続のナタローザー定 義関数はそれぞれのPerson コンテスト内で実行されます。新規のPerson アイテムがターゲト側で作成される都度、関数が個人の Details 要素を作成するためコ呼び出されます。この関数は3つの入力・ラメーターを取ります。最初の、ラメーター(OfficeName)はマッ ピングの入力、ラメーターからデーダを読み取るように設定されています。マピング出力を変更することなくこの、ラメーターのためのName ソ ースデータはノースアイテムにより提供されることもできます。しずれにせよ、ソース値は親コンテキストからの値と同じです。内部では、ルッグアッ ブ関数におし数として取得された値を連結し単一の値を出力します。、詳細に関しては LookupPerson 関数のしくみと<u>例、ルッグアップと</u> 連結を参照してくたさい。

4.18.2.1 データベース

効率性を向上し、ハードウェアとや・トワークリソースの使用を削減するために同じマピング内で同じデータベースへの不必要な複数回の呼び 出しを回避することが奨励されます。マピングの性質から同じデータベースを複数回呼び出すことを回避できないシチュエーションが存在する 場合がありますが、以下の一般的な点を考慮してください

- データベースを一度のみ呼び出す場合、データベースを複数回呼び出すことを必要とするデータベースエンポーネトを親エンテキスト内に配置することを回避してください。これは、例えば、データベースエンポーネントを値のシーケンスを入力として受け取り、このためにシーケンス内で各アイテムのためコ呼び出しを受けるユーザー定義関数内に追加すると発生する場合かがます。ユーザー定義 関数 も参照してください。ターゲットコンポーネントについてする前にコービュンテキストにデータを集めるために変数は役に立ちます。
- データベースから値を集計する必要かある場合(count 関数を使用してレコードの数量を計算する場合など)、集計関数の出 力をcompute-when=once である箇所の変数に接続することが奨励されます。これは<u>例 テーブル行のカウナト</u>内で説明されている通りデータベースへの複数回の呼び出しを回避します。
- マピングに複数回同じデータベースコンポーネントを追加する代わりに(例えば、SQL-SELECT ステートメント、おけはストアド プロシージンなどの)一度の呼び出しでデータベースのデータを抽出することを試みてくたさい。
- 同じデータベースから複数のテーブルませまビューのデータを抽出する必要かある場合、(SQLモード)ジョインコンポーネント、またはSQL-SELECTステートメントを使用することが奨励されます。SQLSELECTステートメントを自身で作成する場合、後者

のオプションが便利です。データベースデーダを非データベースデータ、おけま異なるデータベースのデータにショインする必要がある場合、非-SQL ジョインを使用してくたさい。データ集中マンピング内で非-SQL ジョインの実行を最適化するコま、マンピングを MapForce Server Advanced Edition を使用して実行してくたさい。

 データベースからデータをフィルターする場合、通常のフィルターの代わりにSQL-WHERE コンポーネントを使用することが効率が 良いです。これは前者のコンポーネントが対応するデータベースの文法上、データベースと作業するために最適化されているためで す。

4.18.2.2 ユーザー定義関数

ユーザー定義関数(UDF)はマピング内に埋め込まれている入力、出力、および処理ロジックを定義するカスタム関数です。各ユーザー定 義関数にはWebサービスとデータベースを含むメインマピングと同じエレポーネトか含まれている可能性があります。

デフォルトではUDF かデータベースまけはWeb サービスエンポーネントを含む場合、そして、UDF への入力データが複数の値のシーケンスの場合、各入力値はUDF を呼び出し、データベースまけはWeb サービス呼び出しか結果として行われます。

これらの振る舞いは、代替法が無く入力の値の数量の回数 UDF を呼び出す必要のあるマピングの箇所で受け入れられます。

この発生を希望したい場合、UDFを構成して、入力として値のシーケンスを取得するように呼び出しを1回に制限するように構成することができます。通常これを返される前に値のセルを操作する(平均ませる総数を計算する)UDFのために行います。

同じ呼び出し内でUDF を複数の値を受け入れるように構成するコまUDF が型「インライン」ではよく「標準」の場合可能です。(詳細に 関しては、<u>ユーザー定義関数</u> チャプターを参照してくたさい。)標準関数では、「入力はシーケンスです」チェックボックスを選択することによ り入力、デメーターがシーケンスであることを指定することができます。入力、デメーターのタイトルレーをダブルクトックするとこのチェックボックスはコ ンポーネント設定で表示されます。このチェックボックスは以下に示される通り関数の呼び出しの頻度に影響を与えます:

- 入力データがシーケンス・プレーターは接続されていると、ユーザー定義関数は一度のみ呼び出されシーケンス全体がユーザー 定義関数についてもれます。
- 入力データが非---シーケンス・テメーターは接続されていると、ユーザー定義関数はシーケンス内の各単一アイテムのために それそれ呼び出されます。

サンプルとて以下のデモマピングを開いてくたさい、<マイドキュメント>WitovaWapForce2021WapForceExamples¥



上のマピングは植のセルを操作し、一度の呼び出し内で全ての入力値を必要とするUDFの典型的なケーズを表しています。具体的に はCalculate ユーザー定義関数は、XML ファイルから入力デーダを取り、最小、最大、および平均の気温をかえします。期待されるマッ ピングの出力は以下のとおりです



通常とおりターゲトコンポーネントのトップアイテム(このサンプルではYearlyStats)からマメングの実行は開始されます。このノードを作成 するココンイルターをトリガーする UDF からノースデータを取得しようと試みます。このマメング内のフィルターの役割は 2008 からの気温のみ を UDF につ なすることです。

チェックボックス「入力はシーケンスです」がUDFの入力パラメーターのために選択されています、このチェックボックスを確認する日よ Calculate 関数のタイトルドーをダブルクトックし、関数のマッピングを入力し、入力パラメーターのタイトルドーをダブルクトックします)。上記の通り「入力はシーケンスです」オテンコンは関数に入力として提供された値のシーケンスを完了し、関数は一度のみ呼び出されます。

🙁 Edit Input			×
Name: Tempera	atures		
Туре			
◯ <u>S</u> imple type (in	teger, string, etc.)		
Datatype;	~		
Complex type (Structure:	(tree structure) Temperatures.xsd	C <u>h</u> oose	Edit
<u>R</u> oot:	Temperatures/data	Ch <u>o</u> ose	
	Save structure file path relative to MFD	file	
✓ Input is required ✓ Input is a Sequen	ce		
		OK	Cancel

「入力はシーケンスです」チェックボックスが選択されていない場合、UDF はノース内の各値のケムウロ呼び出されます。この結果、最低、 最高、および平均の値が各単一の値のケムウェ計算され、正確でけない出力が生成されます。

データベースまけはWebサービスの呼び出しを含む更に複雑なUDFに適用することで、実行を最適化し、データベースまけはWebサービスへの不必要な呼び出しを回避することができます。「入力はシーケンスです」 チェックボックスは関数に入力 後に値のシーケンスに発生する事項を管理することはできません。すなわち、受信する値のシーケンスをWebサービスの入力に接続し複数回呼び出しが行われることを防くものはありません次の例を考慮してくたさい



上記のUDF は外部マピングから値のシーケンスを受け取ります。具体的には、入力パラメーターに与えられたデータはデータベースから来る ものです。入力パラメーターはオプション「入力がシーケンス」を選択しており、シーケンス全体が「つの呼び出し内で関数に提供されていま す。この関数は quantity 値を集計し、結果をWebサービスにポストします。1つのWebサービスの呼び出しか期待されます。しかし マピングが実行されるとWebサービスは正確に複数回呼び出されます。これはWebサービスのRequest入力が単一の値ではなく 値のシーケンスを受け取るからです。

問題を解決するためこWebサービスのRequest入力をsum 関数の結果に接続してくたさい。Webサービスが一度のみ呼び出されるように、関数はつの単一の値を処理します



通常 sum、count などの集計関数は単一の値を生成します。しかしなから、許可する親接続か存在する場合、値のシーケンスを 生成することができます。詳細に関しては以下を参照してください。例 親コンテキストの変更で詳細に説明されています。

4.18.2.3 例: 親コンテキストの変更

マピングエポーネントの一部は狂意のparent-context アイテムを持っている場合があります。このアイテムを使用して、エルポーネントが作動し結果的にマピングの出力を変更するようにマピングエンテキストに影響を与えることができます。任意のparent-contextを持つ コンポーネントは以下の通りです:集計関数、変数、ジョインコンポーネント。

	f_{8} count	
ļ	parent-context	rocutt
Ľ	nodes/rows	result.

親コンテキストを変更する方法に関しては次のマピングを開いてくたさい、<マイドキュメント
>\Altova\MapForce2021\MapForceExamples\Tutorial\ParentContext.mfd。



上のマピング内のノースXML では、2つのOffice ノードを含む単一のCompany が存在します。各 Office ノードは複数の Department ノードを含み、Department は複数のPerson ノードを含んでいます。 XML エディターで XML ファイルを開くと、オフィ ス内の職員の配属は以下のようになっています:

オフィス	部署	職員数
Nanonull, Inc.	管理	3
	マーケティング	2
	エンジニアトレグ	6
	IT & 技術 リポト	4
Nanonull Partners, Inc.	管理	2
	マーケティング	1
	IT & 技術 リポト	3

マピングはすべての部署の職員数を計算します。この目的のためにcore ライブラルらcount 関数を使用します。マピングをプレビューするために「出力」 タブをクリックすると、ソース XML ファイル内の総職員数である単一の値 21 か生成されます。

マピングの壮組みお以下のとおりです。

- 通常どおりターゲトコンポーネントのトップノード(このサンプルではrows)からマンングの実行は開始されます。rows への接続線は存在しません。この結果 Company(ソースコンポーネントの上のアイテム)とrows(ターゲトコンポーネントの上のアイテム)間の明示的なマンピングコンテキストが確立されます。
- ソースファイルコンの会社が存在するため関数の結果は単一の値です。

• coll ターゲホアイテムを作成するけっかに、MapForce は、全ての部署のすべてのオフィスから Person ノードを数えるけっかに、 上記の 明示的な親コンテキスト 内の count 関数を実行します。

関数のparent-context 引数によりマッピングコンテキストを変更することができます。これにより、例えば、各部署内の職員数を数えること ができます。これを行うために、以下に示される通りと本の接続線を描きます:



上のマピングでは、接続線 A lahe count 関数の親エンテキストを Department に変更します。これにより、関数は各部署内の職員数を数えることができます。重要な点は、ソース内に複数の部署が存在するけば関数は単一の結果の代わりに結果の sequence シーケンスを返します。このけばは接続線 B が存在します:結果シーケンス内の各アイテムのけばはマーゲトファイル内に新規の行が作成されます。マピングの出力は上記に従い変更されています(各部署内の職員数に結果が対応していることに注意してくたさい)

```
<rows>
    <row>
        <col1>3</col1>
    </row>
    <row>
        <col1>2</col1>
    </row>
    <row>
        <col1>6</col1>
    </row>
    <row>
        <col1>4</col1>
    </row>
    <row>
        <col1>2</col1>
    </row>
    <row>
        <col1>1</col1>
    </row>
    <row>
        <col1>3</col1>
```

</row>

現在のマルングは各部署のために行を作成することから、接続線 C とD を描くことにより、ターゲットファイル内にオフィス名と部署名もコピーすることができます:



このようして、出力で職員数だけではなく、対応するオフィス名と部署名も表示されます。

各オフィス内の職員数を数えるためコよ count 関数の親コンテキストをノース内のOffice アイテムコ接続します。



上記の接続では count 関数は各オフィスのためコ つの結果を返します。ソースファイル内 2 つのオフィスか存在するために、関数 2 つの シーケンスを返します。この結果、出力内 1 は各行かそのオフィス内の職員数を示す2 つの行か存在します:

```
<rows>

<rows

<coll>15</coll>

<coll>Nanonull, Inc.</coll>

</rows

<coll>6</coll>

<coll>6</coll>

</rows

</rows
```

4.18.3 優先コンテキスト

優先コンテキストは関数の入力パラメーターカ評価される順序を影響する方法です。優先コンテキストの設定はマンピングがデータを2つの関連しないツースからジョインする場合必要になる場合かあります。

優先コンテキストのしくみを理解するコよ、マピングの実行時入力アイテムへの接続か複数の値のシーケンスを持つ場合かあることを思い だしてください。2つの入力パラメーターを持つ関数の場合、これはMapForceがつのループが最初に処理される必要かある2つのループを 作成する必要かあることを意味します。最初に処理されるループは「外側の」ループです。例えばequal 関数は2つのパラメーターを受け取 ります: a とb。a とbの両方が値のシーケンスを受け取る場合 MapForce は以下のように処理します:

- a の個別の発生のためこ
 - b の個別の発生のために
 - a は b と等価か?

上記で明確のようにかけ各 a のエレテキスト内で評価されます。優先コレテキストは処理ロジックを変更することを許可するため、各 a は各 b のエレテキストで評価されます。すなけち、内部ループと外部ループを置き換えることが可能しないます。例

- b の個別の発生のために
 - aの個別の発生のために
 - a は b と等価か?

マピングの結果に影響を与える優先コンテキストを見てみましょう。下のマピングではconcat 関数には2つの入力パラメーターが存在します。各入力パラメーターはgenerate-sequence 関数の助にを使用して生成されるシーケンスです。最初のシーケンスは「1,2」で2番目のシーケンスは「3,4」です。



最初に、優先コンテキストを設定せずにマンピングを実行してみましょう。Concat 関数はカップのシーケンスの評価を開始し、次の順序で植を連結します:

- 1と3
- 1と4
- 2と3
- 2 2 4

これはマピングにも反映されています

<data> <value>13</value> <value>14</value> <value>23</value> <value>24</value> </data>

2番目の入力パラメーターを右クハクレ 優先コンテキストをコンテキストメニューから選択すると、優先コンテキストに設定されます。下に示されているように、優先コンテキスト入力は丸て囲まれています。



今回は2番目の入力/ デメーターカ最初に評価されます。、concat、関数は同じ値と連結されていますが、シーケンス、3,4、を最初に処理 します。結果的に、出力は以下のとおりてなります:

```
<data>

<value>13</value>

<value>23</value>

<value>14</value>

<value>24</value>

</data>
```

ここまでは、優先コンテキストの理論的な部分のみが紹介されています。実質的なシナレオニ関しては<u>例、優先コンテキストを使用してフィル</u>ターするを参照してくたさい。

4.18.3.1 例: 優先コンテキストを使用してフィルターする

関数がフィルターは接続されている場合、優先コンテキストは巻数のみではなく、フィルターの評価にも影響を与えます。下のマッピングは正確な出力を取得するために優先コンテキストの設定が必要とされる典型的なケースを説明しています。このマッピングを次のの文で見つけることができます: <マイドキュメント > Withova Wap Force 2021 Wap Force Examples ¥ Tutorial ¥Filter With Priority.mfd。

メモ このマッピングはXML コンポーネントを使用しますが、下に説明されている同じロジックがMapForce 内のEDI, JSON, など を含む他のすべてのコンポーネントに適用されます。データベースに関しては標準のフィルターを使用する代わりに <u>SQL WHERE</u> の使用が奨励されます。


このマピングの目的はArticles.xmlからデータを異なるスキーマarticledata.xmlを持つ新規のXML ファイルニピーすることです。 同時にマピングはProducts.xml ファイル内の各アーティクルの詳細をルックアップし、対応するアーティクルのレコードにごョインします。 Articles.xml 内の各レコードはNumber を持ち、Products.xml 内の各レコードはid を持っています。これらつの値が等価の場 合、他のすべての値 (Name、SinglePrice、color、size) はターゲント内の同じ row にこピーされます。

この目的はフィルターを追加することには達成することができます。各フィルターイズール条件を入力として必要とします。条件を満たす nodes/rowのみがターゲットにコピーされます。この目的のためにequal 関数がマンピング上に存在します。equal 関数はアーティクル番 号と製品 ID が両方のノースで等価化をチェックします。結果は入力としてフィルターに提供されます。true の場合 Article アイテム」おー ゲットにコピーされます。

2番目のequal 関数の2番目の入力/ラメーター上で優先コンテキストが定義されていることに注意してくたさい。このマッピングでは、優先 コンテキストは大きな違いをもたらし、優先コンテキストを設定してい場合、間違・オニマッピングの出力が結果として生成されます。

初期のマピング:優先コンテキストなし

優先コンテキストを持たれマピングロジック

- 一般的なマピングルールに従うとフィルターの条件を満たす Article と新規のrow がターゲット内で作成されます。(関数とフィ ルターを使用して) Article とrow 間の接続線はこの部分を対応します。
- フィルターは各アーティクルのために条件をチェックします。これを処理するには、全ての製品内で反復し、現在のエレテキスト内に複数の製品をもたらします。
- ターゲト側でid を作成するコよ MapForce は一般的なルールゴビレます(ソース内の核アイテムのナガ」コターゲト内でアイ テムか作成されます)。しかしなから、上記の通りProducts.xml からのすべての製品が現在のエンテキスト内に存在します。特 定の製品のid を読み取るナガの product ど他の箇所への接続線が存在しません。この結果、複数のid 要素はターゲト内 の各 Article のナガロゴ作成されます。color とsize にも同様な状況が発生します。

要約: Products.xml からのアイテムよ各製品を反復する必要があるフィルターコンテキストを有します。このため、id、color、および size 値は各ターゲナ row にノースファイル内の製品の数だけエピーされ、以下のように不正確な出力を生成します:

```
<rows>
<row>
<id>1</id>
```

```
<id>2</id>
<id>2</id>
<id>2</id>
<id>3</id>
</id>

<id>3</id>

<name>T-Shirt</name>

<color>red</color>

<color>blue</color>

<color>green</color>

<size>10</size>

<size>20</size>

<size>30</size>

<price>25</price>

</row>
```

ソリューション A: 優先コンテキストの使用

上記の問題は関数にフィルターのブール条件を計算する優先コンテキストを追加することで解決することができます。

具体的には、equal 関数の2番目の入力パラメーターが優先テキストとして選択されている場合、Products.xml からのシーケンスが優先されます。これは次のマッピングロジックに反映されます

- 各製品のためにequal 関数の入力 b が作成されます(すなわち b が優先されます)。この段階では、現在の製品の詳細はコンテキスト内に存在します。
- 各アーティクルのためにequal 関数の入力 a は条件がtrue かをチェックします。条件を満たしている場合、アーティクルの詳細を現在のエンテキストにも配置します。
- 次に、アーティクルと製品の詳細を現在のコンテキストからターゲット内の対応するアイテムニピーします。

製品のマピングロジックは正確な出力です。例

```
<rows>

<row>

<id>1</id>

<name>T-Shirt</name>

<color>red</color>

<size>10</size>

<price>25</price>

</row>
```

ソリューション B: 変数の使用

代替ソリューションとして、中間変数を使用してフィルターの条件を満たす各アーティクルと製品を同じコンテキストに置くことができます。マッピング上にデーダを一時的に保管しコンテキストを必要に応じて変更することができなけな、変数はこのようなシナリオに適しています。

このようなシナリオのナックに、マンピングにターゲトコンポーネトと同じスキーでを持つ変数を追加することができます。「挿入」メニューで「変数」をクリックして articledata.xsd スキーでをプロンプトされると構造として提供します。



上記のマピングでは次か発生します

- 優先コンテキストしまう使用されません。代わりにターゲトコンポーネトと同じ構造を持つ変数が存在します。
- いつも通り、マンピングの実行はターゲオル・トノードから開始されます。ターゲオを作成する前に、マンピングはデータを変数に収集します。
- 各製品のエンテキスト内で変数が計算されます。これが発生する理由はproduct から変数のcompute-when 入力への接続が存在するからです。
- フィルターの条件はこのナダ各製品のエンテキスト内でチェックされます。条件がtrueの場合のみ変数の構造は作成され、ターゲットについてれます。

4.18.4 複数のターゲットコンポーネント

マピングゴは複数のノースとターゲナコンポーネントか存在する場合かあります。複数のターゲナコンポーネントか存在する場合

MapForce 内では一度につつつてポーネト出力のみ Control プレビュー ボタンをクトックすることにおりパビューすることができます。他の実 行環境では(MapForce Server おくは生成されたコード) 全てのターゲナ コンポーネト がシーケンスで実行され、各コンポーネントの対応する出力が生成されます。

デフォルでは、ターゲトコンポーネントは上から下、左から右に処理されます。必要であれば、マンピングウイドウ内でターゲトコンポーネントの位置を変更することに以処理順序に影響を与えることができます。レフォンンスポイントは各コンポーネントの左角です。以下の点に注意してくたさい

- 2つのコンポーネトか同じ垂直方向の位置にある場合、左側が優先されます。
- 2つのエレポーネントが同じ水平方向の位置にある場合、上側が優先されます。
- コンポーネントか同じ位置に存在することは少なく、定義された順序を保証するが、変更不可能な一意の内部コンポーネント ID か自動的に使用されます。

この壮組みを説明するサンプルとしては、以下のデモマッピングを開いてくたさい、くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。このマピングは複数のノースと複数のターゲトコンポーネントから構成されています。マピングの一部のみが表示されています。



ルールゴ従うと、MapForce Server 内のこのマンピングと生成されたコード内のデフォルトの処理順序は上から下です。この場合 XSLT 2.0 コードを生成してチェックすることができます。

- 1. 「ファイル」メニューカら「コード生成 | XSLT 2.0」をクリックします。
- 2. プロンプトされると、生成されるコードのためのターゲットディレクトリを選択します。

生成後、ターゲトディレクトリコは複数のXSLT ファイルとDoTransform.bat ファイルは含まれています。後者は個別のライセンスを必要とする)RaptorXML Server により実行されることができます。DoTransform.bat ファイルはマッピングの定義と同じ順序で上から下 にコンポーネントを処理します。これは各変換の--output パレメーターを確認することで検証可能です。

```
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-by.xml" --xml-
validation-error-as-warning=true %* "MappingMapTogroups.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-adjacent.xml" --
xml-validation-error-as-warning=true %* "MappingMapTogroups2.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records.xml" --output="group-into-blocks.xml"
--xml-validation-error-as-warning=true %* "MappingMapTogroups3.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v2.xml" --output="group-starting-
with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups4.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v3.xml" --
output="group_ending_with.xml" --xml-validation-error-as-warning=true %* "MappingMapTogroups5.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
RaptorXML xslt --xslt-version=2 --input="records-v3.xml" --
output="group_ending_with.xml" --xml-validation-error-as-warning=true %*
"MappingMapTogroups5.xslt"
IF ERRORLEVEL 1 EXIT/B %ERRORLEVEL%
```

最後の変換はgroup-ending-with.xml と呼ばれる出力ファイルを生成します。このターゲナオンポーネントをマソングの上の部分に 移動します:



XSLT 2.0 コードをもう一度生成すると、処理順序が変更に従い変更されます:



上のコードリスティングでは、最初の呼び出しはgroup-ending-with.xml を生成します。

他のコード言語と生成された MapForceServer 実行ファイル(.mfx)内で同様に処理順序を変更することができます。

チェーンされたマピング

上記と同様の処理シーケンスがチェーンされたマンピングにも適用されますチェーンされたマンピンググループは つのユニットとしてとられます。単 ーのチェーンされたマンピングの中間ませる最終ターゲットコンポーネントの位置変更は処理シーケンスに影響を与えません。マンピング内に存 在する複数の「チェーン」ませる複数のターゲットコンポーネントのチが各グループの最初に処理される最終ターゲットコンポーネントの位置を決 定します。

- 最終ターゲナコンポーネントか同じ垂直の位置にある場合、左側が優先されます。
- 最終ターゲナコンポーネントが同じ水平の位置にある場合、上側が優先されます。
- コンポーネントが同じ位置に存在することは少なく、定義された」順序を保証するが、変更不可能な一意の内部コンポーネント ID が自動的に使用されます。

5 マッピングのデバッグ

MapForce は、MapForce 内蔵の変換言語のためのマピングデバッガーを含みます。マピングデバッガーは以下の目的を達成する手助けをします:

- •マピングにより生成された値を個別のコネクタレベルで確認して分析します。
- 特定の値を作成するなめのコンテキスト (ノードのセオ)をマングで イライトします。
- 各値をリアリタイムで処理おけま計算するかを確認するため、おけま生成されるマピング出力をプレビューするため、MapForceをステップ、イステップで実行します。
- 目的(ブレークポイント)を、マピングの実行に設定し、現在処理されている値を停止し、表示します。
- 現在の実行の箇所までのマピングの実行によ処理された値の履歴を確認します。

マナングの変換言語がBUILTHN の場合、マナングデバッガーを使用することができます。異なる言語のナックにデザインされたマナナングのデバッグを開始すると、マナピングの言語をBUILTHN に変換するようこプロンプトされます。メニューコマイ「出力 / 内蔵の実行エンジン」を選択することことは、マナビングをBUILTHN にすることにひきます。とちらの場合でも、マナングがBUILTHN 言語内(例: XSLT 関数) で使用することのできな、ロンポーネントを含む場合 BUILTHN への変換し成功します。

MapForce デバッガーは (MapForceを使用してコードを書く必要がむ サング) 従来のデバッガーと異なり、プログラムコードをラインごと コーマースしません。代わりに、 MapForce により生成されたコードの結果は、デバッガーにより表示されます。 具体的な方法は、デ バッガーは、入力と出力コネクタを介してい やされた値をログします。 ログされた値は、マッピング専用のウィンドウを介して、直接分析のため にで使用することができます。

以下のセクションは、マノピングデバッガーを使用する複数の方法をハイライトします。

ブレークポイントを使用するデバッグ

マルングの特定の箇所で、デバッグの実行を停止する必要がある場合、従来の開発環境で行われるように、ブレークポイトを設定します。相違点は、ブレークポイトはコードのラインに追加されるのではよく、マルングコンポーネントの入力ませま出カコネクタに追加されます。ブレークポイトの条件も追加することができます(これは、設定された条件が満たされると実行を停止する場合役に立ちます)。



ブレークポイント

ブークポイトを希望するコネクタ上で定義し、最初のブークポイトに到達するまでマピングを実行し、封之のブレークポイトまで マピングを継続することができます。このようして、選択されたコネクタと関連したマピングコンテキストと値を分析することができます。デ バッガーにより与えられた「ステップイン」、「ステップアウト」、「ステップオーバー」、および「最小ステップ」コマイドを使用して実行の スピードを遅くしたり速くしたりすることができます。これらのコマイドによりマピングの一部をスキップすることができ、封たの反対に、必要であ ればマピングの一部を更に詳しく実行することもできます。

ステップバイステップのデバッグ

デッシグをステップ、イステップで行い、マンピングコンテキストとそれぞれのステップ、「関連した植を分析することができます。このシナリオは前の ものと類似しており、「ステップイン」、「ステップアウト」、「ステップオーバー」および「最小ステップ」コマンド、を使用して実行のスピ ードを速くおけることができます。



値のログを分析する

MapForce にマンピングのデッジやに全てのコネクタによ処理された全ての値を記憶、履歴のトレース)するように構成することができます。履歴のトレースを保持することは、データを中心としたマンピングには、向いていないかもしれません。ですから、このオプションは、必要であれば無効化することができます。オプションが有効化されていると、各コネクタによ処理された値の現在の実行のパジションまでの、完全なログを分析することができます。値の作成の理由を理解することのできる、MapForceに特定の値に関連したマンピングを再作成することもできます。



現在の実行の箇所に関連した値のコンテキストを設定する

デバッガーがマンピングで特定の実行の位置にある場合、現在の実行の箇所に対応した過去の値のエンテキストを分析することができます。(これは、少し前の時間に戻ることに比較することができます):



コンテキストはなぜ植か計算されるかを説明します。すなわち、生成されるマンングの特定の値について説明します。コンテキストとは、 MapForce により近し過去のコンテキストとして設定することもできますが、通常現在の実行の箇所です。コンテキストが特定の値に設定されてしる場合、MapForce は関連するノードをマンングで直接ハイライトし、マンングコネクタの横にコントを表示し、デバッガーに関連したウィンドウカ(値、コンテキストとブレークポイントウィンドウ)で追加情報を表示します。

現在の実行の箇所と同じではない、マピングのコンテキストを確認した後、コンテキストを現在の場所に再設定することができます



制限

- 現在、マピングデバッガーは、Visual Studio のためのMapForce プラグイン おはEclipse ではサポートされていません。
- MapForce がマピングを実行する際、内部でコードを最適化します(例 デーダキャッチ、おけは注意の箇所で中間結果を計算す る場合)。これにより、特定のコネクダンまりブレークポイト)はデバッグで到達できない場合があります、この場合、MapForce は通知を表示します。MapForce コード最適化は、マピングに対するマピング出力(をしてこの結果、デバッガーの振る舞い)は同じ場合でも、MapForce のリースによ興なる場合があります。
- デバッガー1つのターゲットコンポーネントのナダの出力をデバッグすることができます。マンピング上に複数のターゲットコンポーネントが存在する場合、デバッガーイこと実行されるコンポーネントを選択する必要があります。
- 現在、データベーステーブルアグションデータベースコンポーネントの"全て挿入" や" If の場合更新"などでは、バッギングはサポートされません。
- ブレークポイントは次のエンティティーに追加することはできません 定数、core | position 関数、「全てコピー」 接続の子孫アイテム ベインライン ユーザー定義関数の ラメーター。

5.1 デバッガーの準備

デバッグの準備は、大きなシステムメモルの実行を必要とする通常大きなデータマッピングに対して必要とされます。大きな入力ませ出力ファイルを処理するマッピングの場合、ませま、大きなデータのコレクションを繰り返す場合などか当てしまります。

デバッグのスピードを向上し、メモリの使用量を減らすために、デバッグを開始する前に以下を行うことが奨励されます

- マピングが複合型の場合、デバッグが必要のは、マピングの一部の接続を削除ませ、捕除してくたさい。
- マノピング大きな入力ファイルを使用する場合、小さなサイズのファイルと置き換えてください。
- 「フルトレース履歴を保持」が無効化されていることを確認してくたさい(デバッガー設定を参照)。

正確な出力をデバッグするために、以下を適用することができるか確認してくたさい

- マピングー、複数のターゲナコボーネトかある場合、ターゲナコポーネトを選択して、「プレビュー」ボタン()をクリックします。
- マピングがチェーシマピング(<u>チェーンマピング</u>を参照)の場合、中間コンポーネントの「パススルー」(→) ボタンをリレースします。「パススルー」コンポーネントのデバッグは現在サポートされていません。

オプションで、デバッガーか重要なコネクタの一部を停止する場合、値を分析するこれらのコネクタヘブレークポイントを追加します(<u>ブレークポ</u> <u>イントの追加と削除</u>を参照)。

5.2 デバッガーコマンド

次のデバッガーコマンドにアクセスすることができます

- 「デバッグ」メニュー内
- キーボードのショートカメ
- デバッグソールドー内

ᡟᠵᠵᡄ᠆ᡄᠴᢣ	キーボードショー トカット	ツールバーボ タン	説明	
デバッグ デバッグの開始	F5	ト デッジを開始ませま、ブレークポイトに到達ませまマメピン か完了するまで継続します。		
デバッグ デバッグの中止	Shift + F5	 デバッグを停止します。このコマイドはデバッグモードを終了して、MapForceを標準モードに切り替えます。 		
デバッグ ステップイン	F11	¢۶	マメピング内で単一のステップを終了するまでマメピングを実行します。マメピングでデバッガーは、シーケンスの単一アイテムを生成する独立した計算の理論的なグループです。	
			マンノノノナキストにより、シリコマントは、ケストラリックーノットに移動子/ノースの親へ移動」と解釈することができます。	
デバッグ ステップオーバー	F10	¢	現在のステップが終了するまで、おうは他のシーケンスのアイテム を終了するまで)実行を継続します。このコマンドは、現在のス テップの入力である計算をステップオーバーします。	
<u>テ</u> ᡝ᠉ᡃᠮ᠋ᡰ᠌ᡘ᠋ ᡪ ᢧᡗ ᡘ ᡃᠯ	Shift + F11	Ş	現在のステップの結果が使用されるまで、お当よ入力お当ま使 用の子ではない マテップが実行されるまで実行を継続します。こ のコマンドは現在の計算をステップアウトします。	
			マンピングコンテキストにより、このコマンドは「右へ移動/ターゲットへ移動 親/ソースの子へ移動」と解釈することができます。	
デバッグ 最小ステップ	Ctrl + F11	 +	値が生成ませま使用されるまで、実行を継続します。このコマンドは、ステップを細分化し、接続ごせご通常2回停止します:ソースか値を生成した際、ターゲトが値を使用した際。 MapForceは、マノニングの提示に従い値を計算する必要はなく、生成と使用イベントは順序に従われ、場合があります。	

🔋 🕨 🕒 🚷 🚱 🕒 🖕

デバッグツールバー

5.3 デバッグモードについて

デバックを開始すると、MapForce は(「F5」ませは「F11」、ませは「Ctrl + F11」を押してマッピンクをデバッグモードで実行します。

MapForce かデバッグモードの場合、マッピングは読み取り専用です。マッピングエリアでエッポーネントを移動させることはできますが、コマンドの多くを使用することはできません。これは、マッピングの検証、デプロイ、コードの生成、マッピングのドキュメント、マッピングエリアに新規コンポーネントを追加する、おけは既存のコンポーネントを再ロードするなどが含まれます。

デッジグモードにより特定の値を生成するコンテキストを分析することができます。この情報は、マンピング上および、値、コンテキスト、とブレー クポイント・ウインドウで直接使用することができます。デフォルトでは、これらのウインドウは、デッジグを開始すると表示され、デッジグを中止する と隠されます。

MapForce は、「Shift + F5」を使用して、(おけは「デバッグの中止」 📮 ツール ディオタンをクリックすることにより、)デバッグが中止されるまで、デバッグモードにあります(マッピング は読み取り専用です)。

以下のイメージは段階的にデバッグされるサンプルのマッピングを表しています(<マイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレクトリからの SimpleTotal.mfd) (「F11」を押すと、先に進みます)。



デバッグモード中のMapForce 開発環境

デバッグモード中にMapForce には提供される他の情報と視覚的ヒントは以下で説明されています。

マピングペイン

デバッグ中、マピングペインは追加情報を表示します

- データオーバーレイ(下を参照してくたさい)は現在の値と関連した値をコネクタの近くに表示しています。
- (コンテキストウィンドウ内で構造と表示されている現在のコンテキストは以下のようこ イライトされています:
 - コンテキスト内のコネクタ縞のマゼタで表示されています(№)。
 - 曖昧なコンテキスト内のコネクタは点線のマゼタで表示されています(…)。
- 現在の実行の場所は、緑のコネクタアイコン(🌔)で表示されています。

データオーバーレイ

各コネクタによ処理された値は、対応するコネクタの近くに小さな三角形のデータオー・トレイとて表示されています。最近選択されたオ ー、トレイは赤い枠で表示されます。前回のステップから変更された値は、暗い赤で表示されています。単純なコンテンツを持つノードには、デ ータオー、トレイは以下の2つの値を結合します - ノード名と値。ノード名が現在の実行ポジンコンの前に複数回反復されると、現在の反復 のインデックスは角かっ二内の数値で表されます。

データオーバーレイは以下の振る舞いを行います

- データオーバーレイにマウスをポイントすると、一時的に前景に常時されます。クリックすると常時前景に表示されます。おこ、クリック すると対応するコネクタを選択します。
- データオーバーレイはボラッグして移動することができます。
- データオーバーレイはエレポーネトか移動されると、移動します。ですから、データオーバーレイが重なって表示される場合はエレポー ネント同士が近すぎるけっかであり、コレポーネントをマンピングエリア内で移動してスペースを与えてください。データオーバーレイもコン ポーネントと共に移動します。
- データオーバーレイをクリックすると、値ウインドウ内に植か表示されます。
- コネクタをクリックすると、データオーバーレイを選択します。

ブレークポイント

ブークポイトは、デ・シグモード中にマッピングの実行を中断するマイルストーンを指します。この用語は、他の統合された開発環境などで 使用されているため、既知かもしれません。コードのライノニブレークポイトを追加する他の開発環境とは異なり、MapForce は、入力ま だは出力コネクタイズ接続の左おさば右側に小さな三角形で表示されるブレークポイトを追加することができます。マメピングペインでは、ブ レークポイトには赤いサークルで表示されています。定義されたブレークポイトはブレークポイトウイドウ内に表示されます。 <u>ブレークポイト</u>は赤いケークルで表示されています。定義されたブレークポイトはブレークポイトウイドウ内に表示されます。 <u>ブレークポイト</u>

現在のデバッガーのポジション

緑の三角形(🖻)はデバッガーを表しています。このポジョンはコンポーネントの入力ませは出力コネクタです。

現在処理されている値は「コンテキスト」タブの「値」ウインドウでも表示されます。

接続およびますはコネクタのセナは、縞のマゼクで現在のマピングコンテキストを表します。同じ情報が「コンテキスト」ウィドウ内で階層構造として表示されています(<u>コンテキストウィドウの使用</u>を参照)。

値のエンテキストを手動で設定する場合、現在のデバッガーのポジュンは最近の実行ポジュンに相対的な過去にあるポジュンです。最近のポジュンと過去のポジンコンの違いを明確にするために、現在のポジュンのエネクタは、デバッガーインターフェイス内で以下の色別に表示されています。

۵	緑は "現在" を意味し、現在の実行のポジェンを示します (ユネクタの現在の値を表示するを参照してくたさい)。
Þ	黄色は "過去" を意味し、現在の実行のポジョンに相対して、過去のロネクタを観察していることを示します。これは、コンテキストを手動で設定した後に発生する場合があります (コンテキストを値に設定するを参照してくたさい)。

値ウイボウ

値ウイボウは、マピングにより処理された値についての精報を提供します。現在の実行のポジション、おけはユーザーが設定した特定のエン テキストでどのマピング処理されたかについての情報が提供されます。<u>値ウイボウの使用</u>を参照してくたさい。

エレテキストウィンドウ

コンテキストウイズウは、現在のデバッガーのポジョンは関連したノードのセメと関数の階層的なビューを提供します。 コンテキストウイズ ウの使用を参照してください。

ブレークポイント ウィンドウ

ブレーケポイトは、MapForce が開始されてから作成されたデバッグブレーケポイトのJストを表示します。複数のマピングでブレーケポ イトを定義した場合、ブレーケポイトウィドウ内に全てのブレーケポイトか表示されます。 ブレーケポイトウィドウク使用を参照してく たさい。

5.4 ブレークポイントの追加と削除

ブレークポイトは、デ・シグモード中の実行でどのマピングをブレークするかを指定するマイルストーンです。作成されるブレークポイトは、す べてのマピングのナムリングロー・シリン保管され、ブレークポイト・ウィンドウで表示されます。ブレークポイトは明示的に削除されるか、 MapForce か閉じられるまで有効です。

メモ ブレークポイントは以下のエンティティーコは追加することはできません 定数、core | position 関数、「全てユピー」 接続の 子孫アイテム、 ベインライン ユーザー定義関数の ラメーター。

ブークポイトは簡単おは条件付きで作成することができます。簡単なブークポイトはマピングの実行を条件無しに停止します。条件 付ブークポインはマピングの実行を、割り当てられた条件が満たされた場合のみ停止します。条件はカスタム化された値を提供する、 MapForce ビルドインライブラリ関数の書式を取ります。すなわち、条件が true を返す場合、ブレークポイトはマピングの実行を停止 します。

ブレークポイントを作成するには、以下を行います

- 入力おけは出力コネクタコンポーネントの右おけば左横にある小さな三角形)を右クリックして、「デバッガーブレークポイント」を 選択します。
- 入力おけ出力コネクタをクリックして、「F9」を押します。

条件付ブレークポイントの作成:

1. コネクタを右クリックして、「ブレークポイントプロパティ」を選択します。

🕑 ブレークォ	イントプロパティ
場所	
コンポーネント	: 🛃 books
コネクタ:	⊳ id
	/ファイル: books.xml/books/book/id
▼ ブレークポイ	ット(B)
✓ 条件(C)	
関数:	core.greater 💌
值 :	2
	OK キャンセル
	/_

- 2. 両方の「ブレークポイント」を選択するために「条件」チェックボックスをクリックします。
- 3. リストから必要な関数を選択し、関数の値を、適用できる場合は、入力します。例 上のサンプルでは、ブレークポイントは、通過する値がとい大き、場合、マピングの実行を停止します。

条件ブレークポイトを追加するコネクタのデータ型が、関数にと期待される型と一致しない場合、MapForce は、データ 型を自動的に変換しようと試みます。自動変換が無理な場合は、マンピングの実行は、失敗します。これを回避するため に、互換性のあるデータの型を使用するようにしてくたさい。例、関数 core.starts-with は、文字列の値を期待しま す、ですからブレークポイトのコネクタは同じ型を有する必要があります。

ブークポイトの削除

ブルークポイントを削除するコよブレークポイントが存在するコネクタを右クトックして、「デバッガーブレークポイント」を選択します。また、 ブルークポイントが存在する入力または出力コネクタをクトックし、「F9」を押します。

ブレークポイトウイドウからブレークポイトを削除することもできます(ブレークポイトウィンドウの使用を参照)

到達することのできないブレークポイント

MapForce は "ブレークポイトに到達することかできません" メッセージを表示する場合かあります:



これは、以下の理由のためブレークポイトにデバッガーが到達できないことを意味します

- マピングの一部ではおいロネクタでブレークポイントが定義されている場合。
- 実行の最適化のためMapForce がブレークポイトに到達できない場合(制限を参照)。

次の定義されたブレークポイントに到達するために、続けるをクリックします。次に定義されたブレークポイントに移動する場合、デバッグの実行の終わりに移動する場合、デバッグをステップで実行する場合、ステップをクリックします。

デバッガーが直面する到達することのできないブレークポイントに関する通知を無効化することが、このメッセージを再度表示しないをクリックすることに設定できます。手順は以下のとおりです:

- 1. 「ツール」メニューから「オプション」をクトックします。
- 2. 「メッセージ」をクリックします。
- 3. 「到達することのできないブレークポイントに関する情報」チェックボックスのチェックを解除するためにチェックボックスをクトックします。

5.5 値ウィンドウの使用

値ウイボウは、デバックモード中にマッピングにより処理される値についての情報を表示します。値ウイボウに表示される情報は、現在のデバッガーのポジョン、およびクリックするユーザーインターフェイスにより異なります。値ウイボウにはよ以下のダンカ含まれます:

"コンテキスト"ダブ

「コンテキスト」タブは、現在処理されている値を表示します。(コンテキストウイドウで同じ値のコンテキストが表示されています)。これは、デバッガーの現在の実行のポジションの値、ませば過去に処理されたコネクタの値を表示します。MapForce はこれらの2つの違いを異なる 色を使用して区別します:

۲	緑は "現在" を意味し、現在の実行のドジェンを示します (ユネクタの現在の値を表示するを参照してくたさい)。
Þ	黄色は "過去" を意味し、現在の実行のドジョンに相対して、過去のコネクタを観察していることを示します。これは、コンテキストを手動で設定した後に発生する場合からます (ユンテキストを値に設定するを参照してくたさい)。

"関連した" ダブ

関連した。タブは、関連した値、おけお現在処理されている、おけは、「近い過去」の値を表示します。通常、このダブを明示的にクトックする必要があります。MapForceは自動的にデバッガーの現在の実行のポジンョンは関連したコネクタのデータオーバーレイをクトックすると切り 替えます。最近の過去にステップパックを参照してくたさい。

"シーケンズ" ダブ

存在する場合、シーケンス タブよ シーケンスを処理するコネクタの値にアクセスすることを有効化します。このタブはコネクタかアイテムのシー ケンスを処理した場合のみ表示されます(例 sum おうよ count などの 関数)。アイテムのシーケンスを処理したコネクタのデータオーバ ーレイをクトックすると、値ウィドウよ フォーマナで "n アイテム"エトリを、n がコネクタにより処理されたアイテム数である箇所で表示しま す。それぞれの値にアクセスするには、このエトリをダブルクトックします。(おうさ右クトックして、コンテキストメニューから「シーケンスの展開」 を選択します)。

Values			-	ņ	×
Context	Related	History			
values					
2 item(s)					
	Expand	Sequence			

値はシーケンスダブ内に表示されます。

Values		→ ‡ ×
Context	Related	Sequence 4 >
> values		
170		
425		

履歴ダブ

「履歴」タブは、デバッグが開始してから現在の実行のドジョンまでに処理された値を表示します。 コネクタにより処理された値の履歴を 確認するを参照してくたさい。

5.6 コンテキストウィンドウの使用

MapForce がデバッグモードの場合、コンテキストウインドウは、デバッガーの現在のポジションイン対して関連のあるコネクタの構造を表示します。すなわち、現在のマッピングの値を作成するマッピングコンテキストを与えます。

MapForce は、現在のコンテキストを以下の通り作成します:

- 1. ターゲナ構造のルートノードから開始します。
- 2. 現在のターゲオノードに降順します。
- 3. 現在のターゲトノードから、マピング内で現在のポジョンコノードするエレポーネントを左側に移動します。これらのエレポーネントは、エレポーネント、ビルドインませるユーザー関数、変数などをフィルターおよび述べ替える可能性かあります。

コンテキストウイドウは、情報およびサビゲートを助ける役割を果たします。マピング内の現在のコンテキストから直接マピング内で特定の ノードを選択するコは、コンテキストウイドウ内のノードを右クトックして、「マッピングの選択」をクトックします。これはマピングが大き、ヒスク ロールを減らすことにより役に立ちます。

コンテキストウイドウ以下の特別なアイコンと表記を表示する場合があります。



アイコン	説明
	稀に、計算された値が複数のコネクタのために使用される場合があります。この場合、複数の緑色のアイコンか表示される可能性があります。
	デバシガーか最近の実行ポストレス相対的な過去に存在する場合を表示します。これは値にコンテキストを設定すると生じる可能性があります(<u>コンテキストを値に設定する</u> を参照)。

上のアイコノコルえ、コンテキストウインドウは、マピング内に存在するコンポーネントの標準アイコンを含みます。

コンテキストウインドウとユーザー定義関数

現在のエレテキストがユーザー定義関数を含む場合、これらはエレテキストウィドウにも表示されます。現在のエレテキストがユーザー定義関数の入力の値を計算するようの場合、エレテキストは以下のように決定されます。

- 1. ターゲトからユーザー定義関数の出力コネクタ、ユーザー定義関数の入力コネクタへ。
- 2. 更に左側に。
- メモ ユーザー定義関数は コンテキスト内で複数回生じる可能性かあります。これは、関数の呼び出しの一部かチェーンされている、またはユーザー定義関数が再帰的と定義されているためです。

5.7 ブレークポイントウィンドウの使用

ブレークポイントを使用することにシリブレークポイントをグロー Vu的に表示し管理することができます。デフォルトでは、MapForce がデバッ グモードの場合、ブレークポイントウィンドウは表示されます。ブレークポイントウィンドウを常に表示するコよ「表示 | デバッグ ウィンドウ | ブ レークポイント」を選択します。

ブレーケポイトウイドウは、MapForceを開始してから作成された全てのブレークポイトをマッピングファイルごとごグループ化し表示します。MapForce が開かれていると、マッピングに関連する全てのブレークポイトはMapForce にお記憶されており、ブレークポイトウィンドウで表示されます。現在ディッグされているマッピングは標準テキストの色でブレークポイトウィンドウオで表示され、他のマッピング、閉じられているませ」は非アクティブはマッピング、は灰色で表示されています。

ブレークポイント		
名前	親	値をトレースする 条件
🐯 Complete PO.mfd		
O Amount	🛃 ShortPO	
O Customer	🛃 Customers	
🐯 DB_UserList.mfd		
O Username	🖯 Accounts	
🗟 GroupTemperaturesByYear.mfd		
🛑 data	😫 Temperatures	<data>[1]</data>

ブレークポイントウインドウ内の全てのマンピング、おけまでの中のブレークポイント)をダブルクリックすることで素早く開くことができます。

メモ MapForceを閉じる、おけ、再起動すると、すべてのブレークポイトイノ消除されます。

ブレークポイントに関する情報は下の表にまとめられています。

カラム	説明
名前	ブレークポイントが属するノードの名前
親	ブークポイトが属するマピングコンポーネトの名前
トレ—スする 値	ブレークポイントで接続をパススレーする値。トレースする値は、値デバッグの実行中に表示されます。
条件	ブレークポイトに条件がある場合、この列はブレークポイトの条件を表示します。

ブレークポイト次のアイコンス関連している可能性があります。

アイコン	説明
•	アクティブンブレークポイト。現在デバッグされているブレークポイントをマッピングから表します。
0	非アクティブなブレークポイト。現在ディッグされていない、開かれているブレークポイトをマピングから表します。
Q	到達することのできな、ブレークポイト。ディッガーにより別達することのできな、ブレークポイトを表します。
0	条件付のブレークポイント。条件付のブレークポイントを表します。

ブレークポイントのプロパティを表示または変更する方法:

• 右クトックして、コンテキストメニューから「ブレークポイントプロパティ」を選択します

ブレークポイントを削除する方法。:

- 削除するブレークポイントを右クリックして、コンテキストメニューから「ブレークポイントの削除」を選択します。
- ブレークポイントをクリックして、「削除」を押します。

コンテキストコマンド「全てのブレークポイントを削除」は、マンピング内の存在箇所に関わらずブレークポイントウィンドウ内に表示される全てのブレークポイントを削除します。

以下を参照してくたさい、ブレークポイトの追加と削除

5.8 部分的に生成された出力のプレビュー

段階的に、おけオルークポイトを使用してデバックを行う場合、現在のデバッガーのドジョンまでの出力されたマメピングの出力を確認することができます。生成された出力の一部のプレビューはXML、flattext、およびEDIターゲトコンポーネトによりポートされています。

デフォルトでは、「F5」を押すと、「ルークポイトを定義せず」こ MapForce は、デバッグモードで全体のマッピングを実行し、「出力」 タブゴル「潜え、最終的に生成された出力を表示します。しかしながら、ブレークポイトを定義している場合、オゴはデバッグを段階的に 行っている場合 (F11 オゴは Ctrl + F11) デバッガー実行はマッピング出力が生成されているにもかかわらず停止されます。マッピング出力 が部分的に書かれている場合でも、「出力」、タブをクリックすることができ、オニ、プレビューすることができます。

Generating result C:\Users\altova\Documents\Altova\MapForce2016\MapForceExamples\CompletePO.xml...

1	xml version="1.0" encoding="UTF-8"?			
2	CompletePO xsi:noNamespaceSchemaLocation="			
	file:///C:/Users/altova/Documents/Altova/MapForce2016/MapForceExamples/CompletePO.xsd" xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance">			
3				
4	<number>3</number>			
5	<firstname>Ted</firstname>			
6	<lastname>Little</lastname>			
7	♦ <address></address>			
8	<street>Long Way</street>			
9	<city>Los-Angeles</city>			
10	<zip>34424</zip>			
11	<state>CA</state>			
12				
13	-			
14				
15	Linettem			

制限

- 現在計算されるターゲオノードは、「出力」 タブニ常に表示されているわけではありません。例 XML 属性は内部で集められ、 一度に書き込まれます。
- 出力か複数のファイルを作成する場合、現在書き込まれているファイルのみか表示されます。他の出力ファイルに切り替えると無効 化されます。

5.9 コネクタの現在の値を表示する

(ステップでデッジグを実行しているか、コネクタ上にブレークポイトカ定義されているナメリデッジガーの現在の実行のポジション()) か特定のコネクタ上にある場合、コネクタにより処理される現在の値は値ウィイドウの「コンテキスト」タブに表示されます。この値が出力に書き込まれようとしている値であり、すなわち、「現在」の値」なります。これはまたエンテキストウィイドウにエンテキストカ表示される値でもあります(コンテキストウィイドウの使用を参照)。

このシチュエーションを理解するは、 <マイドキュメント> Altova Map Force 2021 Map Force Examples ¥ デルノトリから Preserve Formatting.mfdを開きます。 ターゲナコンポーネントの Number ノードの入力コネクタをクルクし、「F9」を押して、ブレー クポイトを追加します。



次に、 F5 を押して、 デッジグを開始して、 結果全体を確認します。



イメージで表示されるように、現在のデバッガーのポジュン 🌔 (およびブレークポイント 🗣) ターゲルコンポーネントの Number ノード 上口あ ります。値ウインドウ はこのノード が値 "1"を有することを示しています (この値もマンピング上では、赤い縦じま) イライトされています)。

5.10 最近の過去にステップバックする

マピングコネクタの横のデータオー・トレイ(小さな表方形のドクス)をクリックすると、値 ウイドウは名前、とオプションで、選択されたコネクタ 「関連した値を表示します。フォーカスは現在のデバッガーのドジョンでは無く、選択されたデータオー・トレイです。このビューをデバッグの 履歴を少し戻ってビューと考えることができます。マピングは、現在のデバッガーのドジョンは関連した最後の、くつかのコネクタのすっかカー バーレイのみを表示するため、これは、「近い"過去です。関連した"データオー・トレイをクリックすると、値 ウイドウは自動的に関連した タブに切り替えます。

このシナけの説明に関しては、〈マイドキュメント〉 Wiltova Wap Force 2021 Wap Force Examples ¥ デルノトリカらマピング Preserve Formatting.mfdを開いてくたさい。

マピングを開いた後、ターゲトコンポーネントのNumberノードの横のコネクタをクリックし、「F9」を押して、ブレークポイトを追加します。 「F5」を押してディシグを開始します。ソースコンポーネントのNumberノードの横にあるデータオーィ・レイ(小さな表方形のドックス)をク リックします。



マピング内でコヤクタは、通常、複数回反復されるため、反復の現在のインデックスは各かっこ内に表示されます: <Number>「1]。また、 コヤクタは値を持ち、その値がまた等号サインの後に以下のように表示されます: <Number>「1]=1。同じ値は以下のように値ウインドウの 新規の行に表示されます。

特定の値についての追加情報が必要な場合、その値を生成したコンテキストを再作成できることに注意してくたさい(<u>コンテキストを値に設</u> 定するを参照)。

5.11 コネクタにより処理された値の履歴を確認する

オブション「トレースの履歴の全体を保管」が有効化されている場合、(現在の実行のポジションまで)そのコネクタにより処理された全ての 値の履歴をビューすることができます(デッジガー設定を参照)。

コネクタをクトックすると、履歴が表示されます。値ウィンドウの「履歴」タブをクトックします。このオペレーションはマッピングの実行の開始から、現在のディッガーのポジンョンまでの間、値を処理したコネクターン対してのみ意味があることに注意してくたさい。

このケースを説明するコよ、ブレークポイトを使用せず、始めからマピングをデバッグし、特定のコネクタによ処理されたすべての値の履歴 を確認してみましょう。最初に、〈マイドキュメント〉 Altova MapForce 2021 MapForce Examples ¥ ディレクトリからマピング Preserve Formatting.mfdを開きます。既に開かれている場合は、以下を行います:

- ブレークポイトが存在する場合、全てクリアします、ブレークポイトの追加は削除を参照)。
- Shift + F5 を押すと、現在実行中のデバッグを中止することができます。

準備ができると、「F5」を押して、新規のディシグの操作を開始します。「F5」を押すと、MapForce はマピングをディシグモードで実行し、「出力」 タブニジル替えます。「マピング」 タブをクルクして、メインマピング ウイドウ format-number 関数 のresult ノードに移動します(下のイメージでは赤で イライトされています)。最後に、値ウイドウの「履歴」 タブをクルクします。表示される値を確認してください。



上のイメージで表示されるとおり、この特定のノード(result)は全部で4つの値を処理しまた。特定の値についての追加の情報が必要な場合、その値を作成したコンテキストを再作成することができることに注意してください(コンテキストを値に設定するを参照)。

5.12 コンテキストを値に設定する

コンテキストの値に設定することは、その値を生成したマンビングコンテキストを詳細に確認する。過去にさかの説比較するアグションです。コン テキストを、関連した、タブ、シーケンス、タブ、おけよ「履歴」、タブ内の、値ウインドウに表示される値に設定することができます。「トレースの 履歴の全体を保管」オプションを有効化した場合(デバッガー設定を参照)、「履歴」タブは、現在選択されているコネクタによ処理さ れたすべての値を表示します。ですから、この場合、追加して、そのコネクタのために過去にさかの説してンテキストの値を設定することができま す。

Values		▼ ₽ ×
Context	Related History	r
result		
"00025.00"		_
"00002.30"	Set Context	
"00034.00"		
"00057.50"		

コンテキストを値に設定するコよ、以下を行います。

- 値を右クルクして、コンテキストメニューから、「コンテキストの設定」を選択します
- 値をダブルクトックします。

値をエンテキストに設定すると値を生成した状況を再作成し、値ウインドウとコンテキストウインドウを選択したコンテキストに従い作成するよめ に、MapForceはマメピングコリアをハイライトします。コンテキスト内のマメピングで使用されたビジュアルヒントのシジェンドに関してはデ バッグモードについてを参照してくたさい。コンテキスト自身についての詳細は、コンテキストウインドウの使用を参照してくたさい。

手動で設定されたコネクタは、最近の実行ポジョンでないことを示す黄色(👂)です。最近の実行ポジョンに切り替えるには(適用 できる場合)値ウィバウの「コンテキスト」タブから、「現在にリセット」ボタンをクリックします。

Values			•	ņ	x
Context	Related	History			
🜔 result					
"00025.00"					
ĺ	Reset to C	urrent			

5.13 デバッガー設定

MapForce デバッガーに適応することのできる設定へのアクセスの方法は、以下のとおりです。メニューコマンド「ツール オプション」を選択して、から「デバッガー」をクリックします。使用することのできる設定は、以下のとおりです。

値の最高ストレージの長さ

値ウイズ・ウ内で表示される文字列の長さの値を定義します(最低 15 文字)。ストレージの長さを高い値に設定すると、使用可能なシステムメモルを消耗する点に注意してくたさい。

トレースの履歴の全体を保管

MapForce に、デッジグ中マピング内の全てのエレポーネトのコネクタによ処理された全ての値を保管するようこ命令します。このオプラ ンが有効化されていると、デッジグの始めから MapForce によ処理された全ての値が、メモル内に保管され、デッジグを中止するまで、ウルン ドウ内で分析のかめに使用することができます。データ優先型マピングをデッジグする場合にこのオプションを有効化することは、デッジグの実 行を遅らせ、使用できるシステムメモレを消耗するすが奨励されません。このオプションが無効化されている場合、MapForce は、現在の実 行のポジェコム相対するノードの最も新ししトレース履歴のみを保管します。

6 データソースとターゲット

このチャプターではMapForce Professional Edition がマピングに必要なソースとターゲオコンポーネントの型について説明されています。

- XML &XML X+-7
- <u>データベース</u>
- <u>CSV とテキストファイル</u>

6.1 XML と XML スキーマ

Altova W eb サイト: 🔗 <u>XML マメピング</u>

このドキュメトの北対ニ、XMLとXMLスキーマファイルをノースませおサーゲトコンポーネントとして、使用する簡単なマンピングのサンプル について説明しました。このセクションはマンピング内でXMLコンポーネントの使用についての詳細について説明します。以下のトピックか含ま れます:

- <u>XML スキーマの生成</u>
- <u>XML コパーネト設定</u>
- <u>DTD をスキーマエンポーネントとて使用</u>
- <u>派生した XML スキーマ型</u>
- <u>QName</u>
- <u>Nil 値 / Nillable</u>
- コメトと処理命令
- <u>CData セクション</u>
- <u>ワイルドカード xs:any / xs:anyAttribute</u>
- 複数のスキーマからのデータのマージ
- <u>カスタム名前空間の宣言</u>

6.1.1 XML スキーマの生成

スキーマが乱場合、MapForce は、自動的にXML スキーマを既存のファイルをベースに生成することができます。(メニューコマ・ド「挿入 |XML スキーマ/ファイル」を使用して)マメピングエリアニスキーマを持たないXML ファイルを追加すると、次のダイアログボックスか表示されます。 す。

MapForce	8
?	選択された XML ファイルにはスキーマ参照がありません。 MapForce によりスキーマの自動生成を行いますか?
	はい(Y) いいえ(N)

「はい」をクリックして、スキーマを生成します。生成されたスキーマを保存するディレクトリを選択するようにプロンプトされます。

MapForce がXML ファイルからスキーマを生成する場合、要素/属性は、XML インスタンスドキュメトから推定されなければならず、予期しないものの可能性があります。生成されたスキーマがインスタンスデータを正確に表していることを確認することが奨励されます。

要素と属性がつ以上の名前空間に存在する場合、MapForce は個別のXML スキーマをそれそれの名前空間のために生成します。このため、複数のファイルかディスク上で作成される可能性があります。

6.1.2 XML コンポーネント設定

マピングエリアにXML コンポーネントを追加した後、コンポーネント設定ダイアログボックスから適用することのできる設定を構成することができます。コンポーネント設定 ダイアログボックスを以下の方法で開くことができます:

- マピング上のコンポーネントを選択し、「コンポーネント」メニューからプロノティ」をクリックします。
- エンポーネントへッダーをダブルクトックします。
- コポーネトへダーを右クリックして、「プロノティ」をクリックします。

😢 コンポーネント設定
コンポーネント名: ShortPO
スキーマ ファイ Jル(F) ShortPO xsd 参照(wo) 編集(エ)
入力 XML ファイル(I)
ShortPO.xml 参照(B)
出力 XML ファイル(0)
参照(R) 編集(D)
ターゲット 名前空間のプレフィックス(P):
◎ コキニフ/DTD券昭を沪加/コキニフの給対パコを使用する場合(けコノールドを空に)(A)。
▼ XML 旦 言の者さ込み(X) 値をなったいちの刑にたらった(無かけ、中市を書き込むリュカがある時に物値またけぞ
✓ 信をスークリーの主にすべく(へに気)な 出力を 皆さたの クスタルの む時に 数 信む たい プレータ 値の フォーマッティングの保持を 無効化)(G)
✓ 整形して出力(Y)
■デジタル署名を作成(ビルトイン実行のみ)(L) 署名設定(N)
作成に失敗した際には: ② 処理を中止
 ● 署名無しで処理を続ける
出力エンコーディング
Iンコーディング名(C): Unicode UTF-8 ▼
バイトオーダー: リトル エンディアン マ ロバイトオーダーマークを含む(M)
StyleVision Power Stylesheet ファイル
参照 作成
☑ min/macOccursをベースにした入力の最適化処理を有効にする(E)
✓ 全ファイルパスを MFD ファイルへの相対パスで保存(S)
ОК キャンセル

XML ユンポーネント設定 ダイアログボックス

使用することのできる設定は、以下のとおりです

<i>그ンポーネン</i> ト名	コンポーネント名はコンポーネントを作成する際に自動的に生成されます。 ですが 後に 名前を変更することが可能です。
------------------	--

	コパーネト名が自動的に生成され、インスタンスファイルをその後に選択した場合、コンパーネト名もオプィンで更新するようにMapForce は、プロンプトします。
	コンポーネント名 はスペース(例: "ソースXML ファイルン) おこフルストップ文字も含むこ とができます(例: "Orders.EDI")。コンポーネント名は スラッシュ、バックスラッシュ、コ ロン、二重引用符、行頭および沫尾スペースを含んではよりません。 全般的に、コンポー ネントの名前を変更するさいの以下の影響を考慮してくたさい
	 マピングをFlowForce Server にデプロイする場合、エレポーネント名は一意である必要があります。 コマ・ドライノに入力できる文字のみを使用することが奨励されます。コマ・ドラインとWindows 内では、それぞれの文字で異なオニエンコードを使用する場合があります。
スキーマファイル	MapForce によりデータを検証とマップするために使用された XML スキーマファイルの 名前と 文を指定します。
	スキーマファイルを変更するコよ「参照」をクトックして、新しいファイルを選択します。 XMLSpy 内のファイルを編集するコよ「編集」をクトックします。
入力 XML ファイル	MapForce がデータを読み込むXML インスタンスファイルを指定します。このフィールド はノースエンポーネトにとり重要で、最初にエンポーネトを作成した際に入力され、 XML インスタンスファイルに割り当てられます。
	ソースコンポーネント内で、インスタンスファイル名はXMLルート要素とノファレンススキー マを検出するかか、おっ選択されたスキーマーズして検証するかかに使用されます。
	ファイルのロケーションを変更するコよ「参照」新しいファイルを選択します。 XMLSpy 内のファイルを編集するコよ「編集」をクトックします。
出力 XML ファイル	MapForce かデーダを書き込むXML インスタンスファイルを指定します。このフィールド はターゲトコンポーネントにとり重要です。
	ファイルのロケーションを変更するコは、「参照」をクトックして、新しいファイルを選択します。XMLSpy 内のファイルを編集するコは、「編集」をクトックします。
ターゲット 名前空間のためのプレフィックス	ターゲナ名前空間のためこプレフィックスを入力することを許可します。プレフィックスを割り 当てる前に、ターゲナスキーマ内でターゲナ名前空間が定義されている必要がありま す。
スキーマのTD レファレシスの追加	参照されたXML スキーマファイルをXML 出力のルート要素に対して、父を追加しま す。このフィールドに入力されたスキーマの、父は、xsi:schemaLocation 属性おこ はDTD が使用される場合、DOCTYPE 宣言内の生成されたターゲト・インスタンスファ イルに書き込まれます。
	XQuery おっぱC++ 出コードを生成する場合、DTD レファレンスの追加はサポートされません。
	このフィールドに、文を入力することにより、XML インスタンスファイルにより参照されたスキ ーマファイルかどの箇所に位置するかを決定することができます。これにより、マノピングが実 行される際、出力インスタンスがマノピングのデスティネーションで検証されることを保証でき ます。また、http://アドレスや絶対及び相対/文をこのフィールドに入力することもできま す。

	このオプションを無効化することには、XML インスタンスを参照された XML スキーマま だはDTD から、切り離すことができまず、例:結果する XML 出力を基となる XML ス キーマを持たな、相手に送信する場合など)。		
XML 宣言の書き込み	このオプタンにより生成された出力からXML 宣言を表示しないようにすることができます。デフォルトでは、オプノョンが有効化されていると、XML 宣言が出力に表示されます。		
	この機能はMapForce ターゲオ 言語と実行エンジンでサポートされています。		
	ターゲット 言語 / 実行エ ンジン	出力がファイルの場合	出力が文字列の場合
	ビバイン	は、	はい
	MapForce Server	は、	はい
	XSLT, XQuery	は、	いえ
	コード ジェネレータ(C+ +、C#、Java)	は、	はい
ターゲット型に値をキャストする	 Aターゲト XML スキー型がマングで使用されるかを定義します。ますはターゲトコンポーネトにマップされる全てのデータが文字列値とて扱われるかを定義します。デフォルトでは、この設定か有効化されています。 このオプションを無効化することにはり、正確なフォーマトの値を保持することができます。 例 これは、数値内で特定の小数点を必要とするスキーマ内の、ターシファセルを満足させる場合につないたちます。 マソニング関数を使用して、数値を文字列として必要とされるフォーマト内で使用し、この文字列をターゲットにマップすることができます。 このオプションを無効化にすると、無効な値の検出も無効化されます。例 数値フィールド に文字がり入力された場合。 		
整形出力	出力 XML ドキュメントに構造化されたかり見を与えるために整形出力します。それぞれ の子ノードは、単一タブ文字のそれぞれの親のオフセルです。		
<i>出力工ンコー</i> +*	出カインスタンスファイルの次の ・ エンコード名 ・ バトオーダー ・ バトオーダー ・ バトオーダーマー デフォルトでは「新しいコンス 定義されたエンコードを持つ来 ンコアクセスし、タブを生成しま マンピングがXSLT 1.0/2.0 化しても、これらの言語はンジ	の設定を指定することを許可し ク(BOM) 文字が含まれるか ポーネントのためのデフォルト んしっこンポーネント。「ツール だす。 を生成すると、バイトオーダ・ トオーダーマークをサポートし	します: 否か。 ・のエンコード」オプション内で ノ オプション」からこのオプショ ーマーク チェックボックスを有効 は、オンダ効果におりません。

StyleVision Power スタイルシートファイ ル	このオプションにより、Altova Style Vision スタイルシートファイルを選択ませま作成することができます。このようなファイルは、XML インスタンスファイルからのデータをHTML、 RTF などの多種のレポートに適したフォーマナで出力することができます。	
	以下も参照 ユンポーネント上で相対/ 文を使用する	
min/maxOccurs をベースに入力処理 の最適化の有効化	このオプションにより、minOccursとmaxOccurs=~1~を持つ必要な属性おけよ 子要素などの1つのアイテムのみを含むシーケンスを特別に処理することか許可されます。 この場合、シーケンスの最初のアイテムか抽出され、アイテムはシーケンスとしてではな ◇直接動的な値として処理されます	
	入力データがスキーマイン対して場合、有効でない場合、エラーメッセージと共にマッピン グを停止する空のシーケンスがマッピング内で生じる可能性があります。このような無効な 入力を処理するには、このチェックボックスを無効化にします。	
MFD ファイルニ相対的なすべてのファイル パスを保存する	このオプションか有効化されると、MapForceは、エンポーホント設定ダイアログボックス に表示された、MapForce Design (.mfd)ファイルの場所に相対したファイルの文を 保存します。以下も参照: エンポーホント上で相対パンを使用する。	

6.1.3 *"スキーマ"コンポーネントと*してを DTD 使用する

MapForce 2006 SP2 以降では、ソースならびにターゲナオンポーネントにて名前空間を意識したDTD がサポートされます。名前空間 URI がDTD の「xsIns」「属性宣言から抽出され、マンピングを行うことが可能」ないます。

しかし StyleVision によ使用される DTD の様に、名前空間 URI を持たずにxmlns*属性を含む DTD も存在します。このような DTD は、MapForce で使用することができるように拡張する必要があります。 以下に示されるように DTD を修正して名前空間 URI を 含む xmlns 属性を定義する必要があります。

6.1.4 派生した XML スキーマの型

MapForce では複合型の派生型に対するマンピングがサポートされます。派生型は xsi:type 属性により特定の派生型を指定する XML スキーマの複合型です。

以下のスクレージョナは、XMLSpy にて派生型の "US-Address" の定義を示したものです。基底型(おさオレジナルとなっている複合型)はこの場合 AddressType どなります。Zip とState の2 つの要素を新たご追加することで、派生型のUS-Address が作成されます。



生成された型のサンプル(XMLSpy スキーマビュー)

次のサンプルは、派生したXMLスキーマ型からデータをマップ、お台はスキーマ型へのデータのマップの方法を説明しています。

- 1. 「挿入」メニューカら XML スキーマ/ファイル」をクリックします。以下のXML スキーマ、 <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\MFCompany.xsd を開きます
- 2. インスタンスファイルを提供するようこプロンプトされると、「スキップ」をクリックして、ルート要素をとて Company 選択します。



3. Address 要素の横のTPE ボタンをクリックします。このボタンはスキーマ内のこの要素のために存在する派生した型であることを示しています。
| C | ⑦派生型 (xsi:type) | × |
|---|--------------------------|----|
| | 個々のノードに表示する派生したスキーマ型を選択: | |
| | AddressType | * |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | - |
| | | |
| | OK キャンセ | JU |

4. 使用する派生した型の横のチェックボックスを選択し、(この場合、US-Address)、「OK」を押して確認します。新規の要素 Address xsi:type="US-Address" がンポーネントに追加されます。

🔮 MFCompany				
[□□ファイル: MFCompany.xml <u>ファイ</u>	ル/文字列 🕻			
¦¢ ⁱ ⊖ () Company Root element	L L			
🗟 🕀 🗘 Address 🔤	بر 14			
🖕 🕀 🗘 Address xsi:type="US-Add	ress ්			
P Name	ដ			
Street	۲ ۲			
City	ដ			
C Zip	ដ			
🟱 🔤 🗘 State	ដ			
🍃 🖽 🗘 Person	14 14			

これらのアイテムマネルて直接、おはUS-Addressから、マピングを行うことか可能にないます。

派生型ダイアログボックスにて複数のアイテムを選択することで、複数の派生型を追加/挿入することができます。各ノードにはそれそれ独自のxsi:type要素が与えられます。

6.1.5 QNames

MapForce はマピングの実行ランタイムでXML ファイルからデータを読み込む際にQName (修飾名) プレフィックス (<u>http://www.w3.org/TR/xml-names/#ns-qualnames</u>) を解決します。

QNames は、XML インスタンスドキュメント内の名前空間 URI を参照および省略するために使用されます。QNames には、2つの種類があります: プレフィックスを持つ、ませまプレフィックスを持たない QNames.

PrefixedName	Prefix ':'	LocalPart
UnPrefixedName		LocalPart
		LocalPart が要素 おけよ属性の名前の箇所

例: 下のJストでは <x:p/> がQName です:

- プレフィックス "x" は名前空間 "http://myCompany.com" の省略形です。
- p は ローカル ト です。

MapForce は、xpath2 | qname-related 関数 ライブラルにQName に関連した関数の一部を含みます。

6.1.6 Nil の値 / Nillable

XML スキーマ仕様に従えば nillable="true" 属性をスキーマ内の要素に対して指定することで、(コンテンソのデータ型により空のコンテン "が許されていない場合でも)コンテンンを含まれ、要素でも妥当とみなすことができるようてよります。 XML インスタンスドキュメント 内に記述 される属性の xsi:nil="true" により、要素は存在するが、コンテンソは含まれないくつまり空要素である」 ことが示されます。 このセグションはど のように、MapForce がソースとターゲナ・コンポーネント内の nil 要素を扱うかを説明します。

'xsi:nil'と'nillable'の比較

xsi:nil="true" 属性はXML インスタンスファイルにおいてのみ定義されます。

14	
15	<pre><primarykey>2</primarykey></pre>
16	<foreignkey>1</foreignkey>
17	<email>biff@amail.com</email>
18	<first>biff</first>
19	<last>bander</last>
20	<phoneext>22</phoneext>
21	<orderid xsi:nil="true"></orderid>
22	<title>IT services</title>
23	-

xsi:nil 属性はMapForce のマピングにて視覚的(かつ明示的)に示されることは無く、自動的に処理されます。通常、null 値が許容さ れたノード(xsi:nil="true" 属性が含まれるノード)は存在するが、コンテンソコは何も含まれない状態にないます。

xsi:nil 属性は、通常自動的に処理されるため、MapForce グラフィカルなマッピングで明示的に表示されません。特に "nilled" ノード(xsi:nil="true" 属性を持つ が存在する場合、そのエンテンソは存在しません。

マピングノース」ある null 値が許容された要素

マピングにて null 値を許容する(nillable な)データを XML 要素から読み取った場合、xsi:nil 属性が自動的にチェックされます。xsi:nil の値がtrue どおっている場合、エンテンソは存在したいものとして扱われます。

nillable なソース要素から単純コンテンツの、属性は含まれるか子要素を含まれ、) nillable なターゲオ要素に対して、<u>target-driven</u> 接続を作成し、xsi:nil がソース要素にセオされている場合、xsi:nil 属性がターゲオ要素にも挿入されます(例: <OrderID xsi:nil="true"/>)。

<u>全て</u> 接続を nillable なノース要素から nillable なターゲト要素へマピングして、xsi:nil がノース要素にてセトされている場合、xsi:nil 属性がターゲト要素にも挿入されます(例: <OrderID xsi:nil="true"/>)。

ソース要素のxsi:nil にtrue がセナされているかを明示的にチェックするコは、<u>is-xsi-nil</u> 関数を使用します。「nil 化」された要素に 対してtrue が返され、それ以外のノードに対してfalse が返されます。

nil 化された(存在しない)ソース要素の値を別の値で代用するコよ substitute-missing 関数を使用してくたさい。

ᄯ

 コンテンンが無くても要素ノードが実際存在するため、exists 関数をnill 化されたノース要素 は接続すると、TRUE を 返します。 xsi:nil か設定されている箇所で、multiply とconcat などの)単純型の値を期待する関数を要素に対して使用 すると、要素コンテンツが存在しなく、結果を取得することはできません。これらの関数はノースノードが存在しないかのようこ 振舞します。

マピングターゲナとしてのNillable 要素

<u>ターゲト優先</u> 接続からnilable ソース要素に単純型コンテンツ(任意の追加属性は含むか子要素を含まむ、単一の値)を持つノース 要素がxsi:nil に設定されているターゲト要素に作成する場合、MapForce はxsi:nil 属性をターゲト要素に挿入しまず(例 <OrderID xsi:nil="true"/>)。 xsi:nil= "true" 属性がXML ソース要素にセルされている 場合、通常の方法で要素コンテンパターゲト要素へマピングされます。

マピングが、子要素を持つ、複合型のnillable要素に対して行われている場合、xsi:nil属性が自動的に書き込まれることはありません。 これはマピングが行われた後で子要素が書き込まれるか分からはい為です。全てコピー接続を定義することで、ソース要素からxsi:nil属 性をコピーすることができます。

空のシーケンスおけまデータベースのNULL 値をターゲット要素へマッピングする場合、それがnillable であるかどうかにかかわらず、要素が作成されることはありません。

空のターゲト要素をxsi:nil="true" 属性がセナされた状態で(強制的に)作成するコよ、<u>set-xsi-nil</u> 関数をターゲト要素へ直 接接続してください。この方法は単純型ならびに複合型のターゲト要素に対して使用することができます。

<mark>substitute-missing-with-xsi-nil</mark> 関数を使用することで、マピングノースから値を得ることができない場合にxsi:nil をター ゲト へ挿入することができます。これはノースノードが全く存在しない場合、おけは(multiply のような)計算に対して nil 化されたノースノー ドが与えられ、結果が得られなかった場合などに使用することができます。

この関数はデータベースのNULL値からxsi:nil="true"を持つ要素を作成するために使用することができます。この関数は単純型のコンテンクを持つノードに対してのみ使用することができます。

xsi:nilを生成する関数は(if-else 関数などの)値に対して動作する関数おゴまンポーネントに ススルーされることはできません。

NULL データベースフィールドからxsi:nil ヘマピング

NULL データベースフィーリドがXML スキーマのnillable 要素へマピングされた場合、実際にデータを含んでいるフィーリドに対応したター ゲト要素だけが生成されます。NULL データベースフィーリドの要素におーゲトコンポーネトにて作成されません。

ターゲナコンポーネント内ですべての要素の作成を強制するコよアライブラルソード関数から<u>substitute-missing-with-xsi-</u> nil 関数を使用します。 田田Orders Orders



上のスクレーンショナは、どのようにsubstitute-missing-with-xsi-nil 関数が全てのデータベースフィールドのセクリニターゲオー要素を作成するかを示しています。

- ターゲナ要素内で < OrderID xsi:nil="true"/> を含むすべての不足しているNULL データベースフィールド
- データベースフィールドカターゲト要素に直接マップされる既存のデータ例: <OrderID>1</OrderID>.

データベースエンポーネントのNull フィールドを確認するコよ データベース クエリ ポタンをクトックして、データベース テーブルコオしてクエレを 実行します。Null フィールドは「結果」 ウィンドウ内で「NULL」と表示されます。

	EMail •	First 🔹	Last •	PhoneExt •	OrderlD 🔹	Title
1	v.callaby@nanonull.com	Vernon	Callaby	582	[NULL]	Office Manager
2	f.further@nanonull.com	Frank	Further	471	[NULL]	Accounts Recei
3	l.matise@nanonull.com	Loby	Matise	963	1	Accounting Man
4	j.firstbread@nanonull.com	Joe	Firstbread	621	[NULL]	Marketing Manag

xsi:nilをNULLデータベースフィールドにマップする

nill 化されたXML 要素をデータベース列にマップするコよ、MapForce はNULL 値をデータベースに書き込みます。<u>set-null</u> 関数を 使用して、データベースフィールドをNULL に無条件で設定することもできます。

6.1.7 コメントと処理命令

コメトと処理命令はターゲト XML コンポーネトに挿入することができます。処理命令は、XML ドキュメト。を更に処理するアプリケー ションコ情報をパスするためコ使用されます。コメントと処理命令は、全てコピーマップグループのためのノードのためココは定義することができない ことご注意してください。

処理命令の挿入方法:

- 1. ターゲトコンポーネト内の要素を右クトックして、コメント/処理命令を選択し、メニュー(前、後)から処理命令オプションを選択します。
- 2. 処理命令 (ターゲナ) 名をダイアログ内に入力し、「OK」を押して確認します。例: xml-stylesheet。 これによしての名前をコンポーネントツノーに追加します。



3. 以下を使用することができます。例 処理命令属性の値を与えるために定数コンポーネントを使用する。例 href="book.css" type="text/css".

灹

複数の処理命令は、ターゲトコンポーネント内の要素の前ませば後ろに追加することができます。

コメントの挿入:

1. ターゲトコンポーネト内の要素を右クトックして、コメント /処理命令を選択し、メニューからコメントオプション、前、後ろを選択します。



これは コパーネトツノーコメトノード(<!--comment())を追加します。

2. 定数コンポーネトを使用して、コメトコンテキストを与えます、おはソースノードをコメトノードに接続します。

灹

単一のターゲナノードははつのコメントのみを追加することができます。複数のコメントを作成するは、入力関数の複製を使用します。

コメント/処理命令の削除:

• 対応するノードを右クリックして、コメント/処理命令を選択し、フライアウトメニューから「削除」コメント/処理命令を選択します。

6.1.8 CDATA セクション

CDATA セクションは、通常マークアップとして解釈される文字を含むテキストのブロックをエスケープするかがに使用されます。CDATA セクションは ~<![CDATA["で開始し、 ~]]> ~ で終わります。

ターゲナノードは、CDATA セクションとして受信される入力データを書き込むことができます。ターゲナノードコンポーネントは以下であることができます。

- XMLデータ
- データベースフィールド内に埋め込まれたXML データ
- XBRL ターゲナ内の型指定されたディメンションのXML 子要素

CDATA セクションの作成:

1. CDATA セクションとして定義するターゲホノードを右クトックして、 "CDATA セクションとしてエンテンンを書き込む"を選択します。



入力データは、区切り記号 ']]> で終了された CDATA セクションを含まない 用に警告プロンプトを表示します。「OK」をクリック してプロンプトを閉じます。

下に表示される[C..アイコンでは、要素タグがこのノードがCDATAセクションとして定義されていることを示しています。

O Description

灹

CDATA セクションは、複製されたノード、および、xsi:type ノード上でも定義されることができます。

例

… MapForceExamples フォルダー内で使用することのできるHTMLinCDATA.mfd マパングファイルは CDATA セクション か有効利用できる例を表示しています。

この例

- 太字の開始()と終了()タグはTrademark ソース要素のコンテンソーに追加することができます。
- イタリックの開始(<i>)と終了(</>/>//シ)タグはKeyword ソース要素のエンテンソニ追加することができます。
- 結果のデータは、サブセグション要素コネクタがソース優先(複合型コンテンツ)ノードとして定義されているかな、ソースドキュメント 内に表示されるかがに複製 text()ノードにつびされます。
- 複合型ノードの出力は、CDATA セクションとして定義されている ShortInfo ターゲオ コンポーネント内の詳細 ノードに なさ れます。



「出力」ボタンをクリックすることにより、マークアップされたテキストを含むCDATA セクションの表示されます。

7	⇔ <info></info>		
8	<title>MapForce</title>		
9	<description><[CDATA[Altova MapForce 2014 Enterprise Edition is the premier <i>XML</i></description>		
	/ <i>database</i> / <i>flat file</i> / <i>EDI</i> data mapping tool that auto-generates mapping code in		
	<i>XSLT</i> 1.0/2.0, <i>XQuery</i> , <i>Java</i> , <i>C++</i> and <i>C#</i> . It is the definitive tool for		
data integration and information leverage.]]>			
10	-		

6.1.9 ワイルドカード - xs:any / xs:anyAttribute

ワイルドカード xs:any (および xs:any Attribute) は、すべての要素/属性をスキーマから使用することを許可します。スクレーンショナは XMLSpy のスキーマビュー内の "any" 要素を表示しています。



MapForce 内では選択の変更(excell-compicon)ボンがxs:any(おはxs:anyAttribute)要素の横に表示されます。



クトックすると、選択の変更ボタン excell-compicon が開かれ、パワイルドカードの選択、ダイアログボックスが開かれます。エトリは、現在のスキーマ内で宣言されているグロー・ジレな要素と属性を表示します。

	Elements	Namespace
	Address	
	Altova	
•	Department	
	Office	
	Person	
	elements or attributes from a different (schema, dick Import a different schema

クリックすると、選択の変更ボタン excell compicon が開かれ、 パワイルドカードの選択 ダイアログボックスが開かれます。エトリは、現在のスキーマ内で宣言されてい るグロー・ シルな要素と属性を表示します。

🕀 🗘 Person	
💳 xs:anyAttribute 🔚	
PrimaryKey	
ForeignKey	
{>EMail	
🕻 🔚 🔚 Annie	
🕀 🕻 Department (xs:any)	
First	
()Last	
PhoneExt	
Title	

.これらのノードから、ませまノードへ、他の要素と同様にマップすることができます。

コポーネト上では、ワイルドカード要素、おけば属性は、名前に追加されている(xs:any)テキストとて識別されることができます。

ワイルドカード要素を削除するコよ選択の変更(excell-compicon)ボタンをクリックして、「ワイルドカードの選択"ダイアログボックスから選択の解除を行います。

ワイルドカードと動的なノード名

ワイルドカードから、おイオフイルドカードへのマンピングは、通常、使用することのできる、要素おイオ属性がインスタンス内に表示されており、 XML インスタンスがコンポーネントのXML スキーマインと「定言されている場合(おイオ、外部のスキーマからインポートされている場合)」に適し ています。しかしなから、インスタンス内に表示される要素おイオ属性が多すぎ、スキーマ内で宣言されるロゴよ多すぎる場合があります。 <message>の子要素の数が任意である次のインスタンスの場合を考慮してみてくたさい

上記のようなシチュエーションでは、ノード名への動的なアクセスがフイルドカードの使用よりも適しています(ノード名のマノピングを参照)。

異なるスキーマからワイルドカードとして要素を追加する

コンポーネントに割り当てられた要素以外のスキーマからの要素は、ワイルドカードとして使用することができます。このような要素をコンポーネント上で表示するコス、「ワイルドカードの選択、ダイアログボックス上の異なるスキーマのインポートポタンをクリックします。これにより以下の2つのオプションを持つ新規ダイアログボックスか開かれます:

1.スキーマのインポート

2.ラッパースキーマの生成

例えば、下のイメージは、HasExpenses.xsdとう外部スキーマを現在のスキーマにインポートしようと試みると何か起こるかを表しています。

スキーマのインポート
選択したスキーマ C:¥Users¥t.a.yasaka¥Documents¥Altova¥MapForce2016¥MapForceExamples¥HasExpenses.xsd を現在のコンポーネントにインポートの詳細の指定します。 スキーマのインポート(I)
選択したスキーマ C:¥Users¥ ¥Documents¥Altova¥MapForce2016¥MapForceExamples¥HasExpe をコンポーネントの現在のスキーマにインポートします。 ラッパー スキーマを生成する (W)
新しいスキーマ ファイルを生成します。このラッパー スキーマはコンポーネントの現在のス キーマのみを含み選択したスキーマ C:¥Users¥ ¥Documents¥Altova¥MapForce2016¥MapForceExamples¥HasExpe をインポートします。その後この新しいラッパー スキーマは現在のコンポーネントに設定さ れます。
キャンセル

スキーマのクポトオプションは外部スキーマをコンポーネトに割り当てられている現在のスキーマにクポトします。このオプションはディスク上のコンポーネントの既存のスキーマをオーバーライドすることに注意してください。現在のスキーマがディスクではなくURLにより開かれたとたートのスキーマである場合、変更することはできません(URLからコンポーネントを追加するを参照してくたさい)。この場合、ラッパースキーマの生成オプションを使用してくたさい。

ラットースキーマの生成オプションは "ラット" スキーマと ら名前の新し マキーマファイルを作成します。このオプションの利点は、エンテンソの 既存のスキーマが変更されない点です。 代わりに、既存のスキーマとインポートされるスキーマの両方を含む、ラットースキーマとう)新し マキ ーマが作成されます。 このオプションを選択すると、 ラットースキーマを保存する場所を問われます。 デンオルトでは、 ラットースキーマは somefile-wrapper.xsd. という書式の名前を持ちます。 ラットースキーマを保存した後、 デンオルトでエンポーネントに書い当てられ、 ダイアロ グボックスがプロンプトします:



「はい」をクリックして、前のスキーマに戻します。それ以外の場合は「いいえ」をクリックして新規に作成された、コンポーネトに割り当てられたラッパースキーでを保持するためはいえをクリックします。

6.1.10 複数のスキーマからのデータのマージ

このサンプルは複数のファイルを単一のファイルにマージする方法にこして説明しています。具体的には、異なるスキーマを持つ複数のノースコンポーネントをターゲルスキーマにマージします。同一のスキーマをベースにした複数のファイルをマージする方法にこしては、<u>複数の入力また</u> 「出力ファイルを動的に処理を参照くたさい。

CompletePO.mfd ファイルは ...、MapForceExamples フォルダー内にあり、3 つのファイルを1 つのXML ファイルヘマージする方法を表示しています。



単一のターゲナ XML ファイル Complete PO.xml に複数のノースコンポーネトデータか組み合わされていることに注意してくたさい。

- ShortPO はXML インスタンスファイルは関連付けられたスキーマで、番号と数量などの顧客番号と品物に関するデータのみが 含まれています。(このファイルは「顕客一人だけが収められています。)
- Customers はXML インスタンスファイルコ関連付けられたスキーマで、顧客番号と、Name やAddress とった顧客に関する詳細情報が収められています。
- Articles はXML インスタンスファイルが関連付けられたスキーマで、品物名やその数、単価といた品物に関する情報が収められています。
- CompletePO はインスタンスファイルを持たないスキーマファイルです。全てのデータはつのXML インスタンスファイルとり与えられています。このファイルの階層構造により、XML データのマージと出力を行うことが可能しています。

このスキーマファイルはXMLSpy のような XML エディターにより作成されており、 MapForce から生成されたものではありません (CompletePO.xml インスタンスファイルが存在する場合、スキーマファイルの生成を行うことも可能です)。

CompletePO の構造はメースとなる XML ファイルの構造を組み合わせたものとなります。

filter エンポーネト(Customer)を使用することで、ShortPOとCustomer XML ファイルで顧客番号が一致するデーを検索することができ、関連するデーをターゲトのCompletePOエンポーネトへ渡すことができます。

- equal 関数に以、ShortPO のCustomerNr がCustomer 以下にある Number 出較されます。
- ShortPO はは1 つの顧客情報(番号が3)しか含まれていないかめ、顧客番号が3 どなっている顧客の情報だけがウィルターコンポーネントには彼されます。
- フィルターコンポーネトのnode/row 、ラメーターはbool、ラメーターがtrue の時にCustomer データを "on-true" ノード へ渡します。つまり、顧客番号が3の時だ1出力が行われます。
- 顧客と品物データの残りは、他にある2 つのフィルターコンポーネトによりターゲナスキーママン いされます。

6.1.11 カスタム名前空間の宣言

デフォルトでは、マンピングがXML出力を生成する際に、各要素と属性の名前空間(おけよ名前空間のセナ)か自動的に MapForceによ<u>ルーゲトコンポーネント</u>と関連するスキーマから生成されます。これは、MapForce内のデフォルトの振る舞いで、XML 出力の生成を含むマンピングのンナノオの大半に適切です。

しかしなから、結果のXML出力内で要素の名前空間の管理を更に行う場合、マメピングから要素の名前空間を手動で宣言することが 必要な可能性かあります。

この壮組みを理解するコま、 <マイドキュメント>¥AltovaWapForce2021WapForceExamples¥Tutorial¥内にある BooksToLibrary.mfd マピングを開きます。library ノードを右クリックし、コンテキストメニューから「名前空間の追加」を選択します。

賃 library 甲ロファイル: library.xml>				
	前に入力を複製する(B)			
► C p	後に入力を複製する(I)			
	複製を削除(U)			
ŏ	上へ(0)			
	下へ(V)			
	名前空間の追加(N)			

library ノードの下で次の2つの新しいードが使用することができるようこなっていることに注意してください namespace とprefix.



マピングからの文字列の値をマップすることができます。下のイメージでは、名前空間「altova.library」とprefix「lib」を与える、2つの定数が「挿入 | 定数」メニューコマンドを使用して)定義されています:



生成された出力内の結果としては xmlns:<prefix>="<namespace>" 属性が <prefix> と<namespace> がマピングこの場合は定数)から来る値である箇所の要素に追加されます。生成された出力は、以下のようことはます(ハイライトされている部分に注意してくたさい。



メモ カスタム名前空間の宣言(と「名前空間の追加」コマイ)は、ターゲト XML コンポーネトに取り有益であり、要素のみに 適用することができます。属性に対して、「名前空間の追加」コマイドは、属性とフイルドカードノードに対しては使用することが てきません。ノードの子への動的なアクセスが有効化されている場合でも、おゴゴ 全てをコピー」接続を使用してノードがデータを 受け取るノードの場合でも使用することはできません(ノード名のマッピングを参照してくたさい)。

必要に応じて、複数の名前空間を同じ要素のために宣言することができます。これを行うはは、ノードをもう一度右クトックし、コンテキストメニューから「名前空間の追加」を選択します。新規のプレフィックスと名前空間の値に接続することのできる名前空間とプレフィックスノードの新規のペアを使用できるようてよります。

以前に追加された名前空間の宣言を削除するコは、ns:namespaceノードを右クリックし、コンテキストメニューから「名前空間の削除」を選択します。

空の値が与えられる場合でも、namespaceとprefix入力コネクタの両方がマップされている必要があります。

(xmlns="mydefaultnamespace"のフォーマナの)デフォルの名前空間を宣言する場合、prefix に対して空の文字列をマップします。このケースをマップ上で確認する場合は、2番目の定数が空の文字列になるように上のサンプルマングを編集します。



結果の出力は以下のようてなります

```
<?xml version="1.0" encoding="UTF-8"?>
<library xmlns="altova.library" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="library.xsd">
...
```

属性の名前のためこしフィックスを作成する場合、例えば <number prod:id="prod557">557</number> 、ノードの属性への動的なアクセスを有効化する、おけま、number> のためこprod:id 属性を持つようにスキーでを編集することに以属性の名前のため にプレフィックスを作成することができます(次を参照してくたさい)ノード名のマッピング)。

6.2 MapForce とデータベース

Altova W eb サイト: 🔗 <u>データベースマメピング</u>

MapForce 2021 ではデータベースから、XML、フラオファイルと他のデータベースフォーマナへのマピングがサポートされています。 MapForce Enterprise エディンコンでは、追加してデータベースをEDIフォーマナ、Excel 2007+、JSON、XBRLとWebサービ スへのマピングがサポートされます。

次のデータベースはサポートされています。各データベースの使用できるレートオブンゴントもリストされています。Altova しお他の ODBC /ADO データベースへのサポートの向上のナダの努力をしていますが、以下にリストされたデータベースのみで接続とデータ処理のテス トロス功しています。使用中のAltova アプリケーションが、64 ビット バージョンの場合、接続されているデータベース特定の64 ビットデー タベース ドライバー ニアクセスしてください。Altova アプリケーションはIBM iSeries データベースのロジカルファイルをサポートし、ロジカル ファイルを表示します。

7-9 ~	<i>大</i> モ
Firebird 2.5.4, 3.0	
IBM DB2 8.x, 9.1, 9.5, 9.7, 10.1, 10.5	
IBM Db2 for i 6.1, 7.1, 7.2, 7.3, 7.4	ロジカルファイルはサポートされ、ビューとして表示されています。
IBM Informix 11.70, 12.10, 14.10	Informix はADO、JDBC おば ODBC を介した接続をサポートします。 実装は、すべてのコード生成言語による大きなオブジェクト データ型をサポートし ません。MapForce は、(コード生成中) これらのデータ型が使用された場合エ ラーメッセージを表示します。
MariaDB 10.2, 10.3, 10.4, 10.5	
Microsoft Access 2003, 2007, 2010, 2013, 2016, 2019	ドキュントの作成時(2019年9月初旬) Access 2019のためご使用可 能な Microsoft Access Runtime が存在しませんでした。Microsoft Access 2016 Runtime がインストールされており、データベースが「大量」の データ型を使用しては、場合のみ Altova 製品から Access 2019 にデータベ ース接続することができます。
Microsoft Azure SQL Database	SQL Server 2016 codebase
Microsoft SQL Server 2005, 2008, 2012, 2014, 2016, 2017, 2019 Microsoft SQL Server on Linux	
MySQL 5.0, 5.1, 5.5, 5.6, 5.7, 8.0	
Oracle 9i, 10g, 11g, 12c, 18, 19	
PostgreSQL 8.0, 8.1, 8.2, 8.3, 9.0.10, 9.1.6, 9.2.1, 9.4, 9.5, 9.6, 10, 11, 12.1	PostgreSQL 接続は、ODBC おけよJDBC などのインターフェイス(ドライ ハシによりネイティブム・ライバ・・ベースの接続でサポートされています。ネイティブな 接続はボライルを必要としません。
Progress OpenEdge 11.6	
SQLite 3.33.0	SQLite 接続は、ネイティブとしてサポートされます。SQLite データベースファイ ルへの直接の接続です。

データベース	た
Sybase ASE 15, 16	
Teradata 16	ADO.NET、JDBC、およびODBCを介した接続かりポートされています。
	マピングがデータをデータベースに挿入する場合、データベースにより生成される 識別フィールドはサポートされていません。

多種の実行環境内でのデータベースマピング

マルングからプログラムコードを生成する場合、おけよMapForce Server 実行可能ファイルにマルングをエルパルする場合、 FlowForce Server にマルングをデプロイする場合、生成されたファイルと共に保存された・データベース接続の詳細は、対応するドライバーによい適用、おけよ選択されたターゲオ環境によりサポートされています。例えば、マルング 変換言語が Java に設定されている場合、 ADO 接続は、マルングから Java コードが生成されると、JDBC に変換されます。

MapForce 以外の環境でマンピングが実行されると、データベース接続の詳細がマンピングを実行するマンイン適合していることを確認してく たさい(例えば、データベースドライバーがインストールされている、データベースのいなが正確である、データベースサーバーへのアクセスかある など)。

データベースへの接続の種類の一部は、下のテーブルで示されているように、ターゲオ環境の一部でサポートされていません。

接続の型/実行環 境	C#	C++	Java	W indows 上の MapForce Server	Linux <i>/</i> Mac 上の MapForce Server
ADO	ADO ブルジ	現状のまま	JDBC に変換され ます	現状のまま	JDBC に変換され ます
ADO.NET	現状のまま	ユ ーザ- 定義	JDBC に変換され ます	現状のまま	JDBC に変換され ます
JDBC	ユ ーザ- 定義	ユ ーザ- 定義	現状のまま	現状のまま	現状のまま
ODBC	ODBC ブルジ	ODBC ブルジ	JDBC に変換され ます	現状のまま	JDBC に変換され ます
ネイティブ PostgreSQL	サポート無し	サポート無し	サポート無し	現状のまま	現状のまま
ネイティブ SQLite	サポート無し	サポート無し	ザポーキ無し	現状のまま	現状のまま

テーブル内のレジェンド

- "現状のまま"は、データベース接続型(例えば、JDBC、ODBC)が、MapForceで定義されたままとい意味です。
- "JDBC に変換されます"は、データベース接続が、JDBC のようなデータベース接続 URL に変換されるとい意味です。
- "ADO ブリジン"、おさは、ODBC ブリシジンは、接続文字列は、MapForce内で定義されたままですが、生成されるコードは、ADO ブリシジ(、おさはODBC ブリシジ)とての役割を果たす適切なクラスを使用するという意味です。例えば、System.Data.OleDb.OleDbConnection、おさはSystem.Data.Odbc.OdbcConnection等かありたれます。

 "ユーザー定義"は、生成されたコード内で接続が作動するためには、手動で接続の情報をデータベースエレポーネト設定 ダイア ログボックスこ入力するとう意味です。場合により異なりますが、接続の詳細はADO/OLE-DB-specific settings、また は JDBC-specific settings に入力される必要があります。

次も参照してくたさい

- サーバー実行のためにマパングを準備する
- マピングをMapForce Server 実行可能ファイルコンパルする
- FlowForce Server にマピングをデプロイする
- <u>コードジェネレーター</u>

6.2.1 データソースへの接続

最も簡単なケースとして、データベースは、Microsoft Access、おけよ、SQLite データベースファイルなどのローカルファイル的挙げわれます。高度なケースの例としては、データベースは、アプリケーションが接続し、データで消費するオペレーティングシステムと同じオペレーティングシ ステムを使用するとは限らない、リモート、おけよ、やトワークデータベースサーバーご存在します。例えば、MapForce が、Windows で作動し、データ(例えばMySQL) ヘアクセスするデータベースがLinux マシーンで作動することは可能です。

多種のデータベース型とJモートおよびローカルで作業するコよ、MapForceは、データ接続インターフェイスとオペレーティングシステム内で使用することのできる、まけま、定期的にメジャーなデータベースペンダーによりリレースされるデータベースドライバーに従います。常に進化するデータベース技術の景観の中、このアプローチーは、クロスプラナトフォームに対する柔軟性および相互運用性に対応します。すなオキ、MapForceにより、以下のデータアクセス技術により主要なデータベースにアクセスすることができます

次の図は、最も簡単な方法で、(ジェネトックなクライア・トアプリケーションとして表示されている) MapForce と(データベースサーバー、また は、データベースファイルであることができるデータ保管の間で使用することのできるデータ接続のオプションを示しています。



* 直接なネイティブな接続 は、SQLite とPostgreSQL データベースのためにサポートされています。使用中のシステムにインストールするために追加ドライバーは必要ではありません。

上の図で示されるように、MapForce は、主なデータベース型に次のアクセス技術を使用してアクセスすることができます。

- 基となるOLE DB (Object Linking and Embedding, Database) プロ・イダーを使用するADO (Microsoft® ActiveX® Data Objects)
- ADO.NET(データとのインタアウションを有効化するMicrosoft.NET Framework内で使用することのできるライブラルのセット)
- JDBC (Java データベース接続)
- ODBC (Open データベース接続)

ADO.NET の一部はサポートされていない、ませま、サポートか制限されている場合かあります。次を参照してくたさい: ADO.NET サポートメモ。

選択するデータベースへの接続のインターフェイスは、既存のノフトウェアのインフラストラクチャイによります。データのアクセス技術と、接続するデ ータベースと密に統合することのできる、データベースドライバーを通常選択します。例えば、Microsoft Access 2013 データベースは接続 するコよ Microsoft Office Access データベース Engine OLE DB プロバイダー などのネィティブなプロバイダを使用する ADO 接続文字列を作成します。「注接続するコよ、他方、最新のJDBC、ODBC、 おコよ ADO.NET インターフェイスを Oracle Web サ イトからダウンロードします。

Windows 製品 (Microsoft Access おは SQL Serverなど)のオダのドライバーは、Windows オペーティングシステムで使用可能ですが、他のデーダベースの種類では使用することができない可能性があります。メジャーなデータベースペンダーは、対応するデーダベースに ADO、ADO.NET、ODBC、おはよJDBCの組み合わせからクロスプラナトフォームアクセスをすることができる、一般使用可能なデーダベースでライア・トソストウェアビデライバーを定期的にリリースします。更に、上記の技術のオダのサードパーティーのドライバーを使用することができます。多くの場合は、1つ以上の方法で必要なデータベースへ使用中のオペレーティングシステムから接続することができます、この結果、MapForceからも接続することができます。使用可能な機能、パフォーマンス・ラメーター、と既知の問題はデータアクセス技術および使用されるドライバーにおい異ないます。

6.2.1.1 接続ウィザードの開始

MapForce はボータノースをセットアップするために必要とするステップを導くウィザードが搭載されています。

ウィザードのステップを開始する前に、データベースの一部の種類は、データベースドライバーまけはデータベースクライア・トソフトウェアなどの複数のデータベースの必須条件を個別にインストールまけは構成する必要があることに注意してください。これらは、通常各データベースのベンダ ーにより提供され、特定のWindows バージョン用に調整されたドキュメントを含みます。データベースの種類別にグループ化されたデータベ ースドライバーのリストに関しては、をデータベースドライバーの概要参照してくたさい。

データベースをソースまたはコンポーネントをマッピングに追加するには以下を行ってください

• 挿入メニューからデータベースをクリックしくたさい。

🕑 データベースを選払	R	×
接続ウィザード	接続ウィザード	
	ソース データペースを選択し、[次へ]をクリックしてください。	
	使用したいデータベースのベンダーが下のリストにない場合、ADOまたは ODBCを使用してコネクションを作成してください。	
既存の接続 ADO接続	Microsoft Access (ADO)(A) Progress (ODBC) Oracle (ODBC/JDBC)(O)	
DDBC 接続	 MySQL (ODBC/(M) IBM DB2 (ODBC/JDBC)(D) IBM Informix (ODBC / JDBC)(I) Sybase (ODBC)(Y) 	
) JDBC 接続	 PostgreSQL Firebird (ODBC / JDBC)(F) SQLite(L) 	
COUNT ADO.NET 接続		
グローバル リソー ス		
POSTGRE	□ ウィザードの 構成をスキップ	
PostgreSQL 接続	次へ(N) > 閉じ;	₹(L)

データベースを再利用可能なグローバリリソースとして追加するには以下を行ってください

- 1. ツールメニューから [グローバリリノース]をクリックします。
- 2. 「追加」をクリックして、データベースをクリックします。
- 3. [データベースの選択]をクルクします。

データベース型を選択し「次へ」をクトックするとデータベースの種類、技術(ADO, ADO.NET, ODBC, JDBC)と使用されるドライバーにより異なる命令が画面に表示されます。

例えば、各データベースの種類に適用できるサンプルは<u>データベース接続のサンプル</u>のセクションを参照してくたさい。各データベースへのアクセ ス技術に関しては以下のトピックを参照してくたさい

- <u>ADO 接続の設定</u>
- <u>ADO.NET 接続の設定</u>
- <u>ODBC 接続の設定</u>
- JDBC 接続の設定

6.2.1.2 データベースドライバー概要

以下のテーブルは、特定のデータベースに特定のデータアクセス技術を使用して接続するオークによく使用されるデータベースのドライバーのノスト です。このノストは、完全まず」は規範的なものではありません。以下に表示されているドライバン加え他のネィティブます」はサードパーティー代替 を使用することができます。

Windows オペレーティングシステムの複数のデータベースドライバーか既に使用されている場合かありますが、代替ドライバーをダウンロードする必要かあります。データベースの一部に対して、オペレーティングシステムに対随するドライバーより、ビデータベースペンダーより提供される最新のドライバーの、ウォーマンスの方か良い場合かあります。

データベースのペンダーは、個別にダウンロードすることのできる。ッケージ、おけは、ハンドルされたデータベースクライアントックトウェアを提供します。後者の場合、データベースクライアントックトウェアは、通常、データベースの管理と接続、およびデータベースクライアントとそのエンポーネントのインストールと構成を簡素化するなりに管理と構成ユーティノティーには構成されています。

データベースクライアントを正確に構成することは、データベースへの接続を成功させるために必須です。データベースをインストーリルて、使用 する前に、各データベースバージョンおよび各Windows バージョンにより通常異なるデータベースのインストーリにおい構成の手順を確認す ることが奨励されます。

データベースの種類に関して各データアクセステクノロジーの機能と制限を理解するコよ、特定のデータベース製品のドキュメントを参照し、特定の環境に対しての接続をテストします。よく発生する接続問題を回避するために、以下に注意してくたさい。:

- ADO.NET の一部のプロレイダーはサポートされない、ませよ、サポートが制限されています。次を参照してくたさい、ADO.NET サポートメモ。
- データベースドライバーをインストールする際、Altova アプリケーション(32-ビナト おけま64-ビナト)などの同じプラナフォームが使用されることが奨励されます。例えば、32ビナトのAltova アプリケーションが64-ビナトオペレーティングシステムで使用される場合、が使用される場合、32-ビナトのドライバーをインストールし、32-ビナトのドライバーを使用して、データベースの接続をセナアップします。詳細に関しては、使用可能な ODBC ドライバーの表示 を参照してください。
- ODBC データノースをセナアップする際、データノース名 (DSN) をユーザー DSN の代わりに システム DSN として作成することが奨励されます。詳細に関しては、使用可能な ODBC ドライバーの表示 を参照してくたさい。
- •
- ターゲトデータベースがODBCを使用したMySQL おはMariaDBの場合、オプションは「影響された行の代わりに一致した行を返す」がMySQL ODBCコネクタのカーソル結果 タブで有効化されている必要があります。おは、データベース接続ウィザドを使用して接続文字列を手動で入力する場合Option=2を接続文字列に追加します。
 Dsn=mydsn; Option=2;
- When setting
- JDBC データノースをセナアップする際、JRE (Java Runtime Environment)、Java Development Kit (JDK) がインストールされている、 おっオペレーティングシステムのCLASSPATH 環境変数が構成されていることを確認してくたさい。 詳細に関しては、 JDBC 接続のセットアップを参照してくたさい。
- データベースペンダーからインストールするドライバーおよびデータベースクライア・ホソフトウェアのインストールの手順とサポートの詳細 は、インストール・シケージのドキュメントを参照してください。公式なおゴコよサードパーティデータベースドライバーの場合、特定のオペレーティングシステムでの特定のドライバーイご適用することのできる最も包括的な情報と構成手順は、通常ドライバーインストールパッケージの一部です。

データベース	インターフェイ ス	ドライバー
Firebird	ADO.NET	Firebird ADO.NET Data Provider (<u>https://www.firebirdsql.org/en/additional-downloads/</u>)
	JDBC	Firebird JDBC driver(<u>https://www.firebirdsql.org/en/jdbc-driver/</u>)

データベース	インターフェイス	ドライバー
	ODBC	Firebird ODBC driver(<u>https://www.firebirdsql.org/en/odbc-driver/</u>)
IBM DB2	ADO	IBM OLE DB Provider for DB2
ADO.NET		IBM Data Server Provider for .NET
	JDBC	IBM Data Server Driver for JDBC and SQLJ
	ODBC	IBM DB2 ODBC Driver
IBM DB2 for i	ADO	 IBM DB2 for i5/OS IBMDA400 OLE DB Provider IBM DB2 for i5/OS IBMDARLA OLE DB Provider IBM DB2 for i5/OS IBMDASQL OLE DB Provider
	ADO.NET	.NET Framework Data Provider for IBM i
	JDBC	IBM Toolbox for Java JDBC Driver
	ODBC	iSeries Access ODBC Driver
IBM Informix	ADO	IBM Informix OLE DB Provider
	JDBC	IBM Informix JDBC Driver
	ODBC	IBM Informix ODBC Driver
Microsoft Access	ADO	Microsoft Jet OLE DB ProviderMicrosoft Access Database Engine OLE DB Provider
	ADO.NET	.NET Framework Data Provider for OLE DB
	ODBC	Microsoft Access Driver
MariaDB	ADO.NET	MariaDB のようの専用の.NET コネクタが不足している場合、MySQL のようのコネク タハET を使用してくたさい (<u>https://dev.mysql.com/downloads/connector/net/</u>).
	JDBC	MariaDB Connector/J (<u>https://downloads.mariadb.org/</u>)
	ODBC	MariaDB Connector/ODBC (<u>https://downloads.mariadb.org/</u>)
Microsoft SQL Server	ADO	 Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) Microsoft OLE DB Provider for SQL Server (SQLOLEDB) SQL Server Native Client (SQLNCLI)
	ADO.NET	 .NET Framework Data Provider for SQL Server .NET Framework Data Provider for OLE DB
	JDBC	Microsoft JDBC Driver for SQL Server (<u>https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server</u>)
	ODBC	ODBC Driver for Microsoft SQL Server (<u>https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server</u>)

データベース	インターフェイ ス	ドライバー
MySQL	ADO.NET	 Connector/NET (<u>https://dev.mysql.com/downloads/connector/net/</u>)
	JDBC	Connector/J (<u>https://dev.mysql.com/downloads/connector/j/</u>)
	ODBC	Connector/ODBC (<u>https://dev.mysql.com/downloads/connector/odbc/</u>)
Oracle	ADO	Oracle Provider for OLE DBMicrosoft OLE DB Provider for Oracle
	ADO.NET	Oracle Data Provider for .NET (<u>http://www.oracle.com/technetwork/topics/dotnet/index-085163.html</u>)
	JDBC	• JDBC Thin Driver • JDBC Oracle Call Interface (OCI) Driver これらのドライバーは、通常、データベースクライアトのインストール中にインストールされます。 (Thin ドライバーではない) OCI ドライバーを使用する接続は、Oracle XML DBのエンポ ーネントを使用しています。
	ODBC	 Microsoft ODBC for Oracle Oracle ODBC Driver (通常、Oracle データベースクライアントのインストール中にインストールされます)
PostgreSQL	JDBC	PostgreSQL JDBC Driver(<u>https://jdbc.postgresql.org/download.html</u>)
	ODBC	psqIODBC(<u>https://odbc.postgresql.org/</u>)
	Native Connection	使用可能。ネイティブな接続を使用する場合、ドライバーをインストールする必要はありません。
Progress	JDBC	JDBC Connector (<u>https://www.progress.com/jdbc/openedge</u>)
Open⊏age	ODBC	ODBC Connector (<u>https://www.progress.com/odbc/openedge</u>)
SQLite	Native Connection	使用可能。ネイティブな接続を使用する場合、ドライバーをインストールする必要はありません。
Sybase	ADO	Sybase ASE OLE DB Provider
	JDBC	jConnect™ for JDBC
	ODBC	Sybase ASE ODBC Driver
Teradata	ADO.NET	.NET Data Provider for Teradata (<u>https://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata</u>)
	JDBC	Teradata JDBC Driver (<u>https://downloads.teradata.com/download/connectivity/jdbc-driver</u>)
	ODBC	Teradata ODBC Driver for Windows (<u>https://downloads.teradata.com/download/connectivity/odbc-</u> <u>driver/windows</u>)

6.2.1.3 ADO 接続の設定

Microsoft ActiveX Data Objects (ADO) はOLE DB を介して多種のデータノースにデータベースアクセステクノロジーです。OLE DB はODBC おけよJDBC への代替インターフェイスです。COM (コンポーネント オブジェクト モデル) 環境内での統一されたアクセスを 提供します。ADO は新規のADO.NET 前の デジョンであり、他のデータノースのナダンミ使用可能ですがMicrosoft Access or SQL Server などのMicrosoft ネイティブなデータベースにアクセスを可能にする方法です。

需要な点は、複数のADOプロ・イダー間から選択することができ、一部はダウムロードされワークステーションに使用する前にインストールされる必要があります。SQL Server への接続のために以下のADOプロ・イダーを使用することができます。

- Microsoft OLE DB *Driver* for SQL Server (MSOLEDBSQL)
- Microsoft OLE DB *Provider* for SQL Server (SQLOLEDB)
- SQL Server Native Client (SQLNCLI)

上記でリストされナプロ・イダーで奨励されるプロ・イダーはMSOLEDBSQLです。<u>https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-server-ver15</u>からダウロード可能です。 MapForce (32 ビット おけよ64 ビット)のプラトフォームに一致する必要があることに注意してくたさい。SQLOLEDB とSQLNCLI プロ・インダーは非奨励であると考えられています。

Microsoft OLE DB Provider for SQL Server (SQLOLEDB) は共通テーブル式 (CTE) とれた SELECT ステート メート などの複雑なクロリの デメーター シディング 引する問題があることが知られています。

ADO 接続の設定:

- 1. データベース接続ウィザードを開始します。
- 2. 「ADO 接続」をクリックします。

ADO接続	
「構築」ボタンをクリックして、 有効な接続を生成してください。 接続文字列が生成された後、 「 ボタンをクリックして 次に進んでください。	接約
構築	

3. 構築 をクルクします。

iii データ リンク プロパティ	×
プロバイダー 接続 詳細設定 すべて	
接続するデータを選択します:	
OLE DB プロバイダー	
Microsoft Jet 4.0 OLE DB Provider	
Microsoft Office 12.0 Access Database Engine OLE DB Pro	
Microsoft Office 15.0 Access Database Engine OLE DB Pro	
Microsoft OLE DB Provider for Indexing Services	
Microsoft OLE DB Provider for ODBC Drivers	
Microsoft OLE DB Provider for Oracle	
Microsoft OLE DB Provider for Search	
Microsoft OLE DB Provider for SQL Server	
Microsoft OLE DB Simple Provider	
OLE DB Provider for Microsoft Directory Services	
SQL Native Client	
次へ (N)>>	
OK キャンセル ヘルプ	

4. 接続するデータプロレイダーを選択します。下のテーブルは一般的なシナリをリストしています。

このデータベースに接続する方法	このプロバイダーを使用する
Microsoft Access	 Microsoft Office Access Database Engine OLE DB Provider (奨励) Microsoft Jet OLE DB Provider
	Microsoft Office Access Database Engine OLE DB Provider がJへりで使用できない場合 Microsoft Access おける Microsoft Access Database Engine Redistributable (<u>https://www.microsoft.com/en-us/download/details.aspx?</u> id=54920) がなビューターにインストールされていることを確認してくたさい。
SQL Server	 Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) がOLE DB プロ・イダーのナックに受励されてい ます。リストにこのプロ・イダーカ表示されるナックコよ プロ・イダーが https://docs.microsoft.com/en-

このデータベースに接続する方法	このプロバイダーを使用する
	us/sql/connect/oledb/download-oledb-driver-for-sql- server?view=sql-server-ver15 からゲンロードされインストールさ れる必要が防火ます。 Microsoft OLE DB Provider for SQL Server (OLEDBSQL) SQL Server Native Client (SQLNCLI)
他のデータベース	データベースに適用可能なプロバイダーを選択してくたさい。
	視聴中のデータベースへのOLE DB プロ・イダーが使用できない場合 データベー スペンダーから必要なドライバーをインストールします(<u>データベースドライバー概要</u> を 参照してくたさい)。代わりにADO.NET、ODBC、お当よJDBC 接続を設定 します。
	オペレーティングシステムが必要なデータベースのためにODBC ドライバーを必要と する場合 Microsoft OLE DB Provider for ODBC Drivers を使用 することもできます。まけよ <u>ODBC 接続</u> を使用してくたさい。

5. プロバイダーを選択後 [次へ]をクリックしてウィザードを完了します。

ウィザードの次のステップは選択されたプロ・イダーにと興ないます。SQL Server には、データベースサー・ドークオスト名、認証メンド、デー タベース名、およびデータベースユーザー名と「スワードを提供おけよ選択する必要があります。サンプルは関しては<u>Microsoft SQL Server</u> (ADO) への接続 セクションを参照ください。Microsoft Access に関しては、データベースファイルへの、文を参照おけは提供するように問 われます。サンプルは関しては<u>Microsoft Access (ADO) への接続</u> セクションを参照ください。

初期化プロノティの完全なリスト(接続ノラメーター)は接続ダイアログボックスの全てのタブで使用可能です。これらのプロノティは選択されたプロノイダーにより異なり、接続を可能にするために明示的に設定される必要かある場合があります。以下のセクションはMicrosoft Access とSQL Server データベースのナッナの基本的な初期化プロノティを構成する方法について説明しています。

- <u>SQL Server データリンクプロ・ティの設定</u>
- <u>Microsoft Access データリンクプロパティの設定</u>

6.2.1.3.1 既存の Microsoft Access データベースへの接続

このアプローチは、パマワードで保護されていないMicrosoft Access データベースは接続する場合に適しています。データベースがパワードによい保護されている場合、Microsoft Access (ADO) への接続で説明されている通りデータベースパマワードを設定してください。

既存のMicrosoft Access データベースへの接続方法:

- 1. データベース接続ウィザードを実行します(データベース接続ウィザードの開始を参照)。
- 2. Microsoft Access (ADO) を選択して、「次へ」をクルクします。
- 3. データベースファイルを参照、おうよ(相対おける絶対)パンを入力します。
- 4. 「接続」をクリックします。

6.2.1.3.2 SQL Server データリンク プロパティの設定

Microsoft SQL Server データベースにADO を介して接続する場合、次のデータレクプロ/ ディダイアログボックスの全ての タブ内の次の接続プロ/ディを設定する必要があります。

プロバイダー 接続 詳細設定 す	57/7	
以下は、この型のデータの初期化: ロパティを選択して [値の編集]をり	プロパティです。値を編集する場 フリックします。	合は、プ
名前	値	
Application Name Auto Translate	True	
Connect Timeout Current Language		=
Data Source DataTypeCompatibility Extended Properties	0	
Failover Partner General Timeout Initial Catalog	0	
Initial File Name Integrated Security Locale Identifier		
MARS Connection		T
値の編集(E)		
ОК	キャンセル	へルプ

プロパティ	冼
統合セキュリティ	プロバイダータブで SQL Server Native Client カデータプロバイダーに選択された 場合は、このプロバティを、空白文字に設定します。
セキュリティ情報を保持	このプロ・ディをTrueに設定する。

6.2.1.3.3 Microsoft Access データリンク プロパティの設定

Microsoft Access データベースにADO を介して接続する場合、次のデータレクプロノティダイアログボックスの全てのタブ内の次の接続プロノティを設定する必要があります。

プロバイダー 接続 詳細設定 すべて		
以下は、この型のデータの初期化プロパティ ロパティを選択して [値の編集]をクリックし	(です。値を編集する場合は ます。	に プ
名前	値	
Data Source		
Extended Properties		
Jet OLEDB:Compact Without Re	False	
Jet OLEDB:Create System Data	False	Ξ
Jet OLEDB:Database Locking M	1	
Jet OLEDB:Database Password		
Jet OLEDB:Don't Copy Locale o	False	
Jet OLEDB:Encrypt Database	False	
Jet OLEDB:Engine Type	0	
Jet OLEDB:Global Bulk Transac	1	
Jet OLEDB:Global Partial Bulk O	2	
Jet ULEDB:New Database Pass		-
	4	
値の編集(E)		
ОК	キャンセル ヘル	1

データ リンク プロハティ ダイアログ ボックス

プロパティ	灹
データソース	このプロ・ディーは、Microsoft Access データベースファイルへの、文を保管します。デ ータベースへの接続問題を回避するために、UNC (Universal Naming Convention 汎用名前付け規則)・ペンオーマトの使用が奨励されます。例: \\anyserver\share\$\filepath
Jet OLEDB: システムデータベース	このプロ・ティオワークグループ情報ファイルへの、文を保管します。データベースは接続す る前に、このプロ・ティを明示的に設定する必要がある可能性があります。 "ワークグループ情報ファイル"エラーが理由で接続できない場合、ワークグループ情報 ファイル(System.MDW)をユーザープロファイルに適用できるように設定します(手順はhttp://support.microsoft.com/kb/305542 を参照してくたさい)、プロ・ティ の値を System.MDW ファイルの、文に設定します。

プロパティ	灹		
	プロパティの値を編集 メ プロパティの説明 Jet OLEDB:System database プロパティの値(V) C:\Users\john.doe\AppData\Roaming\Microsoft\Access (値のリセット(R) OK キャンセル		
Jet OLEDB: データベースパスワード	データベースが なワードで保護されている場合は、このプロッ 守っの値をデータベースのパスワードに設定します。 プロパティの値を編集 プロパティの説明 Jet OLEDB: Database Password プロパティの値(V) [値のリセット(R) OK キャンセル		

6.2.1.4 ADO.NET 接続のセットアップ

ADO.NET は、データベースからのデータを含むデータと作業するか。かごデザインされた Microsoft.NET フレームワーク ライブラルのセオで す。MapForce から、データベースは接続するはよ ADO.NET、Microsoft.NET フレームワーク4、おまよ 以降を使用する必要があ ります。下に表示されているとおり、.NET プロ・イダーを選択して、接続文字列を提供し、データベースにADO.NET を介して接続しま す。

.NET データプロ・イダーは、コマトを実行する特定の型のデータノースへの接続を可能にするクラスのロネクションです(例えば、SQL Server ませまOracle データベースなど)。すなオち、ADO.NET どよ、MapForce などのアプリケーションがデータベースプロ・イダーを 介してデータベースと作動することを意味します。各 データプロ・イダー」は特定の型のデータノースと作動するように最適化されていま す。.NET プロ・イダー」コお以下の2つの種類があります:

- 1. デフォルとしてMicrosoft.NET フレームワークを使用して提供されます。
- 2. 主要なデータベースペンダーを使用して.NET フレームワークの拡張とて提供されます。このような ADO.NET プロバイダーは、 個別人ストールされている必要があり、通常それぞれのデータベースペンダーのケェブサイトからダウンロードすることができます。

メモ 特定のADO.NET プロ・イダーは、サポートされていない、おけは、サポートが制約されています。 <u>ADO.NET サポートに関する</u> メーを参照してください。

ADO.NET 接続の作成:

- 1. <u>データベース接続ウィザードを開始します</u>。
- 2. 「ADO.NET 接続」をクリックします。

3. .NET データプロバイダーをリストから選択します。

.NET フレームワークと共に使用することのできるプロ・イダーのノストはデフォルト でプロ・イダーのノストに表示されます。ベンダーン特有の.NET データプロ・イダーは、使用中のシステムコ既にインストールされている場合のみ使用することができます。.NET データプロ・イダーは、データベースペンダーには提供される.msi、おけよ.exe ファイルを実行して、GAC (Global Assembly Cache) 内にインストールされる必要があります。

 データベース接続文字列を入力します。接続文字列は、データベース接続情報をセミコロンで区切られた接続、デメーターのキー 値ペアとして定義します。例えば、Data Source=DBSQLSERV; Initial Catalog=ProductsDB; User
 ID=dbuser; password=dbpass などの文字列は、サーバーDBSQLSERV 上で、ユーザー名 dbuser といなワード dbpass を使用して、SQL Server データベース ProductsDB に接続します。接続文字列を、キー値ペアを直接「接続文 字列」ダイアログボックスに入力すること」、作成することができます。他の接続オプションは Visual Studio で作成することがで きます(次を参照 Visual Studio 内で接続文字列を作成する)。

接続文字列の構文は、プロ・イダーリストより選択されたプロ・イダーにより異なります。例えば、次を参照してくたさい、サンプルADO.NET接続文字列。

ADO.NET 接続		
リストからプロバイダーを	・選択し、有効な接続文字列を入力し、「接続」をクリックして継続しま	ます。
プロバイダー:	.Net Framework Data Provider for SqlServer	-
接続文字列:	Data Source=DBSQLSERV;Initial Catalog=ProductsDB;User ID=dbuser;Password=dbpass	*
		-፩(L)

5. 「接続」をクリックします。

6.2.1.4.1 Visual Studio 内で接続文字列を作成する

ADO.NET を使用してデータノースコ接続するコよ、有効な接続文字列が必要です。次の手順は、Visual Studio から接続文字列を作成する方法を説明しています。

Visual Studio 内で接続文字列を作成する

- 1. 「ツール」メニューから「データベースへの接続」をクルクします。
- 2. リストからデータノースを選択します(この例の場合、Microsoft SQL Server)。データプロレイダーは、選択に従い自動的に記入されます。

Choose Data Source	? ×		
Data <u>s</u> ource: Microsoft Access Database File Microsoft ODBC Data Source <u>Microsoft SQL Server</u> Microsoft SQL Server Database File Oracle Database < other> Data <u>provider</u> :	Description Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.		
Always use this selection	Continue Cancel		

3. 「接続」をクリックします。

Modify Connection			? ×		
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.					
Data <u>s</u> ource:					
Microsoft SQL Server (SqlClient)		<u>C</u> hange			
S <u>e</u> rver name:					
DBSQLSERV	DBSQLSERV ~		<u>R</u> efresh		
Log on to the serv	/er				
O Use Windows	Authentication				
Use SQL Serve	r Authentication				
User name:	dbuser				
- Password:					
<u>r</u> assword.					
	ave my password				
Connect to a data	base				
Select or enter	r a <u>d</u> atabase name:				
ProductsDB			~		
Attach a database file:					
			Browse		
Logical name	2 1				
			Ad <u>v</u> anced		
Test Connection		ОК	Cancel		

- 4. データベースへのサーバーオスト名、ユーザー名、および マワードを入力します。この例の場合、サーバー DBSQLSERV 上のデー タベース Products DB にSQL Server 認証を使用して接続します。
- 5. 「OK」をクルクします。

データベースへの接続に成功した場合、サードーエクスプローラーウイドウに表示されます。サードーエクスプローラーウイドウを、メニューコマ ンド「表示 | サーバーエクスプローラー」を使用して表示することができます。データベース接続文字列を取得するコよ、サードーエクスプロ ーラーウイドウ内の接続を右クルクし、「プロパティ」を選択します。接続文字列が、Visual Studio プロパティウイドウに表示されま す。MapForceの「接続文字列」ボックスに文字列を貼り付ける前に、アスタリスク(*)文字を実際のパマワードと置き換える必要がある ことに注意してください。

6.2.1.4.2 サンプル ADO.NET 接続文字列

ADO.NET 接続をセトアップするコよ データベース接続ダイアログボックスからADO.NET プロ・イダーを選択し、接続文字列を入力 します。(次も参照してくたさい<u>ADO.NET 接続のセ・トアップ</u>)。異なるデータベースのサンプルADO.NET 接続文字列は.NET プロ・バ イダーの下にコストされています。

Teradata のための.NET Data

このプロ・イダーは、次のTeradata Web サイトからダウノロードすることができます (<u>http://downloads.teradata.com/download/connectivity/net-data-provider-for-teradata</u>)。サンプル接続は以下のとおしで す:

Data Source=ServerAddress;User Id=user;Password=password;

IBM i のための.NET Framework データプロレイダー

このプロバイダーは、IBM i Access Client Solutions - Windows Application Package の一部としてインストールされます。サンプル接続文字列は以下のとおりです。

DataSource=ServerAddress;UserID=user;Password=password;DataCompression=True;

詳細に関しては、インストールッケージ内に含まれる「NETプロバイダー技術レファレンスへルプファイル」を参照してくたさい。

MySQL のための.NET Framework データプロレイダー

このプロ・イダーをMySQL ウェブサイトからダウンロードすることができます(<u>https://dev.mysql.com/downloads/connector/net/</u>)。 サンプル接続文字列は以下のとおどです:

Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;

次も参照して
が
さい
https://dev.mysql.com/doc/connector-net/en/connector-net-programming-connecting-connection-string.html

SQL Server のための.NET Framework データプロバイダー

サンプル接続文字列は以下のとおりです。

Data Source=DBSQLSERV; Initial Catalog=ProductsDB; User ID=dbuser; Password=dbpass

次も参照してくだとい https://msdn.microsoft.com/en-us/library/ms254500(v=vs.110).aspx
.NET Framework 4.0 のためのIBM DB2 データプロバイダー10.1.2

Database=PRODUCTS;UID=user;Password=password;Server=localhost:50000;

メモ このプロ・イダーは、通常 IBM DB2 Data Server Client パッケージと共にインストールされます。 IBM DB2 Data Server Client パッケージをインストール後 ADO.NET プロ・イダーのノスト内でプロ・イダーが不足している場合、次の技術メモを参照してくたさい、http://www-01.ibm.com/support/docview.wss?uid=swg21429586。

次も参照してくたさい

http://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.1.0/com.ibm.swg.im.dbclient.adonet.ref.doc/do c/DB2ConnectionClassConnectionStringProperty.html

.NET (ODP.NET) のためのOracle データプロバイダー

ODP.NET プロバイダーを含むインストールパッケージを Oracle ウェブサイトからダウンロードすることができます(次を参照してくたさい: http://www.oracle.com/technetwork/topics/dotnet/downloads/index.html).サンプル接続文字列は以下のとおしてす:

Data Source=DSORCL;User Id=user;Password=password;

Oracle への接続 ODBC)内で説明されているようこtnsnames.ora ファイル内で定義された Oracle サービス名をさす DSORCL かデ ータノース名の場合。

tnsnames.ora ファイル内のサービス名を構成せず接続する場合は、以下のような文字列を使用してくたさい

Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=user;Password=password;

次も参照して(たさい) https://docs.oracle.com/cd/B28359 01/win.111/b28375/featConnecting.htm

6.2.1.4.3 ADO.NET サポートに関するメモ

下のテーブルはMapForce内で現在サポトされていない、ませま制限のあるサポトのある既知のADO.NETデータベースドライバーがリストされています。

データベース	ドライバー	ザポート ノート
全てのデータベース	ODBC のためのNet Framework Data Provider	制限つ さりポート 。MicrosoftAccess 接続の際に既 知の問題がみます。ODBC への直接の接続を代わり に使用することが奨励されます。
	OleDb <i>のナ_めの</i> .Net Framework Data Provider	制限つ さりポー 。MicrosoftAccess 接続の際に既 知の問題がみます。ADO への直接の接続を代わりに 使用することが奨励されます。
Firebird	Firebird ADO.NET Data Provider	制限つきサポト。ODBC おはJDBC を代わりに使

データベース	ドライバー	ザポート ノート
		用することが奨励されます。
Informix	.NET Framework 4.0 のためのIBM Informix Data Provider	サポートされて、ません。 DB2 Data Server Provider を代わりに使用してくたさい。
IBM DB2 for i (iSeries)	i5/OS <i>の†ුහි</i> D.Net Framework Data Provider	サポートされて、ません。IBM i Access Client Solutions - Windows Application Package の 一部とて提供されている.Net Framework Data Provider for IBM i を代わりに使用してください。
Oracle	Oracle のための.Net Framework Data Provider	制限つきサポト。.NET Framework と共にこのドライ バは提供されて、ますが、Microsoft によ使用を奨励 されて、ません。
PostgreSQL	-	このシダーのオダのADO.NETドライバはサポートされていません。ネイティブは接続を代わりに使用してくたさい。
Sybase	-	このペダーのオダのADO.NETドライバはサポートされていません。

6.2.1.5 ODBC 接続の設定

ODBC (Open Database Connectivity)は、一般に使用されているデータアクセス技術で、MapForce からデータベースは接続する ことができます。データベースにアクセスするための主要な手段として、ませまネイティブな OLE DB- やJDBC ーによる接続の代替として使用 することができます。

ODBCを使用してデータベースは接続するはよ、最初にODBCデータノース名(DSN)を作成する必要が対けます。このステップは、他の ユーザーませはオペレーティングシステムによりDSN が既に作成されている場合必要ありません。DSN は、統一された、MapForceを含 むオペレーティングシステム上のODBC に対応したクライアントアプリのデータベース接続方法を表します。DSN はは以下の種類が対けます。

- ファイルDSN
- ユーザーDSN
- ファイルDSN

オペレーティングシステムで特権を持つすべてのユーザーはシステムデータソースにアクセスすることができます。作成したユーザーのみがユーザ ーデータソースにアクセスすることができます。ファイルDSNを作成した場合、データノースは.dsn 拡張子を持つファイルとして作成され、 データノースにより使用されているドライバーがインストーリにわていることが前提で、他のユーザーと共有することができます。

ODBC 接続ダイアログボックス内の[ODBC 接続] をクリックすると、使用中のエレビューターで既に使用することのできるDSN はデータベ ース接続ダイアログボックスに表示されています。

◎システム DSN ◎ユーザー DSN	◎ 接続文字列を構築	
○ファイル DSN	▼	
ノーダンニス-石	Driver de Microsoft rere erguivos texto (# tyt: #.cey)	
BASE Files	Microsoft Access dBASE Driver (# dbf # ndv # mdv)	
ddd	Driver da Microsoft para arguivos texto (* txt: * csv)	
Excel Files	Microsoft Excel Driver (*xls. *xlsx. *xlsm. *xlsb)	
MS Access Database	Microsoft Access Driver (* mdb. * accdb)	
OfficeSales DB2	Driver da Microsoft para arquivos texto (* txt: *.csv)	
/ietsql05	Driver da Microsoft para arquivos texto (*.txt; *.csv)	
√isio Database Samples	Microsoft Access Driver (*.mdb, *.accdb)	
+, ₩ ⊘ 5		

ODBC 接続ダイアログボックス

データベースこと必要とされるDSN が無い場合、MapForce データベース接続ウィザードが作成の手助けをします。しかし、Windows オペレーティングシステムニ直接作成することもできます。いずれにせよ、続行する前に、使用中のデータベースにODBC ドライバーカオペレー ティングシステムのODBC ドライバーリストに適用できることを確認してくけざい。(使用可能な ODBC ドライバーの表示を参照)。

新規のDSNを使用しての接続:

- 1. データベース接続ウィザードを開始します。
- 2. データベース接続ダイアログボックスの[ODBC 接続] をクトックします。
- 3. システムDSN を作成するコよ オペレーティングシステムの管理者権限が必要です。

また、MapForce が管理者とて作動している必要があます。

4. 「追加」 🧚 をクルクします。

- 5. ドライバーを選択して、(作成するDSN の種類に応じて) [ユーザーDSN] お」は[システムDSN] をクックします。使用中 のデータベースに適用することのできるドライバーがリストされていない場合、データベースペンダー からダウンロードして、インストール してください。(次を参照: データベースドライバー概要)。
- 6. ポップアップするダイアログボックスに特定の接続情報を入力してセルアップを完了します。

接続を成功させるコよ、データベースサーバーの大スト名(おけよIP アドレス)、とデータベースのユーザー名およびが スワードを提供する必要が あります。他の接続パラメーターがある可能性がありますが、これらの ウメーターはデータベースプロ・イダーにより異なります。各接続方法に 固有の ウメーターについての詳細は、ドライバープロ・イダーのドキュメントを参照してくたさい。一度作成されると、DSN はデータノース名 のノストで使用することができます。これによりデータベース接続の詳細をデータベースの接続を希望する際いつでも使用することができます。ユ ーザーDSN はユーザーDSN のノストに追加されますが、システムDSN はシステムDSN のノストに追加されます。

既存のDSNを使用しての接続:

- 1. データベース接続ウィザードを開始します。
- 2. [ODBC 接続] をクトックします。
- 3. 既存のデータノーズユーザーDSN、システムDSN、ファイルDSN)から種類を選択します
- 4. 既存 DSN レコードをクルクして、「接続」をクルクします。

既存の.dsn ファイルをベースコして接続文字列を構築する

- 1. データベース接続ウィザードを開始します。
- 2. [ODBC 接続] をクルクします。
- 3. [接続文字列の構築]を選択して[構築]をクリックします。
- 4. ファイルDSN を使用して、接続文字列を構築する場合、[ファイルデータソース] ダブをクトックします。それ以外の場合は、[マシ ーンデータソース] ダブをクトックします。(システムDSN とユーザーDSN は "マシン" データノースとして知られています。)
- 5. 必要とされる.dsn ファイルを選択して、「OK」をクリックします。

用意された接続文字列を使用して接続する

- 1. <u>データベース接続ウィザードを開始します</u>。
- 2. [ODBC 接続] をクリックします。
- 3. [接続文字列の構築] を選択してます。
- 4. 接続文字列を与えられたボックスはいけけ、「接続」をクリックします。

6.2.1.5.1 使用可能な ODBC ドライバーの表示

ODBC データノース管理で使用中のオペレーティングシステムで使用可能な ODBC ドライバーを確認することができます。Windows コントロール ないの着理ツールから ODBC データノース管理 (Odbcad32.exe) にアクセスすることができます。64-ビナト版オペレーティングシステムコよ 2 種類の デジョンの実行可能ファイルがあります。

- C:\Windows\SysWoW64 デルオリニある、Odbcad32.exe ファイルの32ビナ版バージョン(C: が使用中のシステムドライブとして前提されています)。
- C:\Windows\System32 デルケリ ある Odbcad32.exe ファイルの64 ビナ版バージョン

インストールされている32-ビナ版データベースドライバーは、ODBC データノース管理の32-ビナ版バージョンで、64-ビナ版バージョン内の64-ビナ版ドライバーで確認することができます。ですから、対応するバージョンのODBC データノース管理でデータベースドライバーを確認してください。

🖥 ODBC データ ソース アドミニストレーター		×	
ユーザー DSN システム DSN ファイル DSN ドライバー ト	レース 接続プール	バージョン情報	
このコンピューターにインストールされている ODBC ドライバー	(0):		
名前	バージョン	会社名	
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)	14.00.7010.1000	Microsoft Cı	
Microsoft Access Driver (*.mdb, *.accdb)	14.00.7010.1000	Microsoft C	
Microsoft Access Text Driver (*.txt, *.csv)	14.00.7010.1000	Microsoft Cı	
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	14.00.7010.1000	Microsoft C	
SQL Native Client	2005.90.5000.00	Microsoft C	
SQL Server	6.01.7601.17514	Microsoft C	
SQL Server Native Client 11.0	2011.110.2100.60	Microsoft C	
•		4	
ODBC ドライバーを使用すると、ODBC が有効なプログラムで ODBC データ ソー スから情報を取得することができます。新しいドライバーをインストールするには、ドラ イバーのセットアップ プログラムを使用してください。			
OK キャンセル 適用(A) ヘルプ			

ODBC データソース管理者

ターゲトデータベースのドライバーがリストに存在してい場合、まけよ、代替ドライバーを追加する場合、データベースのペンダーからダウロードしてくたさい(データベースドライバー概要を参照)。ODBCドライバーがシステムで使用可能になると、ODBC接続を作成することができます(ODBC接続のセルアップを参照)。

6.2.1.6 JDBC 接続の設定

JDBC (Java Database Connectivity) は、Oracle のJava ソストウェアプラオフォームの一部であるデータベースアクセスインター フェイスです。JDBC 接続はODBC 接続に比べ、通常よりドリノースを集中的に使用する接続ですが、ODBC で使用することのできな し機能を使用することができます。 ODBC コネクタにより使用できない データベース機能を使用する場合 JDBC 接続を使用することが奨 励されています。

前提条件

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がインストールされている必要があります。 これは Oracle JDK おは Oracle Open JDK などのオープンノースビルドであることができます。MapForce は次のエケー ションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプリケーション オプション内で設定したカスタ ム JVM パス <u>Java 設定</u>を参照してくたさい。b) Windows レジストリ内で見つかすこ JVM パス。c) JAVA_HOME 環境変 数。
- MapForce (32 ビオ、64 ビオ)のプラオフォームがJRE /JDK のプラオフォームに一致することを確認してくたさい。
- データベースペンダーからのJDBCドライバーがインストールされている必要があります。これは使用可能でありデータベースによりサポートされている場合、データベースクライアントインストールの一部として、おけよ、JDBC ライブラリ(.jar ファイル)として個別にダウンロードされているJDBCドライバーの可能性があります。データベース接続のサンプルも参照してください。
- Oracleデータベースは接続する場合は、一部のOracleドライバーは特定のJRE バージョンに固有のため、追加コンポーネント と構成が必要な点に注意してくたさい。Oracle 製品のドキュメント (例えば、「JDBC開発者ガイドおよびリファレンス」) によ 各 JDBC ドライバーの構成の手続きの詳細が説明されています。

 CLASSPATH環境変数は、(.jar ファイルの)つまけは複数の使用中のWindows オペレーティングシステムのJDBC ドライ バーの文を含む必要かあります。データベースクライアトをインストールする際に、インストーラーはこの変数を自動的に構成する 可能性があります(CLASSPATHの構成を参照)。

Windows 資格情報を使用してJDBC を介してSQL Server に接続する方法

Windows 資格情報(統合されたセキュリティ)を使用して JDBC を介して SQL Server は接続する場合、以下に注意してくたさい

- JDBC ドライバー・ジケージに含まれて、る sqljdbc_auth.dll ファイルはシステム PATH 環境変数上のディレオ・リニピーされる必要があります。このようなファイルはと個存在します。1つはx86 プラホフォームのために、もう1つはx64 プラホフォームのために存在します。JDK プラホフォームコ対応する PATH が追加されていることを確認してくたさい。また、環境変数の変更後MapForce(適用可能な場合マピングを実行しているプログラム)を再起動してくたさい。
- JDBC 接続文字列はプロ/ ディ integratedSecurity=true を含む必要があります。このプロ/ ディを異なる場所から追加することができます。
 - データベース接続ウィザードから追加することができます。以下を参照してください。
 - データベース <u>コレポーネト 設定</u>から追加することができます。
 - 適応可能な場合、生成済みのJava コード内のデータベース接続文字列の編集をすることには追加することができます。

詳細に関しては SQL Server のすめの Microsoft JDBC ドライバードキュメトを参照してくたさい。 https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url。

JDBC 接続のセオアプ方法

- 1. データベース接続ウィザードを開始します。
- 2. 「JDBC 接続」をクトックします。
- 3. 任意で、セミコロンで区切られた.jar ファイル なのノストを「クラス・ス」テキストボックスに入力することもできます。ここで入力される.jar ライブラルは、CLASSPATH環境変数内で既に定義されているものに加え、環境にコードされます。「クラス・ス」テキストボックスの編集が完了すると、ソース.jar ライブラル内で検索されるJDBCドライバーは、「ドライバー」、リストに自動的に追加されます(次のステップを参照してくたさい)。

クラスパス:	C:\jdbc\instantclient_12_1\odbc7.jar
ドライバー:	oracle.jdbc.OracleDriver
ユーザー名:	jphndoe
パスワード:	•••••
データベース URL:	jdbc:oracle:thin:@//ora12c:1521:orcl12d
	~
	接続(C) 閉じる(L)

4. 「ドライバー」の横でJDBC ドライバーをパストから選択、おけよ Java クラス名を入力します。 CLASSPATH 環境変数により 構成された JDBC ドライバーおよび、「クラスパス」テキストボックス内で検出されナドライバーが含まれます(次を参照 CLASSPATH の構成)。 CLASSPATH 変数内で定義されている JDBC ドライバーの やを使用することができます。 おっ、データベース接続ダイア ログボックス ご直接入力された.jar パマは Java Virtual Machine (JVM) によりサポートされています。 JVM (は接続 を構築するけっかのドライバーを決定します。 JVM 「コロードされる Java クラスを確認し、 JDBC ドライバーの競合、 およ び、データベース は接続する際に発生する予期されない ロラーを回避するよう こ、てくたさい。

- 5. 対応するボックスに ユーザー名と マワードを入力します。
- 6. データベースURL テキストボックス内に、JDBC 接続 URL(文字列) をデータベースの型のとおり入力します。次のテーブルは一般のデータベースの型のナダの JDBC 接続 URL(文字列)の構文について説明しています。

データベース	JDBC 接続 URL
Firebird	jdbc:firebirdsql:// <host>[:<port>]/<database or<br="" path="">alias></database></port></host>
IBM DB2	jdbc:db2:// hostName:port/databaseName
IBM DB2 for i	jdbc:as400:// [host]
IBM Informix	jdbc:informix- sqli:// hostName : port/databaseName :INFORMIXSERVER= myserver
MariaDB	jdbc:mariadb:// hostName:port / databaseName
Microsoft SQL Server	jdbc:sqlserver:// hostName:port;databaseName =name
MySQL	jdbc:mysql:// hostName:port / databaseName
Oracle	jdbc:oracle:thin:@ hostName:port:SID jdbc:oracle:thin:@// hostName:port/service
Oracle XML DB	jdbc:oracle:oci:@// hostName : port:service
PostgreSQL	jdbc:postgresql:// hostName:port / databaseName
Progress OpenEdge	jdbc:datadirect:openedge:// host : port; databaseName= db_name
Sybase	jdbc:sybase:Tds: hostName : port / databaseName
Teradata	jdbc:teradata:// databaseServerName

- メモ 上記のフォーマナーに対する構文の・リエーションも使用することができます(例えば、データベースURL はポートを除外、おけまデ ータベースのユーザー名と、マワードを含む可能性があります)。詳細はデータベースのペンダーのドキュメトを参照してくけざい。
 - 7. 「接続」をクリックします。

6.2.1.6.1 CLASSPATH の構成

CLASSPATH 環境変数はクラスと、オペレーティングシステム上の他のリソースファイルを検索するため Java Runtime Environment (JRE) おけま Java Development Kit (JDK) によ使用されます。JDBC を介してデータベースは接続すると、この変数はオペレー

ティングシステム上のJDBC ドライバーへの ぷ、おけま一部の場合、使用中のデータベースの種類に固有の追加ライブライファイルへの ぷ、を含むように構成されなければよりません。

下のテーブルは、CLASSPATH 変数に通常含まれる必要のあるサンプルファイル やをレストしています。重要なことは、システム上の清報をシステム上のJDBC ドライバーのロケーション、JDBC ドライバー名、およびオペレーティングシステム上に存在する JRE /JDK バージョンを ベース したこの情報を調整する必要があります。接続問題を避けるために、使用中のオペレーティングシステムこインストールされている JDBC ドライバニ適用されるインストールの指示、インストール前とインストール後の構成ステップを確認してくたさい。

データベース	サンプルCLASSPATH エントリ	
Firebird	C:\Program Files\Firebird\Jaybird-2.2.8-JDK_1.8\jaybird-full- 2.2.8.jar	
IBM DB2	C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc.jar;C:\Program Files (x86)\IBM\SQLLIB\java\db2jcc_license_cu.jar;	
IBM DB2 for i	C:\jt400\jt400.jar;	
IBM Informix	C:\Informix_JDBC_Driver\lib\ifxjdbc.jar;	
Microsoft SQL Server	C:\Program Files\Microsoft JDBC Driver 4.0 for SQL Server\sqljdbc_4.0\enu\sqljdbc.jar	
MariaDB	<installation directory="">\mariadb-java-client-2.2.0.jar</installation>	
MySQL	<pre>installation directory>\mysql-connector-java-version-bin.jar;</pre>	
Oracle	ORACLE_HOME \jdbc\lib\ojdbc6.jar;	
Oracle (with XML DB)	<pre>ORACLE_HOME\jdbc\lib\ojdbc6.jar;ORACLE_HOME\LIB\xmlparserv2.jar ;ORACLE_HOME\RDBMS\jlib\xdb.jar;</pre>	
PostgreSQL	<installation directory=""><pre>\postgresql.jar</pre></installation>	
Progress OpenEdge	<pre>%DLC%\java\openedge.jar;%DLC%\java\pool.jar;</pre>	
	メモProgress OpenEdge SDK がマンノニインストールされていると想定します。 & DLC% は OpenEdge がインストールされているディレクトリです。	
Sybase	C:\sybase\jConnect-7_0\classes\jconn4.jar	
Teradata	<pre><installation directory="">\tdgssconfig.jar;<installation directory="">\terajdbc4.jar</installation></installation></pre>	

- CLASSPATH 変数の変更は、使用中のエンピュータのJava アプリケーションの振る舞いに影響する可能性があります。続行する前に生じえる影響を把握するために、Java ドキュメトを参照してくたさい。
- 環境変数はユーザーおゴネンステムであることができます。システム環境変数を変更するコネオペレーティングシステムに対する管理者権限が必要です。
- 環境変数を変更後、設定の変更を有効にするコよプログラムを再起動してくたさい。ませよオペレーティングシステムをログオフまた は再起動してくたさい。

Windows 7 でCLASSPATH を構成する

1. [スタート] メニューを開始して、[コンピューター] を右クルクします。

- 2. [プロパティ]をクリックします。
- 3. [システムの詳細設定]をクリックします。
- 4. [詳細設定] タブから[環境変数] をクルクします。
- 5. CLASSPATH 変数を、ユーザーおけよンステム環境変数からロケートして、編集をクリックします。CLASSPATH 変数が存在 しない場合は [新規] をクリックして作成します。
- 6. オペノーティングシステム内で JDBC ドライバー かある場所に ひを含むように変数の値を編集します。 JDBC ドライバー ひを他の既に CLASSPATH 変数内にある ひから区別するためにたるコロン(;)で区切ります。

Windows 10 でCLASSPATH を構成する

- 1. Windows キーを押して、環境編集を入力します。
- 2. [システム環境変数の編集]の提案をクリックします。
- 3. [環境変数]をクルクします。
- 4. CLASSPATH 変数を、ユーザーおゴエンステム環境変数からロケートして、編集をクリックします。CLASSPATH 変数が存在 したは場合は、[新規]をクリックして作成します。
- 5. オペーティングシステム内でJDBCドライバーかある場所に、なを含むように変数の値を編集します。JDBCドライバー、なを他の既にCLASSPATH変数内にある、なから区別するためにたミコロン(;)で区切ります。

6.2.1.7 PostgreSQL 接続のセットアップ

PostgreSQL データベースへの接続は、ネイティブは接続として、封まは、ODBC、JDBC、および他のドライバーを使用して確立することができます。ネイティブは接続の利点は、使用中のシステムニインストールする場合、ドライバーを必要としないことです。また、Linux、また は、macOS サーバー上での実行のためにファイルをデプロイする場合、ターゲオ・サーバー上にボライバーをインストールする必要におりません

ネイティブでは内ドライバーを使用しての接続を作成する場合は、以下のトピックを参照してくたさい

- JDBC 接続の設定
- <u>PostgreSQL(ODBC)</u> ~ の接続

それ以外の場合、PostgreSQL へのオイティブの接続をセナアップする場合、以下の手順に従ってくたさい。継続するコよ、以下の前提 条件が必要です:ホスト名、ポート、データベース名、ユーザー名、および、パマワード。

ネイティブ PostgreSQL 接続のセットアップ:

- 1. データベース接続ウィザードを開始します.
- 2. 「PostgreSQL 接続」をクリックします。
- 3. 対応するボックスに、ホスト(PostgreSQL が同じマンンで作動している場合、*ローカルホスト*)、ポート(通常、5432 ですが、これは任意です)、データベース名、ユーザー名と マワードを入力します。

Postgre データベ て接続してくださ	ースのために必要なパラメーターを入力してください。 データベースの横の[次へ]をクリックし い。
ホスト:	DBSERV
ポート:	5432 [オプション]
データベース:	200
ユーザー名:	dbuser
パスワード:	••••••

4. 「接続」をクリックします。

PostgreSQL データベースサーバーが異なるマンメニ存在する場合、次の点に注意してくたさい

- PostgreSQLデータベースサーバーは、クライアントからの接続を受け入れるようご構成されなければむりません。具体的には、 pg_hba.conf ファイルがローカルではない接続を許可するようご構成される必要かあります。第2に、postgresql.conf ファ イルは、特定されたIP ブドレスとポートをリンスンするようご構成される必要があります。詳細に関しては、PostgreSQLドキュ メントを参照してくたさい(https://www.postgresql.org/docs/9.5/static/client-authentication-problems.html)。
- サーバーマンカウァイアウォールを越えてデザインされたポート(通常、5432)上で接続を受け入れるように構成されている必要が あります。例えば「コントロールレペルン>Windowsファイアウォール>詳細設定>受信の規則」で設定すると Windows上で作動するデータベースサーバーでは、ファイアーウォールを越えてポート 5432 は接続を許可するルールを作成する 必要がない場合かあります。

6.2.1.8 SQLite 接続の設定

SQLite (<u>http://www.sqlite.org</u>) は ポータブル性とインストールの簡便性が重要な場合理想的な構成のファイルベースの、内蔵のデー 夕型です。 Since SQLite データベースは MapForce によりネイティブ こサポート されるため、接続のため ボライバーをインストールする必 要はありません。

SQLite データベースサポートに関するメモ

- (マピングをパビューおけはMapForce Server 実行ファイルを実行する際) SQLite データベースはMapForce 内蔵 変換言語によりサポートされています。
- ユーザー定義関数(UDF)はSQLite データベースをサポートしません。

- Linux では SQLite データベースのステートメート実行タイムアナトはサポートされません。
- フルテキスト検索テーブルはサポートされません。
- テーブルの各行内の異なるデータ型の値をSQLite は許可します。MapForce では、処理されたすべての値は宣言された列の型との互換性が必要です。ですから、SQLite データベースが宣言された列の値と異なる行の値を持つ場合ランタイムエラーが起る可能性があります。
- SQLite データベースにマピングを書き込む場合、おこ、ターゲトデータベースが存在しない場合、外部にターゲトデータベース を作成する必要が決ます。この場合、DatabaseSpy (<u>http://www.altova.com/ja/databasespy.html</u>)などのソールを 使用して作成するか、おけま SQLite コマイラインシェルを公式のWeb サイトカらダウムロードして、コマイライからデータベ ースファイルを作成してください(サンプル XML から SQLite ヘデータをマップ)を参照)。SQLite コマイド構文に関する完全なレ ファレンスは、SQLite の公式のドキュメトを参照してください。

6.2.1.8.1 既存の SQLite データベースへの接続

既存のSQLite データベースに接続する方法:

- 1. データベース接続ウィザードを実行します(データベース接続ウィザードの開始を参照)。
- 2. SQLite を選択して、「次へ」をクリックします。
- 3. SQLite データベースファイルを参照して、おけよ (相対おけは絶対)/ ひを入力します。「接続」 ポタンは データベースファイルへの ひを入力すると有効化されます。
- 4. 「接続」をクリックします。

6.2.1.9 グローバルリソースからの接続の使用

Altova グロー・シリソースはファイル、フォルダー、およびデータベースへのポーダブルなレファレンスです。以前にデータベースをグロー・シリソノース として既に構成している場合、接続をいつでも再使用することができます(異なる Altova アプリケーション間でも使用可能)。詳細に関して は、Altova グロー・シリソノースを参照してください。

グロー・バリノノースからのデータベース接続の使用:

- 1. データベース接続ウィザードを開始します。
- 2. [グロー・・シリノース]をクトックします。グロー・シリノースとして使用可能なデータベース接続がノストされます。

グローバル リソース
グローバル リソース名
Oracle11g
MySQL
SQL_Server

- 3. データベース接続履歴から「接続」をクリックします。
- ヒント: 各グロー・シリノノースの追加情報を取得するために、マウスカーノルをグロー・シリノノースコポイントします。

6.2.1.10 データベース接続のサンプル

ADO、ODBC、およし、DBC を使用したMapForce からのデータベースへの接続のナダのサンプルがこのセクションには含まれています。 ADO.NET 接続サンプルは個別にリストされています サンプルADO.NET 接続文字列 を参照してくたさい。For instructions about establishing a native connection to PostgreSQL とSQLite へのオイティブな接続の作成のナダの手順は、それぞれ PostgreSQL 接続のセナア・プ とSQLite 接続のセナア・プ を参照してくたさい。

以下の点に注意してくたさい

- Windows 構成、やホワーク環境、およびデータベースクライアント、おけよ、サーバーノフトウェアが各サンプルと異なる場合手順は異なります。
- I思ますべてのデータベースの型では、1つ以上のデーダアクセステクノロジー(ADO、ADO.NET、ODBC、JDBC) おさお・ライ バーを使用して接続することができます。データベース接続パフォーマンス、およびパライバーの機能と制限は、MapForce 外で構成された選択されたドライバー、(該当する場合)データベースクライアント・ソストウェア、および追加接続パラメータなどによって異なります。

6.2.1.10.1 Firebird (JDBC) への接続

このサンプルではFirebird データベースサーバーへの接続のサンプルの命令にていて説明されています。

必要条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がんみールされている必要があります。 これは Oracle JDK おけま Oracle OpenJDK などのオープノノースビルドであることができます。MapForce は次のロケー ションから Java Virtual Machine (JVM) への ひを次の順序で決定します: a) アプルケーション オプション内で設定したカスタ ム JVM / 次 Java 設定を参照してくたさい。b) Windows レジストリ内で見つかすこ JVM / 次。c) JAVA_HOME 環境変数。
- MapForce (32ビオ、64ビオ)のプラオフォームがJRE /JDK のプラオフォームに一致することを確認してくたさい。

- Firebird JDBC ドライバーが使用中のオペレーティングシステムで使用可能である必要があります(データベースへの接続を可能 にする.jar ファイルのフォームをとます)。ドライバーはFirebird Web サイトからダウノロードすることができます (<u>http://www.firebirdsql.org/</u>)。このサンプルは、Jaybird 2.2.8 を使用します。
- データベース接続の詳細がおます:ホスト、データベースパスまけはエイリアス、ユーザー名、おとび、マワード。

JDBC を使用して Firebird に接続する

- 1. データベース接続ウィザードを開始します.
- 2. 「JDBC 接続」をクトックします。
- 3. 「クラス・ス」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミコレンで区切られた.jar ファイル マのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることがで きます: C:\jdbc\firebird\jaybird-full-2.2.8.jar。.jar ファイル やをオペレーティングシステムのCLASSPATH 環境変 数に追加した場合、「クラス・ペ」テキストボックスを空白のままにすることができます。(次も参照: <u>CLASSPATH の構成</u>)。
- 4. 「ドライバー」ボックスから org.firebirdsql.jdbc.FBDriver を選択します。このエトリは、有効な.jar ファイル なが、「クラ ス・ス・テキストボックス内、お当ま、オペレーティングシステムのCLASSPATH 環境変数内で検出された場合のみ使用することが できます(前のステップを参照してください)。

JDBC 接続	
接続文字列を入力して 接続ボタンをクリックしま	、有効な JDBCドライバーを選択(または手動で入力)してください。その後 す。
クラスパス:	
ドライバー:	org.firebirdsql.jdbc.FBDriver 👻
ユーザー名:	prod_admin
パスワード:	•••••
データベース URL:	jdbc:firebirdsql://firebirdserv/COMPANY
	•
	接続(C) 閉じる(L)

- 5. 対応するテキストボックスに、データベースへのユーザー名と、スワードを入力します。
- 6. URL テキストボックスに、データベースサーバーに適用することのできる、イライトされた値を置き換える、データベースサーバーの接続文字列を入力します。

jdbc:firebirdsql://<host>[:<port>]/<databese path or alias>

7. 「接続」をクリックします。

6.2.1.10.2 Firebird (ODBC) への接続

このサンプルでは Linux サーバー上で実行される Firebird 2.5.4 データベースへの接続について説明されています。

必要条件:

- Firebird データベースサーバーは クライアントからのTCP/IP 接続を受け入れるように構成されています。
- Firebird ODBC ドライバーを使用中のオペレーティングシステムニインストールする必要があります。このサンプルは、を使用します。Firebird ODBC ドライバー・バージョン 2.0.3.154 は、Firebird ウェブサイト(<u>http://www.firebirdsql.org/</u>)から ダウムロードすることができます。
- Firebird クライアトを、使用中のオペレーティングシステムニインストールはる必要が数はます。Firebird 2.5.4 クライアトのためのスタイドアロンのインストーラーが存在しない点に注意してくたさい。おこ、クライアトは、サーバーのインストール・ペクの一部です。Firebird サーバーインストールパックをFirebird ウェブサイト(<u>http://www.firebirdsql.org/</u>)からダウムードすることができます。ダウムードの際、「スーパークランク/クランククオオはスーパーサーバークオタの、Windows 実行ファイルリインストール・サーバーンール集し]を選択してくたさい。クライアトトファイルのみをインストールする場合、「最小クライアントインストール・サーバーンール無し]を選択します。

重要点:

- Firebird ODBC ドライバーおよび クライアント (32 ビル おけよ64 ビル)のプラルフォームは、 MapForce のプラルフォームに適応する必要があります。
- Firebird ケライアトの デジョンは 接続されている Firebird サーバーの デジョンに適応する必要があます。
- データベース接続の詳細 サーバー上のサーバーホスト名 おはIP アドレス データベースパス(おはエイリアス)、ユーザー名、およびパスワード。

ODBC を使用して Firebird に接続する

- 1. データベース接続ウィザードを開始します。
- 2. [ODBC 接続] をクリックする。
- 3. [ユーザーDSN] を選択して、(まけよ、管理者の特権がある場合、 [システムDSN] を選択して、)[追加 뽜] をクルクします。

ODBC DSN を作成		×
ドライバーを選択し、作り	成する DSNの種類をユーザーかシステムから選択してください。	
Firebird/InterBase(r) driver	▼	
	ユーザー DSN(U) システム DSN(S) キャンセル	(C)

4. Firebird ドライバーを選択して、をクリックします。そして [ユーザー DSN] (前のステップでの選択により、ませよ、[システム DSN])。Firebird ドライバーがリスト内に存在しない場合、使用中のオペレーティングシステムにインストールされていることを確認 してください (使用可能な ODBC ドライバーの表示 を参照)。

Firebird ODBC Setup		.
Data Source Name (DSN)		Driver
firebird_dsn		IscDbc 🗸
Description		
Connects to the Firebird database	on a remote Linux serve	r.
Database		
firebirdserv:products		Browse
Client		
C:\Program Files\Firebird\Firebird_	2_5\bin\fbclient.dll	Browse
Database Account	Password	Role
PROD_ADMIN	•••••	
Character Set		
NONE -	Services	Test connection
Options Transaction read (default write) nowait (default wait) Lock Timeout Dialect	Extended i Quoted sensitiv autoque Set null fie	dentifier properties identifiers e identifier oted identifier eld SCHEMA.
	K Cance	Help

5. データベース接続の詳細を入力します:

データソース名 (DSN)	作成する詳細名とデータノーフを入力します。	
<i>〒-タベ-</i> ス	サーバーホスト名 おけまIP アドレスをコロンとデータベース エイリアズ おけまり ない 後に入力します。このサンプルでは、ホスト名は、firebirdserv、データベースエ リアスは、products です:	
	firebirdserv:products	
	データベースエイリアスを使用することにより、サーバー側上では、データベース管理者 は、エイリアス製品を構成し、サーバー上の実際のFirebird (.fdb) データベースファ イルをポイントします(詳細は、Firebird のドキュメントを参照してくたさい)。	
	ホスト名の代わりに サーイー・IP アドレス エイリアスの代わりに スを使用することも できます。 ですから 以下のサンプル接続の文字列は有効です:	
	firebirdserver:/var/Firebird/databases/butterflies.fdb 127.0.0.1:D:\Misc\Lenders.fdb データベースがWindows マシン上に存在する場合、[参照] をクレックして、 Firebird (.fdb) データベースファイルを直接選択します。	

クライアント	fbclient.dll ファイルへの びを入力します。デフォルトでは、これはFirebird インス トールディレオ・リのサブディレオ・リです。
データベース アカウト	データベース管理者(このサンプルの場合 PROD_ADMIN)により与えられたデータベ ースユーザー名を入力します。
ノなワード	データベース管理者により与えられたデータベースパマワードを入力します。

6. 「OK」をクルクします。

6.2.1.10.3 IBM DB2 (JDBC) への接続

このサンプルでは、JDBCを使用してIBM DB2 データベースサーバーは接続する手順のサンプルについて説明されています。

必要条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がインストールされている必要があります。 Oracle JDK おは Oracle OpenJDK などのオープンノースビルドであることができます。MapForce は次のロケーションから Java Virtual Machine (JVM) への やを次の順序で決定します: a) アプルケーションオプション内で設定したカスタム JVM パス Java 設定を参照してくたさい。b) Windows レジストリ内で見つかすこ JVM パス。c) JAVA_HOME 環境変数。
- MapForce (32-ビオ、64-ビオ) のプラオフォームがJRE/JDK のプラオフォームに一致することを確認してくたさい。このサンプルはOracle のOpenJDK 11.0 64-bitを使用し、その結果 MapForce の64 ビオバージョンを使用します。
- (データベースへの接続を与える1つ、おけよ、複数の.jar ファイル) JDBC ドライバーが使用中のオペレーティングシステムで使用 可能である必要があります。IBM Data Server Client バージョン 10.1 (64-ビット) をインストール後使用することのできる JDBC ドライバーをこのサンプルは使用しています。JDBC ドライバーをインストールするオークはは、通常、インストールを選択してく ださい。おけよ、インストールウィザードでこのオプションを明示的に選択してください。

/ DB	2 Setup - IBM Da	ta Server Client - DE	32COPY1			×
Se	lect the featu	ıres to install				
I	o customize which	features are installe	d, click on icons in the f	ollowin	ıg list.	
Γ		A Data Canvas Clinat			Feature Description	1
		M Data Server Client Client support Base client Spatial Ex- DB2 LDAF DB2 LDAF Interface V IB V IB V OI V OI V OI V SO	at support ttender dient > support s M Data Server Provider BC Support DBC Support LE DB Support QLJ Support	~	JDBC Support allows Java samples, not containing embedded SQL (SQLJ), to be built and run using the JDBC driver.	
Ľ	Testallation Colder	_				·
	Installation folder	ſ				
	Directory	C:\Program Files\IB	M\SQLLIB\		<u>C</u> hange	
		Space required:	707 MB		<u>D</u> isk Space	
InstallS	nield					
			< <u>B</u> ack		Next > Cancel Help	

デフォルトのインストール Vを変更して、ない場合、インストール後必要とされる.jar ファイルはC: (Program Files/IBM/SQLLIB/java デルクトリ内で見つったことができます。

以下のデータベース詳細がと要しなります。ホスト、データベースポート、ユーザー名、および、パマワード。

JDBC を使用して IBM DB2 に接続する

- タベース接続ウィザードを開始します 1. デー
- 2. 「JDBC 接続」をクルクします。 3. 「クラス な」の横に、データベースへの接続を与える.jar ファイルへの なを入力します。このサンプルは C:\Program Files\IBM\SQLLIB\java\db2jcc.jar を参照します。データベースサーバーバージョンによりdb2jcc4.jar ドライバーを参照 する必要がある場合かがます。ドライバーの互換性に関してはIBMドキュメントを参照してくたさい(http://www-<u>01.ibm.com/support/docview.wss?uid=swg21363866</u>)。.jar ファイル やをオペレーティングシステムの CLASSPATH 環境変数に追加している場合クラス・ステキストボックスを空にしておくことができます(CLASSPATHの構成を 参照してたれ、
- 4. 「ドライバー」ボックスから com.ibm.db2.jcc.DB2Driver を選択します。このエトリは 有効な.jar ファイル マが、「クラ スパス」テキストボックス内、まけま、オペレーティングシステムのCLASSPATH 環境変数内で検出された場合のみ使用することが できます(前のステップを参照してくたきい)。

クラスパス:	C:\Program Files\IBM\SQLLIB\java\db2jcc.jar
k54/1+:	com.ibm.db2.jcc.DB2Driver ~
ユーザー名:	username
パスワード	•••••
ቻ−ቃペース URL :	jdbc:db2://dbserver:5000/dbname
	~

- 5. データベースユーザーのユーザー名とペワードを対応するテキストボックスに入力します。
- 6. JDBC 接続文字列をデータベース URL テキストボックスに入力します。使用中のデータベースサーバーに適用することのできる ものと接続の詳細が置き換えられていることを確認してくたさい。

jdbc:db2://hostName:port/databaseName

7. 「接続」をクルクします。

6.2.1.10.4 IBM DB2 (ODBC) への接続

このトピックではODBCを使用した、IBM DB2 への接続のサンプル手順にていて説明されています。

前提条件:

- IBM Data Server Client は、使用中のオペレーティングシステムニインストールおど構成される必要があります(このサンプルの場合はIBM Data Server Client 9.7を使用します)。インストールの手順に関しては、IBM DB2 ソストウェアにお提供されがギュメトを確認してください、IBM Data Server Client インストール後、使用中のコンピューターで ODBC ドライバー が使用可能か確認してください(使用可能な ODBC ドライバーの表示を参照)。
- データベースエイリアスを作成します。複数の作成方法があります:
 - IBM DB2 構成アシスタイから
 - IBM DB2 コマバラインプロセッサーから
 - o ODBC データノースウィザードからこの場合は、下に表示される手順を参照してくたさい)
- データベース接続の詳細ホスト、データベースユーザー名およびマワード

IBM DB2 への接続:

1. <u>データベース接続ウィザードを開始します</u>。IBM DB2 (ODBC/JDBC) を選択します。

接続ウィザード	
ソース データベースを選択し、[次へ]をクリックしてください。	
使用したいデータベースのベンダーが下のリストにない場合、AD ョンを作成してください。	Oまたは ODBCを使用してコネクシ
Microsoft Access (ADO)(A)	
Microsoft SQL Server (ADO)(S)	01109.000 (0220)
Oracle (ODBC/JDBC)(0)	
─ MySQL (ODBC)(M)	
IBM DB2 (ODBC/JDBC)(D)	
🔘 IBM Informix (ODBC / JDBC)(I)	
Sybase (ODBC)(Y)	
PostgreSQL	
○ Firebird (ODBC / JDBC)(F)	
◯ SQLite(L)	
ウィザードの構成をスキップ	
	次へ(N) > 閉じる(L)

2. 「次へ」をクリックします。

JDBCと ODBC
◎ JDBC Javaをペースにした接続では、ODBCではサポートされていないデータペ ースの機能がサポートされている場合もあります。これらの新しい機能がサ ポートされる代償として、パフォーマンスが低下します。
● ODBC ODBC 接続は、JDBC 接続に比べ高速で、リソースも必要としませんが、 (ネイティブの XML 型といった) 最新のデータベース 機能がサポートされま せん。
<前へ(B) 次へ(N) > 閉じる(L)

3. **ODBC** を選択して「次へ」をケルクします。データベースのための既知のノストを編集するようにプロンプトされると、IBM DB2 に適用することのできるドライバーを選択して(<u>前提条件</u>を参照)、「次へ」をケルクします。

ドライバーの選択: 既知のデータベースドライバーリストを編集		
ドライバー		
🗖 SQL Server		
Microsoft ODBC for Oracle		
Microsoft Access Driver (*.mdb)		
Microsoft Access-Treiber (*.mdb)		
Driver do Microsoft Access (*.mdb)		
Microsoft dBase Driver (*.dbf)		
Microsoft dBase-Treiber (*.dbf)		
Driver do Microsoft dBase (*.dbf)	=	
Microsoft Excel Driver (*.xls)		
Microsoft Excel-Treiber (*xls)		
Driver do Microsoft Excel(*.xls)		
L Microsoft Paradox Driver (*.db)		
Microsoft Paradox-Treiber (*.db)		
Driver do Microsoft Paradox (*.db)		
L Microsoft Text Driver (*.txt; *.csv)		
□ Microsoft Text-Treiber (*.txt; *.csv)		
□ Driver da Microsoft para arquivos texto (*.txt; *.csv)		
U Microsoft Visual FoxPro Driver	_	
U Microsoft FoxPro VFP Driver (*.dbf)	_	
L Microsoft dBase VFP Driver (*.dbf)	_	
U Microsoft Visual FoxPro-Treiber	-	
JETTERmer mana a Micmeatt Vienal Favern		
< 前へ(B) 次へ(N) 閉じる	(L)	

4. リストからIBM DB2 を選択して、「接続」をクリックします。(ドライバーで使用可能なリストを編集するコよ [ドライバーの編集] をクリックして、使用可能などドライバーのリストを編集します。 クリックして追加ませ」 ゴリ除する IBM DB2 ドライバーをチェックませ」はチェックの解除をします。ドライバーを選択ませ」は選択の解除をします。)

IBM DB2 に接続中	
	IBM DB2 ドライバーはどこで見つけられますか?
データベースへの接続オブションを選択して、接続をクリックしてください。 ● ドライバーを使用して新規データソースネーム (DSN) を作成:	
IBM DB2 ODBC DRIVER - IBMDBCL1	~
○ 既存のデータソースネームを使用する:	
◎ ユーザー DSN ○ システム DSN	ドライバーを編集
□ウィザードの構成をスキップ	
	< 前へ(B) 接続(C) 閉じる(L)

5. ソース名を入力します(このサンプルではDB2DSN)「追加」をクトックします。

ODBC に登録する DB2 データベースの別名を選択するか、あるいは新 規の別名を作成するために「追加」を選択してください。データ・ソース名 および記述を変更したり、デフォルトを適用することが可能です。		
データ・ソース名	DB2DSN	
データベース別名		~ 追加(A)
記述		
	OK	キャンセル

6. [データソース] タブで、データベースのユーザー名と ペワードを入力します。

CLI/ODBC 設定 - DB2DSN		×
データ・ソース TCP/IP セキュリティー・オ	プション 詳細設定	
データ・ソース名	DB2DSN	
記述	記述	
コーザー ID	iohn doe	
パスワード		
	□ パスワードの保管(S)	
	OK キャンセル	適用(A) ヘルプ

7. [TCP/IP] ダブで、デーダベース名、エイリアスの名前、ホスト名とポート名を入力して「OK」をクリックします。

データ・ソース TCP/IP セキュリティー・オプション	詳細設定
データベース名	database 1
データベース別名	alias 1
ホスト名	host 1
ポート番号	50000
□ データベースは物理的にホストまたは OS/40	
● サーバーへ直接接続(S)	
○ゲートウェイ経由でサーバーに接続(G)	
DCS //3X-9-	
	OK キャンセル 遠田(Δ) ヘルプ

8. ユーザ名とマワードを再入力して「OK」をクルクします。

データベース別名	DB2DSN ~ 追加(A)
ユーザー ID	john_doe
パスワード	•••••
□パスワードの変更(C)	
新規バスワード	
新規バスワードの検査	
接続モード	
● 共有(S)	○排他(E)
	OK キャンセル

6.2.1.10.5 IBM DB2 for i (JDBC) への接続

このサンプルではJDBC を介して IBM DB2 for i データベースサーバー 接続する方法について説明されています。

必要条件

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がイントールされている必要があります。 これは Oracle JDK おは Oracle OpenJDK などのオープンノースビルドであることができます。MapForce は次のロケー ションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプリケーション オプション内で設定したカスタ ム JVM パス <u>Java 設定</u>を参照してくたさい。b) Windows レジストリ内で見つかすこ JVM パス。c) JAVA_HOME 環境変数。
- MapForce (32 ビオ、64 ビオ) のプラオフォームがJRE /JDK のプラオフォームに一致するこを確認してくたさい。このサンプルはOracle のOpenJDK 11.0 64-bit を使用し、その結果 MapForce の64 ビオバージョンを使用します。
- JDBC ドライバー(データベースへの接続を提供する1つ、おけよ、複数の.jar ファイルりが使用中のオペレーティングシステム上で使用可能である必要があります。このサンプルは以下のオープンノースを使用します: Toolbox for Java/JTOpen バージョン 9.8 (<u>http://jt400.sourceforge.net/</u>)。パンケーンをダウノロードし、ローカルディレクトリニアンパンクすると、必要な.jar ファイルが lib サブディレクトリで使用できるよう よいます。
- 以下のデータベース接続の詳細が必要です。ホスト、ユーザー名、および、タワード

JDBC を使用して IBM DB2 for i に接続する方法:

- 1. データベース接続ウィザードを開始します。
- 2. 「JDBC 接続」をクルクします。
- 3. 「ケラス・ス」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミコロンで区切られた.jar ファイル やのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることがで きます: C:\Informix_JDBC_Driver\lib\ifxjdbc.jar。.jar ファイル やをオペレーティングシステムの CLASSPATH 環 境変数に追加した場合、「ケラス・ス」テキストボックスを空白のままにすることができます。(次も参照: CLASSPATH の構成)。
- 4. 「ドライバー」ボックスから、com.ibm.as400.access.AS400JDBCDriver を選択します。このエントリは、有効な.jar ファイルレ なが、「ケラスレ な」テキストボックス内、おけよ オペレーティングシステムのCLASSPATH 環境変数内で検出された場 合のみ使用することができます(前のステップを参照してくたさい)。

クラスパス:	C:\jdbc\jtopen_9_8\jt400.jar
15-111-1	com.ibm.as400.access.AS400JDBCDriver
ユーザー名:	username
パスワード	•••••
ም-ልペース URL:	jdbc:as400://192.0.2.0/

- 5. データベースユーザーのユーザー名となワードを対応するテキストボックスに入力します。
- 6. JDBC 接続文字列をデータベース URL テキストボックスに入力します。host をデータベースサーバーの木スト名、おけよ IP アドレスと置き換えてください。

jdbc:as400://**host**

7. 「接続」をクリックします。

6.2.1.10.6 IBM DB2 for i (ODBC) への接続

このサンプルでは、ODBCを使用してIBM DB2 for iデータベースへの接続にていて説明されています。

前提条件:

IBM System i Access for Windows は使用中のオペレーティングシステム(このサンプルではIBM System i Access for Windows V6R1M0 を使用して、ます) にインストールされる必要があります。インストールの手順は、IBM DB2 for i ソフトウェアには提供されているドキュメトを参照してください。インストール後、ODBC ドライバーの使用可能が確認してください(使用可能なODBC ドライバーの表示を参照)。

🔄 ODBC データ ソース アト	ミニストレー	ーター		×
ユーザー DSN システム DSN	ファイル DSN	ドライバー トレ	・ース 接続プール	バージョン情報
このコンピューターにインストーノ	しされている ロロ	DBC ドライバー(0	D):	
名前		,	バージョン	会社名
Driver do Microsoft Paradox (*.db)	6.03.9600.16384	Microsoft Corporatio	n
IBM DB2 ODBC DRIVER		9.07.301.326	International Busine	ss Machines Corp
IBM DB2 ODBC DRIVER - DB2C	OPY1	9.07.301.326	International Busine	ss Machines Corp
iSeries Access ODBC Driver		12.00.00.00	IBM Corporation	
Microsoft Access Driver (*.mdb)		6.03.9600.16384	Microsoft Corporatio	n
Microsoft Access Driver (*.mdb, *.	accdb)	15.00.4569.1501	Microsoft Corporatio	n
Microsoft Access Text Driver (*.bd	t, *.csv)	15.00.4569.1501	Microsoft Corporatio	n
Microsoft Access-Treiber (*.mdb)		6.03.9600.16384	Microsoft Corporatio	n
Microsoft dBase Driver (*.dbf)		6.03.9600.16384	Microsoft Corporation	n j
•	111			Þ
ODBC ドライバー	を使用すると、 オスニレポラキ=	ODBC が有効な ます 新しいドライ	ネブログラムで OD	BC データソー
スから1月報を4×19	・9 つことか じきさ プ プログラムを1	を9。和しいトラ1 使用してください。	ハーゼインストール	1996667
	ОК	キャンセル) 適田(A)	<u>∧</u> ⊪,
		11200		

- データベース接続の詳細 データベースサーバーの I.P. アドレス データベースユーザー名、および マワード
- i Navigator を実行して、新規接続を作成するシャザードの手順を踏んで新規接続を作成してくたさい、システムの指定をプロンプトされると、データベースサーバーのI.P. アドレスを入力してくたさい。接続を作成した後、検証する接続をクリックして、ファイル > 診断 > 接続の検証 を選択) ことが奨励されます。接続エラーが発生した場合、データベースサーバー管理者に連絡してくたさい。

IBM DB2 for i への接続:

- 1. データベース接続ウィザードを開始します.
- 2. [ODBC 接続] をクルクします。
- 3. [ユーザーDSN]をクルクします。(おけよ [システムDSN] おけよ後に続く手順が同じ [ファイルDSN]をクルクします)。
- 4. 「追加」 🏞 をクリックします。
- 5. [iSeries Access ODBC Driver] をパントから選択します。[ユーザーDSN] (おけよ 該当する場合 [システム DSN]) をクリックします。

ODBC DSN を作成		×
ドライバーを選択し、作	F成する DSNの種類をユーザーかシステム	から選択してください。
iSeries Access ODBC Driv	/er	-
	ニーザー DSN(U) システム D	SN(S) キャンセル(C)

6. データノース名を入力して、システムコンボボックスから接続を選択します。このサンプルでは、データノース名は iSeries DSN でシ ステムは 192.0.2.0 です。

ieneral	Server	Data Types	Packages	Performance	Language	Catalog	Conversions	Diagnostic
		,,						
Data s	ource na	me:						
iSeries	sDSN							
Descri	ption:							
Syster	miAcces	s for Windows	ODBC data	source				
System	n:							
192.0	.2.0		*	Connectio	n Options			
				ОК	Can	cel	<u>A</u> pply	Help

- メモ ODBC データノースを IBM DB2 for i database に追加する場合、クエリのタイムアナトを有効化するデフォルトのフラグか設定されます。この設定は、マメピングファイルを正確にエードするために、MapForce のために無効化されている必要があります。 設定を無効化するには、パフォーマンス タブから、詳細をクリックして、クエリのタイムアナトの許可 チェックボックスをクリアします。
 - 7. [接続オプション] をクトックします。[指定されたユーザー ID を使用] を選択して、データベースユーザー(このサンプル内では DBUSER) の名前を入力します。

 Use Windows user name 			
Use the user ID specified below			
DBUSER			
◯ None			
🔘 Use System i Navigator default			
🔘 Use Kerberos principal			
Never prompt for SQLConnect			
Security			
O Do not use Secured Sockets Laye	(SSL)		
O Han Conversed Condustral away (CCL).			
O Use Secured Sockets Layer (SSL)			

- 8. 「OK」をクリックします。DSN内のノストで新しいデータノースを使用することができます。
- 9. 「接続」をクリックします。
- 10. データベースのユーザー名とマワードをプロンプトされると入力して「OK」をクリックします。

6.2.1.10.7 IBM Informix (JDBC) への接続

このサンプルでは、JDBCを使用してIBM Informix データベースサーバーは接続する手順のサンプルこして説明されています。

前提条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がインストールされている必要があります。 これは Oracle JDK おは Oracle OpenJDK などのオープンノースビルドであることができます。MapForce は次のロケー ションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプリケーション オプション内で設定したカスタ ム JVM パス <u>Java 設定</u>を参照してくたさい。b) Windows レジストリ内で見つかった JVM パス。c) JAVA_HOME 環境変数。
- MapForce (32ビナ、64ビナ)のプラナフォームがJRE /JDK のプラナフォームに一致するとを確認してくたさい。
- (データベースへの接続を与える1つ、おけよ、複数の.jar ファイル) JDBC ドライバーが使用中のオペレーティングシステムで使用 可能である必要がおります。この例では、IBM Informix JDBC ドライバー・バージョン 3.70 が使用されています。ドライバーの インストールの説明に関しては、ドライバーゴ付随するドキュメント、おけよ「IBM Informix JDBC ドライバンログラマーズガイ ド」を参照してくたさい。。
- データベース接続の詳細ホスト、サーバーの名前、データベースポトユーザー名、および、マワード

JDBC を使用して IBM Informix に接続する

- 1. データベース接続ウィザードを開始します.
- 2. 「JDBC 接続」をクリックします。
- 3. 「ケラス・ス」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミコロンで区切られた.jar ファイル やのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることがで きます: C:\Informix_JDBC_Driver\lib\ifxjdbc.jar。.jar ファイル やをオペレーティングシステムのCLASSPATH 環 境変数に追加した場合、「ケラス・マ、Jテキストボックスを空白のままにすることができます。」(次も参照: <u>CLASSPATH の構成</u>)。
- 4. 「ドライバー」ボックスから com.informix.jdbc.lfxDriver を選択します。このエトリは 有効な jar ファイル なが、「クラ ス ペリテキストボックス内、 お当よ オペレーティングシステムの CLASSPATH 環境変数内で検出された場合のみ使用することが できます(前のステップを参照してください)。

クラスパス:	C:\Informix_JDBC_Driver\lib\ifxjdbc.jar
ドライバー:	com.informix.jdbc.lfxDriver 👻
ユーザー名:	dbuser
パスワード:	•••••
データベース URL :	jdbc:infomixsqli;jpst:port/MyDatabase:INFOMIXSERVER=MYServer
	接続(C) 閉じる(L)

- 5. データベースへのユーザー名とアワードを対応するテキストボックスに入力します。
- 6. ハイライトされた値を、使用するデータベースサーバーゴ適用する値と置き換えて、データベースURL テキストボックス内にデータベ ースサーバーへの接続文字列を入力します。

jdbc:informix-sqli://hostName:port/databaseName:INFORMIXSERVER=myserver;

7. 「接続」をクリックします。

6.2.1.10.8 MariaDB (ODBC) への接続

このサンプルでは、ODBCを使用したMariaDBデータベースへの接続方法について説明されています。

前提条件:

- MariaDB コネクタ/ODBC (<u>https://downloads.mariadb.org/connector-odbc/</u>) がインストール にわてい る必要か あます。
- 次のデータベース接続詳細が存在します:ホスト、データベースポート、ユーザー名、および、パマワード。

ODBC を使用した MariaDB への接続の方法:

- 1. データベース接続ウィザードを開始します。
- 2. 「MariaDB (ODBC)」を選択して、「次へ」をクトックします。

MariaDBに接続中				
	MariaDB ドライバーはどこで見つけられますか?			
データベースへの接続オブションを選択して、接続をクリッ ● ドライバーを使用して新規データソースネーム (DSN) *	クしてください。 を作成:			
MariaDB ODBC 3.1 Driver	~			
○既存のデータソースネームを使用する:				
● ユーザー DSN ○ システム DSN	ドライバーを編集			

- 3. 「ドライバーを使用して新規のデータソース名 (DSN)を作成する」を選択し、「MariaDB ODBC 3.0 Driver」を選択 します。このようよドライバーがリスト内で使用できない場合、「ドライバーの編集」をクトックし、使用することのできる MariaDB ドライバーを選択します (リストロコ 使用中のオペレーティングシステム上にインストールされているすべての ODBC ドライバーが含 まれています)。
- 4. 「接続」をクックします。

Create a new Data So	urce to MariaDB		×
	Welcome to the Maria	aDB ODBC Data Source Wizard!	
	This wizard will help you to create an ODBC data source that you can use to connect to a MariaDB server.		
	What name do you w	ant to use to refer to your data source ?	
	Name:	mariadb_connection	
	How do you want to	describe the data source ?	
	Description:		
	1.0		
	< Pre	vious Next > Cancel Help	

5. 名前を入力し、任意で、このODBC データノースを将来識別できる詳細を与えます。

Create a new Data So	ırce to MariaDB		×
	How do you want to connect to Ma	ariaDB	
	TCP/IP Server N	ame: demoserver	
	○ Named Pipe	Port: 3306	
	Please specify a user name and pa	ssword to connect to MariaDB	
	User name: demouser		
	Password:	<u>T</u> est DSN	
	Please specify a user name and pa	ssword to connect to MariaDB	
	Database: mydatabase	~	
	< Previous	Next > Cancel Help	

6. データベース接続資格情報(TCP/IP サーバー、ユーザー、パスワード)を入力し、データベースを選択して、「Test DSN」をク リックします。接続に成功すると、メッセージが表示されます:



- 7. 「次へ」をクリックしてウィザードを完了します。他の「ラメータが必要しなる場合がありますが(例えば、安全な接続を使用して、 MariaDB は接続する場合は、SSL 証明書が必要しなります)。
- メモ データベースサーバーがJモートの場合、サーバー管理者が遠隔の接続を使用中のマシンのIP アドレスから受け入れるように構成 する必要があります。

6.2.1.10.9 Microsoft Access (ADO) への接続

Microsoft Access データベース 接続する コよ 既存の Microsoft Access データベース 接続する て説明されているとおり、データベ ースファイルのナーダのフィザード とブラフザーの手順を踏みます。別の方法としては、このトピックて説明されているとおり、明確的に ADO 接続 をセット アップすることです。 このアプローチ1よ、データベースが ペワードで保護されている場合、役に立ちます。

ODBC 接続を使用して、Microsoft Access に接続することもできますが このシナリオゴ お限度があるため、奨励されません。

パワードで保護されている Microsoft Access データベースに接続する

- 1. データベース接続ウィザードを開始します.
- 2. [ADO 接続] をクルクします。
- 3. 構築]をクリックします。

プロバイダー 接続 詳細設定 すべて	
接続するデータを選択します:	
OLE DB プロバイダー	
Microsoft Jet 4.0 OLE DB Provider Microsoft Office 12.0 Access Database Engine OLE DB Pro Microsoft Office 15.0 Access Database Engine OLE DB Pro Microsoft Office 16.0 Access Database Engine OLE DB Pro Microsoft OLE DB Provider for Analysis Services 11.0 Microsoft OLE DB Provider for Indexing Service	
Microsoft OLE DB Provider for ODBC Drivers Microsoft OLE DB Provider for Oracle Microsoft OLE DB Provider for Search Microsoft OLE DB Provider for SQL Server Microsoft OLE DB Simple Provider MSDateShare	
OLE DB Provider for Microsoft Directory Services	-
次へ (N)>>	

4. [Microsoft Office 15.0 Access Database Engine OLE DB Provider] を選択して、「次へ」をクルクします。

プロバイダー 接続 詳細設定 すべて
ODBC データに接続するために、次の項目に情報を入力します: 1. データ ソースを指定します: ③ データ ソース名を使用する(D): ▼ 更新(R)
● 接続文字列を使用する(O) 接続文字列(C):
2. サーバーへのログオンに必要な情報を入力します:
ユーザー名(N): Admin
パスワード(P):
📝 パスワードを空にする(B) 🛛 「パスワードを保存する(S)
3. 使用する初期カタログを入力します(I):

- データノースボックス内でUNC フォーマナのMicrosoft Access ファイルへの やを入力します。例えば myserver かサー バーの名前であり mynetworkshare か共有するネナワークの名前である箇所での\\myserver\ \mynetworkshare\Reports\Revenue.accdb。
- 6. 全て タブの Jet OLEDB: Database Password パマワード プロパティをダブルクトックレ、データベースパマワードをプロパティ の値として入力します。

プロパティの値を編集	
プロパティの説明	
Jet OLEDB: Database Password	
プロパティの 値(V) 	
値のリセット(R)	OK キャンセル

メモ お接続できない場合、ワーキンググループ情報ファイル(System.MDW)を検索して、ユーザープロフィールに適用できるように します。Jet OLEDB: System database プロパティの値をSystem.MDW ファイルのパン設定します。

6.2.1.10.10 Microsoft SQL Server (ADO) への接続

このサンプルでは、ADOを使用したSQL Server データベースへの接続方法について説明されています。 Https://docs.microsoft.com/en-us/sql/connect/oledb/download-oledb-driver-for-sql-server?view=sql-serverver15 でダウンロード可能な奨励される Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)を使用する場合に 適用される説明です。

これらの説明を読む前に、ワークステーションに上記のプロ・イダーカダウンロードされインストールされていることを確認してくたさい。ADO プロ・イダーはMapForce(32ビットませょ64ビット)のプラナトフォームに一致する必要かあることに注意してくたさい。

SQL Server Native Client (SQLNCLI) おけは Microsoft OLE DB Provider for SQL Server (SQLOLEDB) な どの他のADO プロ・イダーを使用する場合、手順は同様ですが、これらのプロ・イダーは奨励されていません。 奨励されていないプロ・イダ 一への接続を成功させるこは、 SQL Server データレンプロ・ティのセットアップで説明されている追加接続プロ・ティを設定する必要があり ます。

Microsoft OLE DB Provider for SQL Server (SQLOLEDB) は共通テーブル式 (CTE) とネスト された SELECT ステート メント などの複雑なクロノの デメーター・ シディング ご買する問題があることか知られています。

SQL サーバーに接続する方法:

- 1. データベース接続ウィザードを開始します。
- 2. [Microsoft SQL Server (ADO)] を選択して、[次へ] をクトクします。使用することのできるADO プロ・イダーのノストが 表示されます。このサンプルではMicrosoft OLE DB Driver for SQL Server が使用されています。リストに存在しない 場合、上記の通りコンピューターイ:接続されていることを確認してください。



3. 「次へ」をクリックします。データレンクプロ ティダイアログボックスか表示されます。
| 🗊 Data Link Properties | × |
|--|--------------------|
| Provider Connection Advanced All | |
| 1. Select or enter a server name: | |
| ✓ <u>R</u> efree | sh |
| 2. Enter information to log on to the server: | |
| SQL Server Authentication $\qquad \lor$ | |
| Ser <u>v</u> er SPN: | |
| User name: | |
| Password: | |
| ☑ Blank password | assword |
| Use strong encryption for data Trust server ce | rtific <u>a</u> te |
| 3. Select the <u>d</u> atabase: | ~ |
| Attach a database file as a database name: | |
| | |
| Using the <u>fi</u> lename: | |
| | |
| Change Password Test Connection | |
| | |
| OK Cancel | Heip |

- 4. データベースサーバーの名前を選択、おけよ入力します。例えば、SQLSERV01。名前を持つSQL Server インスタンスは接続する場合、サーバー名はSQLSERV01、SOMEINSTANCE になります。
- 5. データベースサーバーがWindowsドメイン上の認証されているユーザーの接続を許可している場合、「Windows 認証」を 選択してくたさい。それ以外の場合は、[SQL Server 認証]を選択し、「空の ペワード」 チェックボックスをクリアし、対応する ボックスにデータベース証明書を入力します。
- 6. パマワードの保存の許可 チェックボックスを選択し、接続するデータベースを選択します(このサンプル内では、Nanonullです)。

🗊 Data Link Proper	ties >
Provider Connection	Advanced All
1. Select or enter a se	erver name:
SI.M.1.1999	✓ <u>R</u> efresh
2. Enter information to	log on to the server:
SQL Server Authe	entication V
Ser <u>v</u> er SPN:	
User <u>n</u> ame:	altova_user
Password:	•••••
Blank password	Allow <u>s</u> aving password
Use strong enc	ryption for data Trust server certificate
3. Select the <u>d</u> ata	base: Nanonull ~
◯ Attac <u>h</u> a databa	ase file as a database name:
Using the filena	ame:
Change Pass	word <u>T</u> est Connection
	OK Cancel Help

- 7. この時点で接続をテストする場合は [接続のテスト]をクリックします。これは任意の奨励されるステップです。
- 8. [OK]をクリックします。

6.2.1.10.11 Microsoft SQL Server (ODBC) への接続

このサンプルでは、ODBCを使用したSQL Server データベースへの接続方法について説明されています。

必要条件:

 Microsoft ODBC Driver for SQL Server をMicrosoft Web サイトからダンロードしてインストールします。 https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server を参照してくたさ https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server">https://docs.microsoft.com/en-us/SQL/connect/odbc/download-odbc-driver-for-sql-server を参照してくたさ https://docs.microsoft ODBC Driver 17 for SQL Server を SQL Server 2016 データベース 接続するた は、このサンプルは Microsoft ODBC Driver 17 for SQL Server を SQL Server 2016 データベース 接続するた がったする 接続する SQL Server の、デジュイニより 異なる ODBC ドライバー、デジュンをダウンロードする必要かあ る場合かあります。使用中の SQL Server によりサポート される ODBC ドライバー、デジョンゴ関する情報は、システムの必要 条件を参照してくたい。

ODBC を使用しての SQL Server への接続:

1. <u>データベース接続ウィザードを開始します</u>。

- 2. [ODBC 接続] をクルクします。
- 3. [ユーザーDSN] を選択して、(おけよ 管理者の特権がある場合、 [システム DSN] を選択して、)[追加 🏞] をクルクしま す。
- 4. ドライバーをリストから選択します。インストール後にのチャディバーはリスト内に表示されることに注意してくたさい。

Create an ODBC DSN		Х
Select a Driver and click on eit you want to create.	ther User or System to determine what kind of DS	N
ODBC Driver 17 for SQL Serve	er	~
	User DSN System DSN Cancel	I

5. [ユーザーDSN] (おけよシステムDSN を作成する場合) [システムDSN]をクルクします。

MapForce か管理者とて作動することを必要とするシステムDSN の作成方法このため System DSN を作成する には、ウイザードをキャンセルし、MapForce か管理者として作動していることを確認し、上記のステップを再度実行します。

6. 名前を入力して、任意でこの接続のDの説明を入力し、接続するSQL Server (この場合 SQLSERV01)をリストから選択します。.

Microsoft SQL Server DS	N Configuration		×
SQL Server	This wizard will help y SQL Server. What name do you w	you create an ODBC data source that you can use to connect to want to use to refer to the data source?	0
	How do you want to <u>D</u> escription:	o describe the data source?	
	Which SQL Server d <u>S</u> erver:	do you want to connect to? : SQLSERV01	~
		Finish <u>N</u> ext > Cancel Help	

7. データベースサーバーがWindowsドメイン上の認証されているユーザーの接続を許可している場合、「統合されたWindows 認証」を選択してくたさい。それ以外の場合、適用可能なオプションからしつを選択してくたさい。このサンプルは SQL Server 認証...を使用しています。これはユーザー名と「スワードが関連するボックスに入力されることを必要としています。

Create a New Data Sour	ce to SQL Server	\times
8	How should SQL Server verify the authenticity of the login ID?	
SQL Server	O <u>W</u> ith Integrated Windows authentication. S <u>P</u> N (Optional):	
	○ With Azure Active Directory Integrated authentication.	
	$\textcircled{With \underline{S}QL}$ Server authentication using a login ID and password entered by the user.	
	O With Azure <u>A</u> ctive Directory Password authentication using a login ID and password entered by the user.	
	\bigcirc With Azure Active $\underline{\rm D}{\rm i}{\rm rectory}$ Interactive authentication using a login ID entered by the user.	
	Login ID: dbuser	
	Password:	
	< <u>B</u> ack <u>N</u> ext > Cancel Help	

8. オプションで デフォルト のデータベースを変更 チェックボックスを選択して、接続するデータベースの名前を入力します (このサンプルではSandbox)。

Create a New Data Sour	ce to SQL Server	×
SQL Server	✓ Change the default database to: Sandbox Mirror server: SPN for mirror server (Optional):	~
	 Attach database filename: ✓ Use ANSI quoted identifiers. ✓ Use ANSI nulls, paddings and warnings. Application intent: READWRITE 	~
	 Multi-subnet <u>f</u>ailover. ✓ <u>T</u>ransparent Network IP Resolution. Column <u>E</u>ncryption. E<u>n</u>clave Attestation Info: Use FMT<u>O</u>NLY metadata discovery. 	
	< <u>B</u> ack <u>N</u> ext > Cancel Help	

9. [次へ]をクリックして、任意で、この接続のための追加ノテメーターを構成します。

Create a New Data Sour	ce to SQL Server	×
Moger	Change the language of SQL Server system messages to: (Default)	\sim
SQL Server	Use strong encryption for data.	
	\square <u>P</u> erform translation for character data.	
	Use regional settings when outputting currency, numbers, dates and times.	
	Save long running queries to the log file:	
	C:\Users\altova\AppData\Local\Temp\QUERY.LO(Browse	
	Long query time (milliseconds): 30000	
	Log ODBC driver statistics to the log file:	
	C:\Users\altova\AppData\Local\Temp\STATS.LOG Browse	
	Connect retry count: 1	
	Connect retry interval (seconds): 10	
	< <u>B</u> ack Finish Cancel Help	

10.「完了」をクリックします。接続の詳細をリストした確認ダイアログボックスか開かれます。

ODBC Microsoft SQL Server Setup	×
A new ODBC data source will be created with the following configura	tion:
Microsoft ODBC Driver for SQL Server Version 17.05.0002 Data Source Name: ds Data Source Description; Server: Use Integrated Security: No Database: (Default) Language: (Default) Data Encryption: No Trust Server Certificate: No Multiple Active Result Sets(MARS): No Translate Character Data: Yes Log Long Running Queries: No Log Driver Statistics: No Use Regional Settings: No Use ANSI Quoted Identifiers: Yes Use ANSI Null, Paddings and Warnings: Yes	<
Test Data Source OK Cance	el

11. [OK]をクリックします。構成された通りユーザーませまシステムデータノースのリスト内でデータノースか表示されます。例

System DSN	O Build a connection string
User DSN	
) File DSN	~ ~
ata Source Name	Driver

6.2.1.10.12 MySQL (ODBC) への接続

このサンプルではODBC ドライバーを使用して、Windows マンクカら MySQL データベース へ接続する方法について説明されています。Windows からよ MySQL ODBC ドライバーを入手することができませんので、個別にダウンロードおよびインストールする必要があります。このサンプルでは、MySQL コネクタイのBC 8.0 を使用しています。

前提条件:

- MySQL ODBC ドライバーをオペレーティングシステム上にインストールする必要があります。使用するデータベースサーバー、 ジョンのオーカニジ目に行いたい、インジョンのオーカのMySQL ドキュメントを確認してくたさい (https://dev.mysgl.com/downloads/connector/odbc/を確認してくたさい)。
- データベース接続の詳細ホスト、データベースユーザー名および、タワード

64-ビナプラナフォームのためにMySQL Connector/ODBC をインストール ている場合、64-ビナプラナフォームのために MapForce をインストール てください。

ODBC を介して MySQL に接続する方法:

- 1. データベース接続ウィザードを開始します。
- 2. MySQL (ODBC) を選択して、「次へ」をクルクします。

Connecting to MySQL	
	Where can I find MySQL drivers?
	to the determined still General
 Create a new Data Source Name (DSN)) with the driver:
Create a new Data Source Name (DSN MySQL ODBC 8.0 Unicode Driver) with the driver:
Select an option how you wish to connect Oreate a new Data Source Name (DSN MySQL ODBC 8.0 Unicode Driver Use an existing Data Source Name:) with the driver:

3. [ドライバーを使用して新しいデータソース名 (DSN) を作成] を選択して、MySQL ドライバーを選択します。MySQL ド ライバーがリストロない場合は、[ドライバーを編集] をクリックして、使用可能な MySQL ドライバーを選択します(使用中のオ ペレーティングシステムコインストールされているすべての ODBC ドライバーがリストされています)。

MapForce 64 ビナをインストールすると、64 ビナ ODBC ドライバーがリスト内に表示されます。それ以外の場合、 32 ビナ ODBC ドライバーが表示されます。使用可能な ODBC ドライバーの確認も参照してくたさい。

4. 「接続」をクリックします。

MySQL Connector/ODBC Data Source Configuration	×
MysqL Connector/ODBC	
Connection Parameters	
Data Source Name: MySQL Data Source	
Description:	
TCP/IP Server: server01 Port: 3306	
O Named Pipe:	
User: dbuser	
Pass <u>w</u> ord:	
Data <u>b</u> ase: dbproducts ~ <u>T</u> est	
Details >> OK <u>C</u> ancel <u>H</u> e	lp

- 5. データノース名ボックスの中に、今後このODBC データノースを認識するために役に立つ詳細を入力します。
- 6. データベース接続資格情報 (TCP/IP サーバー、ユーザー、パマワード) を入力して、データベースを選択し、「OK」をクトックします。
- メモ データベースサーバーがJモートの場合、サーバー管理者が使用中のエンピューターのIP アドレスからのJモート接続を承諾するよう に構成される必要があります。また、[詳細>>]をクリックすると使用することのできる追加パラメーターが表示されます。デフォルト の値を変更する前に、ドライバーのドキュメートを確認してくたさい。

6.2.1.10.13 Oracle (JDBC) への接続

JDBC イターフェイスを使用して、ケライアトマンから、Oracle データベースサーバーへの接続の方法を説明します。接続は、Oracle Web サイトから使用することのできる、Oracle Instant Client Package (Basic) を使用する、Java 接続として作成されます。接続の型の利点は、Oracle Instant Client Package により与えられる Java 環境と.jar ライブラルのみを必要とし、更に複雑なデータベースクライアトをインストールする手間を省くことができる必点です。

前提条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がんみールされている必要があます。 これは Oracle JDK おは Oracle OpenJDK などのオープノノースビルドであることができます。MapForce は次のロケー ションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプルケーション オプション内で設定したカスタ ム JVM / 次 Java 設定を参照してくたさい。b) Windows レジストリ内で見つかけこ JVM / 次。c) JAVA_HOME 環境変数。
- MapForce (32ビオ、64ビオ)のプラオフォームがJRE /JDK のプラオフォームに一致することを確認してくたさい。

- Oracle Instant Client Package (Basic) が使用中のオペーティングシステムで使用可能である必要があります。パッケージは、Oracle の公式 Web サイトからダウロードすることができます。このサンプルは、Windows 32-ビットのための Oracle Instant Client Package バージョン12.1.0.2.0、そして結果的にOracle JDK 32-ビットを使用します。
- データベース接続の詳細があります:ホスト、データベースパスませまエイリアス、ユーザー名、およびパマワード。

クライアント パッケージ を使用して Oracle に接続する

- 1. データベース接続ウィザードを開始します。
- 2. 「JDBC 接続」をクリックします。
- 3. 「クラス・マ」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミコロンで区切られた.jar ファイル マのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることができ ます: C:\jdbc\instantclient_12_1\ojdbc7.jar。.jar ファイル やをオペレーティングシステムの CLASSPATH 環境変数 に追加した場合、「クラス・マ」テキストボックスを空白のままにすることができます。(次も参照 CLASSPATH の構成)。
- 4. 「ドライバー」ボックスから、oracle.jdbc.OracleDriver おけま oracle.jdbc.driver.OracleDriver を選択します。 こ のエトリは、有効な、jar ファイル マが、「クラス マ」テキストボックス内、おけま、オペレーティングシステムの CLASSPATH 環境変数内で検出された場合のみ使用することができます(前のステップを参照してくたきい)。
- 5. データベースへのユーザー名とマワードを対応するテキストボックスに入力します。

クラスパス:	C:\jdbc\instantclient_12_1\odbc7.jar
ドライバー:	oracle.jdbc.OracleDriver
ユーザー名:	jphndoe
パスワード:	•••••
データベース URL:	jdbc:oracle:thin:@//ora12c:1521:orcl12d
_	
	1支前に(C) (お)しる(L)

6. ハイライトされた値を、使用するデータベースサーバーに適用する値と置き換えて、データベース URL テキストボックス内にデータベ ースサーバーへの接続文字列を入力します。

jdbc:oracle:thin:@//host:port:service

7. 「接続」をクルクします。

6.2.1.10.14 Oracle (ODBC) への接続

このサンプルでは、ローカルオペレーティングシステム上にインストールされている Oracle データベースクライアントを介して、ネトワークマシン上でMapForce から Oracle データベースサーバーへの接続のよく使用されている状況について説明されています。

このサンプルでは、MapForce内のデータベース接続ウィザードを使用して、ODBCデータノース(DSN)のセトアップについて説明されています。既にDSN が作成されている場合は Windows 内のODBC データノース管理者から直接作成しプロンプトされたう選択する事も可能です。ODBC データノースの詳細に関しては、ODBC 接続のセトアップを参照してください。

必要条件:

- (ODBC Oracle ドライバーを含む) Oracle データベースクライアントが使用中のオペレーティングシステムでインストール および 構成されてしたければいません。Oracle データベースクライアントのインストールおよび構成方法はOracle ソストウェアニ付随 するドキュメントを参照してくたさい。
- tnsnames.ora ファイルの みは Oracle ホームディレケールされている場所によます。 Oracle データベースクラ イアト 11.2.0 に関しては、デフォルト Oracle ホームディレケールの みお以下の通りです:

```
ORCL =
  (DESCRIPTION =
   (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = server01) (PORT = 1521))
  )
  (CONNECT_DATA =
   (SID = orcl)
   (SERVER = DEDICATED)
  )
)
```

tnsnames.ora ファイルの ぷはOracle ホームディレケリがインストールされている場所によります。Oracle データベースクラ イアナト 11.2.0 に関しては、デフォルト Oracle ホームディレケトリの ぷは以下の通りです:

C:\app\username\product\11.2.0\client_1\network\admin\tnsnames.ora

接続の詳細を貼り付け保存する、おさま使用可能であれば Oracle ネット構成アシスタントウィザード を実行することにより tnsnames.ora ファイルに新規エトリを追加することができます。

ODBC を使用してのOracle への接続:

- 1. データベース接続ウィザードを開始します.
- 2. [Oracle (ODBC / JDBC)] を選択して、「次へ」をクルクします。

JDBC & ODBC
○ JDBC Javaをベースにした接続では、ODBCではサポートされていないデータベ ースの機能がサポートされている場合もあります。これらの新しい機能がサ ポートされる代償として、パフォーマンスが低下します。
● ODBC ODBC 接続は、JDBC 接続に比べ高速で、リソースも必要としませんが、 (ネイティブの XML 型といった) 最新のデータベース機能がサポートされま せん。
<前へ(B) 次へ(N) > 閉じる(L)

3. [ODBC] を選択します。

Oracle に接続中	
	Oracle ドライバーはどこで見つけられますか?
データベースへの接続オプションを選択して、接続をクリックしてくださ ◎ドライバーを使用して新規データソースネーム (DSN)を作成:	<u>کر۱°</u>
Microsoft ODBC for Oracle	▼
◎ 既存のデータソースネームを使用する:	
◎ ユーザー DSN	ドライバーを編集
■ ウィザードの構成をスキップ	
	<前へ(B) 接続(C) 閉じる(L)

4. [ドライバーの編集]をクリックします。

ドライバーの選択: 既知のデータベース ドライバー リストを編集	
ドライバー Microsoft Paradox Driver (*.db) Microsoft Paradox-Treiber (*.db) Driver do Microsoft Paradox (*.db) Microsoft Text Driver (*.txt; *.csv) Microsoft Text-Treiber (*.txt; *.csv) Driver da Microsoft para arquivos texto (*.txt; *.csv) Driver da Microsoft para arquivos texto (*.txt; *.csv) Microsoft Visual FoxPro Driver Microsoft FoxPro VFP Driver (*.dbf)	
 Microsoft dBase VFP Driver (*.dbf) Microsoft Visual FoxPro-Treiber Driver para o Microsoft Visual FoxPro SQL Native Client Microsoft Access Driver (*.mdb, *.accdb) Microsoft Excel Driver (*.xls, *.xlsx, *.xlsb) Microsoft Access Text Driver (*.txt, *.csv) SQL Server Native Client 11.0 	•
< 前へ(B) 閉じる(L)

- 5. 使用する Oracle ドライバーを選択します (このサンプルでは Oracle in OraClient11g_home1です)。 Oracle ケライアントをインストール後システムで使用可能な Oracle ドライバーのノストを表示します。
- 6. [戻る] をクルクします。
- 7. [新しいデータソース(DSN)をドライバーを使用して作成]を選択して、ステップ4 で選択された Oracle ドライバーを選択します。

	Where	e can I find Oracle drivers?
elect an option how Oreate a new Dat	you wish to connect to the data a Source Name (DSN) with the o	abase and click Connect. driver:
Oracle in OraClie	nt11g_home1	~
Use an existing Da	ata Source Name:	
User DSN	System DSN	Edit Drivers
kip the configuratio	n step for wizard	

Microsoft ODBC for Oracle とう名称のMicrosoft にお提供されるドライバーを使用することは避けてください。 Microsoft はOracle にお提供される ODBC ドライバーを使用することを奨励しています。(以下参照: http://msdn.microsoft.com/en-us/library/ms714756%28v=vs.85%29.aspx)。

8. 「接続」をクルクします。

Data Source Name	Ora	cle DSN 1		OK
Description				Cancel
TNS Service Name	OB	CL		, Help
TNU DEMICE NAME			·	Test Connection
User ID				
Application Oracle	Workaro	unds SQLServer Migration]	
Application Oracle Enable Result Sets	Workaro	unds SQLServer Migration	Read-Only Connection	
Application Oracle Enable Result Sets Enable Closing Curs	Workaro	unds SQLServer Migration Enable Query Timeout 🗹 Enable Thread Safety 🔽	Read-Only Connection	
Application Oracle Enable Result Sets Enable Closing Curs Batch Autocommit I	Workaro	unds SQLServer Migration Enable Query Timeout Enable Thread Safety Commit only if all statemen	Read-Only Connection	· · · · · · · · · · · · · · · · · · ·

- 9. データノース名テキストボックスにデータノースを識別する名前を入力します(このサンプルの場合はOracle DSN 1 です)。
- 10. TNS サービス名ボックスでは、tnsnames.ora ファイルで定義されている通りの接続名を入力します。(prerequisites前提 条件を参照してくたさい)。このサンプルでは接続名 ORCL です。
- 11.「OK」をクリックします。

User Name john_doe	OK Cancel
john_doe	Cancel
	Cancer
Password	A <u>b</u> out

12. データベースのユーザー名とマワードを入力して、「OK」。をクリックします。

6.2.1.10.15 PostgreSQL (ODBC) への接続

このトピックではODBCドライバーを使用した、Windows マシンから PostgreSQL データベース への接続のサンプル手順について説明されています。Windows からよ、PostgreSQL ODBCドライバーを入手することができませんので、個別にダウンロードおよびインストールする必要があります。このサンプルでは、公式 Web サイトからダウンロードされたpsqlODBCドライバー(バージョン 11.0)を使用しています。(データベースドライバー(バージョン 11.0)を使用

メモ (ODBC ドライバー無しで直接 PostgreSQL データベースサーバー 接続するコよ 次を参照してくたさい PostgreSQL 接 続のセメアップ。

前提条件:

- psqlODBCドライバーを使用中のオペレーティングシステムにインストールする必要があります。
- データベース接続の詳細ホスト、データベースユーザー名および、ペワード

ODBC を使用した PostgreSQL への接続をセットアップする方法:

- 1. データベース接続ウィザードを開始します。
- 2. PostgreSQL (ODBC) をクリックします。
- 3. ユーザーDSN オプィンを選択します。
- 4. 「新規のDSN の作成」 ** をクトゥク、ドロップダウノストかぶライバーを選択します。 PostgreSQL ドライバーがリスト 内で使用できない場合、上記の必要条件で説明されているとおり、 PostgreSQL ODBC ドライバーカオペレーティングシステム 上にインストーリされていることを確認してくたさい。

ODBC DSN を作成	×
ドライバーを選択し、作成する DSN の種類をユーザーかシステムから選択してください	١٠
PostgreSQL Unicode	~
ューザー DSN(U) システム DSN(S)	キャンセル(C)

5. 「ユーザーDSN」をクルクします。

PostgreSQL Unico	de ODBC セットアップ	×
データソース名: (N)	PostgreSQL35W	
説明:(D)		テスト
SSL Mode:(L)	優先 ~	
サーバー名: (S)	my-postgresql-server	保存
データベース名: (b)	zoodb	キャンセル Port: 5432
既定の認証		オブション(高度な設定)
ユーザー名: (II)	dbs-user	データソース
パスワード: (w)	•••••	全体設定
PostgreS In the o	QL Ver7.3 Copyright (C) 1998-2006; Insight C riginal form, Japanese patch Hiroshi-saito)istribution Systems

6. データベース資格情報を入力し、「保存」をクリックします(これらはデータベースオーナートこと提供されています)。

接続はODBC接続のパト内で使用できるようになります。データベースは接続するには、接続をダブルクトクレ、「接続」をクトックします。

间 データ ソースの追加]	×
	ODBC 接続	
接続ワイサート (100)	◯ システム DSN ● ューザー DSN	○ 接続文字列を構築
ADO接続	🔿 ファーíル DSN 💻 PC	~
ODBC 接続	データソース名	
Sobe	dBASE Files	Microsoft Access dBASE Driver (*.dbf, Microsoft Excel Driver (*.xls, *.xlsx, *.xl
JDBC 接続	mariadb_connection	MariaDB ODBC 3.1 Driver
00	MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
ADO.NET	PostgreSQL35W	PostgreSQL Unicode
ADO.NET 接続		

6.2.1.10.16 Progress OpenEdge (JDBC) への接続

JDBC を使用して Progress OpenEdge 11.6 データベースサーバーへの接続するためのサンプルの手順が説明されています。

前提条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がんストールされている必要があります。 これは Oracle JDK おは Oracle OpenJDK などのオープノノースビルドであることができます。MapForce は次のロケーションから Java Virtual Machine (JVM) への やを次の順序で決定します: a) アプリケーション オプション内で設定したカスタム JVM パス Java 設定を参照してくたさい。b) Windows レジストリ内で見つか オこ JVM パス。c) JAVA_HOME 環境変数。
- MapForce (32ゼナ、64ゼナ)のプラナフォームがJRE /JDK のプラナフォームに一致することを確認してくたさい。
- オペレーティングシステムの PATH 環境変数は、JRE おは JDK インストールの bin ディレクトリークの 次を含んている必要が あます。例、C:\Program Files (x86)\Java\jre1.8.0 51\bin.
- Progress OpenEdge JDBC ドライバーが使用中のオペレーティングシステムで使用可能である必要があります。このサンプルでは、JDBC 接続は、C:\Progress\OpenEdge\java にOpenEdge SDK のインストールの一部として存在する openedge.jar とpool.jar ドライバーコンポーネトファイルにより与えられています。.
- データベース接続の詳細がおます:ホスト、データベースパスまけはエイリアス、ユーザー名、およびパマワード。

JDBCを使用してOpenEdge へ接続する

- 1. データベース接続ウィザードを開始します.
- 2. 「JDBC 接続」をクトックします。
- 3. 「ケラス マ」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミロンで区切られた.jar ファイル マのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることがで きますて: \Progress \OpenEdge \ java \ openedge . jar; C: \Progress \OpenEdge \ java \ pool . jar; 。.jar ファイル やをオペレーティングンステムの CLASSPATH 環境変数に追加した場合、「ケラス や」テキストボックスを空白のままこ することができます。(次も参照: CLASSPATH の構成)。
- 4. 「ドライバー」ボックスから、 com.ddtek.jdbc.openedge.OpenEdgeDriver を選択します。 このエントリは、有効な .jar ファイル なが、「クラス な」テキストボックス内、おけよ オペレーティングシステムの CLASSPATH 環境変数内で検出された場 合のみ使用することができます(前のステップを参照してください)。

JDBC 接続	
接続文字列を入力して ます。	、有効な JDBCドライバーを選択 (または手動で入力) してください。その後接続ボタンをクリックし
クラスパス:	C:\Progress\OpenEdge\java\openedge.jar;C:\Progress\OpenEdge\java\pool.jar;
ドライバー:	com.ddtek.jdbc.openedge.OpenEdgeDriver
ユーザー名:	dbuser
パスワード:	•••••
ቻ −タベ −ス URL:	jdbc:datadirect:openedge:// <host>:<port>;databaseName=<db_name></db_name></port></host>

- 5. データベースへのユーザー名とマワードを対応するテキストボックスに入力します。
- 6. ハイライトされた値を、使用するデータベースサーバーゴ適用する値と置き換えて、データベースURL テキストボックス内にデータベ ースサーバーへの接続文字列を入力します。

jdbc:datadirect:openedge://host:port;databaseName=db name

7. 「接続」をクリックします。

6.2.1.10.17 Progress OpenEdge (ODBC) への接続

このサンプルではProgress OpenEdge 11.6 ODBC ドライバーを使用して、Progress OpenEdge データベースサーバーへの接続の簡単な手順について説明されています。

前提条件:

Progress OpenEdge ドライバーのオメの ODBC コネクタが使用中のオペレーティングシステムニインストールされている必要があります。Progress OpenEdge ODBC ドライバーは、ベンダーのWeb サイトからダウムロードすることができます(次も参照: データベースドライバーの概要)。MapForce の32 ビナバージョンを作動する場合は、32 ビナドライバーをダウムロードし、64 ビナバージョンを作動する場合は、64 ビナドライバーをダウンロードしてくたさい。インストールの後、ODBC ドライバーが使用中のマシンで使用することができるかを確認してくたさい(次も参照: 使用することのできる ODBC ドライバーを確認する)。

ODBC Data Source Administrator		×		
User DSN System DSN File DSN Drivers Tra	acing Connection F	ooling About		
ODBC Drivers that are installed on your system:				
Name	Version	Company		
Microsoft Access Driver (*.mdb, *.accdb) Microsoft Access Text Driver (*.txt, *.csv) Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb) Progress OpenEdge 11.6 Driver SQL Server SQL Server SQL Server Native Client 11.0	15.00.4695.1000 15.00.4695.1000 15.00.4695.1000 7.10.05.167 6.01.7601.17514 2011.110.3000.00	Microsoft Corpc Microsoft Corpc Microsoft Corpc Progress Softw; Microsoft Corpc Microsoft Corpc		
<		Þ		
An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.				
OK Canc	el <u>A</u> pply	Help		

データベース接続の詳細が別ます、ホスト名、ポート番号、データベース名、ユーザーID、およびパスワード。 ٠

ODBC を使用して Progress OpenEdge に接続する

- <u>データベース接続ウィザードを開始します</u>.
 「ODBC 接続」をクリックします。
 「ユーザーDSN」をクリックします。(おけよ 次の手順が類似している「システムDSN」、おけよ「ファイルDSN」をクリック します)。
- 4. 「追加」 **をクルクします。
- 5. 「Progress OpenEdge ドライバー」をパトか選択し、「ユーザーDSN」(おけよ 適用できる場合は、「システム DSN」)をクリックします。

Create an ODBC DSN	x
Select a Driver and click on either User or System to determine what kind of DSI you want to create.	N
Progress OpenEdge 11.6 Driver	•
User DSN System DSN Cancel	

6. データベース接続の資格情報(データベース、サーバー、ポート、ユーザー名、パマワード)を入力し、「OK」をクルクします。入 カされたデータを保存する前に、接続を検証するコよ「接続をテストする」をクルクします。

ODBC Progress OpenEdge W	ire Protocol Driver Setup	? <mark>- x -</mark>
General Advanced Securit	y Failover About	
Data Source Name:	my_progress_dsn	<u>H</u> elp
D <u>e</u> scription:		
Host <u>N</u> ame:	localhost	
Port Number:	8910	
Data <u>b</u> ase Name:	oebpsdev	
<u>U</u> ser ID:	altova	
Test Connect	OK Cancel	Apply

7. 「OK」をクリックします。新規データノースがODBC データノースのリストに表示されます。

ODBC Connection	าร
 System DSN User DSN 	Build a connection string
○ File DSN	· · · · · · · · · · · · · · · · · · ·
Data Source Name	Driver
t, * ⊘ 5	
	<u>C</u> onnect Close

8. 「接続」をクリックします。

6.2.1.10.18 Sybase (JDBC) への接続

このサンプルでは、JDBCを使用して Sybase データベースサーバーは接続する手順にして説明されています。

前提条件:

- JRE (Java Runtime Environment) おは Java Development Kit (JDK) がんストールされている必要があります。 これは Oracle JDK おは Oracle OpenJDK などのオープノノースビルドであることができます。MapForce は次のロケー ションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプルケーション オプション内で設定したカスタ ム JVM パス <u>Java 設定</u>を参照してくたさい。b) Windows レジストリ内で見つかった JVM パス。c) JAVA_HOME 環境変数。
- MapForce (32-ビナ、64-ビナ)のプラナフォームがJRE /JDK のプラナフォームに一致することを確認してくたさい。
- Sybase jConnect コンポーネトカオペレーティングシステムニインストールされている必要かあります(このサンプルでは jConnect 7.0 が使用されており、Sybase Adaptive Server Enterprise PC Client インストールの一部としてインストールされています)。データベースクライアントのインストール命令に関しては、Sybase ドキュメントを参照してくたさい。
- 次のデータベース接続詳細が存在します。ホスト、データベース、ポート、ユーザー名、および、パスワード。

JDBC を使用して Sybase へ接続する

1. <u>データベース接続ウィザードを開始します。</u>

- 2. 「JDBC 接続」をクルクします。
- 3. 「ケラス・ス」の横にデータベースへの接続を与える.jar ファイルへの やを入力します。必要であれば、セミコロンで区切られた.jar ファイル やのノストを入力することもできます。このサンプルでは、必要とされる.jar ファイルは次の やを使用して見つけることがで きます: C:\sybase\jConnect-7_0\classes\jconn4.jar。.jar ファイル やをオペレーティングシステムの CLASSPATH 環境変数に追加した場合、「ケラス・ペ」テキストボックスを空白のままにすることができます。(次も参照: CLASSPATH の構 成)。
- 4. 「ドライバー」ボックスから、 を選択します。「ドライバー」ボックスから、 com.sybase.jdbc4.jdbc.SybDriver を選択しま す。 このエストリは、 有効な.jar ファイリレ なが、「クラスレ な」テキストボックス内、 おけよ オペレーティングシステムの CLASSPATH 環境変数内で検出された場合のみ使用することができます(前のステップを参照してくたさい)。

JDBC 接続	
接続文字列を入力して ます。	有効な JDBCドライバーを選択 (または手動で入力) してください。その後接続ボタンをクリックし
クラスパス:	com.sybase.jdbc4.jdbc.SybDriver
ドライバー:	•
ユーザー名:	
パスワード:	
データベース URL:	jdbc: <vendor>://<server>/<db>?<option>=<value></value></option></db></server></vendor>
	v
	接続(C) 閉じる(L)

- 5. データベースのユーザー名とマワードを対応するテキストボックスに入力します。
- 6. ハイライトされた値をデータベースサーバーに適用することのできる値と置き換えて。データベースURL テキストボックス内のデータベ ースサーバーの接続文字列を入力します。

jdbc:sybase:Tds:hostName:port/databaseName

7. 「接続」をクルクします。

6.2.1.10.19 Teradata (JDBC) への接続

このサンプルでは、JDBCを使用したTeradataデータベースへの接続方法について説明されています。

必要条件:

 JRE (Java Runtime Environment) おは Java Development Kit (JDK) がんストールされている必要があります。 これは Oracle JDK おは Oracle OpenJDK などのオープノノースビルドであることができます。MapForce は次のロケーションから Java Virtual Machine (JVM) への なを次の順序で決定します: a) アプリケーション オプション内で設定したカスタム JVM パス Java 設定を参照してくたさい。b) Windows レジストリ内で見つか オニ JVM パス。c) JAVA_HOME 環境変数。

- MapForce (32ビナ、64ビナ)のプラナフォームがJRE/JDK のプラナフォームに一致することを確認してくたさい。
- JDBC ドライバー(データベースへの接続を提供する1つまけは複数の.jar ファイル) がオペレーティングシステム上で使用できる必要が影けます。このサンプルでは、Teradata JDBC ドライバー16.20.00.02 が使用されます。詳細に関しては、次を参照してくたさい、http://downloads.teradata.com/download/connectivity/jdbc-driver。
- 次のデータベース接続詳細が存在します:ホスト、データベース、ポート、ユーザー名、および、パスワード。

JDBC を使用した Teradata への接続の方法:

- 1. データベース接続ウィザードを開始します。
- 2. 「JDBC 接続」をクトックします。
- 3. 「クラス・ス」の横に、データベースへの接続を与える.jar ファイルへの やを入力します。必要な場合、セミコレンで区切られた.jar ファイル やのノストを入力することができます。このサンプルでは、.jar ファイルは、次の やの場所に存在します: C: \jdbc\teradata\。オペーティングシステムのCLASSPATH 環境変数 に.jar ファイル やが追加されている場合、「クラスパ ス」テキストボックス空のままにすることができます(次も参照してくたさい、CLASSPATH の構成).
- 4. 「ドライバー」ボックス内で、com.teradata.jdbc.TeraDriver を選択します。有効な.jar ファイル なが クラス な」テキスト ボックスで、おさよ、オペレーティングシステムのCLASSPATH 環境変数内に存在すると、使用することができます(前のステップを 参照してください)。

JDBC Connections		
Enter a connection st 'Connect' to proceed.	ring and select (or enter manually) a valid JDBC driver. Click on	
Classpaths:	C:\jdbc\teradata\terajdbc4.jar;C:\jdbc\teradata\tdgssconfig.jar	
Driver:	com.teradata.jdbc.TeraDriver 🗸	
Username:	demouser	
Password:	•••••	
Database URL:	jdbc:teradata://demodatabase	
	×	
	<u>C</u> onnect <u>Close</u>	

5. ユーザー名と、マワードを対応するテキストボックスのデータベース内に入力します。

6. ハイライトされた値をデータベースサーバーと適用することのできる値を置換え、データベースURL テキスト内にデータベースサーバー に接続文字列を入力します。

jdbc:teradata://databaseServerName

7. 「接続」をクリックします。

6.2.1.10.20 Teradata (ODBC) への接続

このサンプルでは、ODBCを使用したSQL Teradata データベースへの接続方法について説明されています。

前提条件:

- Teradata ODBC ドライバーがインストールされる必要があります(参照: <u>http://downloads.teradata.com/download/connectivity/odbc-driver/windows</u>。このサンプルは、Windows バージョン 16.20.00 のためにODBC ドライバーを使用しています。
- 次のデータベース接続詳細が存在します。ホスト、ユーザー名、および、パスワード。

ODBC を使用した Teradata への接続の方法:

1. Windows キーを押して、「ODBC」を入力し、「ODBC データソースのセットアップ(32-ビット)」を提案されるリストから選択します。64-ビット ODBC ドライバー が存在する場合、「ODBC データソースのセットアップ(64-ビット)」を選択し、次の ステップ内で64-ビット MapForce を使用します。

ODBC Data Source Administrator (32-bit)	×
User DSN System DSN File DSN Drivers Tracing Connection Pooling About	1
System Data Sources:	
Name Platform Driver	A <u>d</u> d
	<u>R</u> emove
	<u>C</u> onfigure
An ODBC System data source stores information about how to connect to a A System data source is visible to all users on this machine, including NT so	the indicated data provider. ervices.
OK Cancel	Apply Help
ON Cancer	Арріу пеір

2. 「System DSN」 ダブをクリックして、「追加」をクリックします。

Create New Data Source		×
	<u>Select a driver for which you want to set up a data</u> Name Microsoft Paradox Driver (*.db) Microsoft Paradox-Treiber (*.db) Microsoft Text Driver (*.txt; *.csv) Microsoft Text-Treiber (*.txt; *.csv) SQL Server SQL Server SQL Server Native Client 11.0 Teradata Database ODBC Driver 16.10 <	ata source.
	< <u>B</u> ack Finish	Cancel

3. 「Teradata データベース ODBC ドライバー」を選択して、「終了」をクリックします。

ODBC Driver S	etup for Teradata Database	\times
Data Source		OK
<u>N</u> ame:	my_teradata_source	OK Cancel
Description:		Help
	ver Info	
Name or IP address	demoserver	\sim
Authentication	n	
Use <u>I</u> ntegr	ated Security	
Mechanism:	~	
Pa <u>r</u> ameter:		<u>C</u> hange
<u>U</u> sername:	demouser	
<u>Password</u>	•••••	
🔿 Teradata <u>V</u>	<u>V</u> allet String	
Optional		
D <u>e</u> fault Datab	base:	
Account String	g:	Options >>
Session Charact	ter Set:	
UTF8		~
L		

- 4. 名前を入力し、任意で、このODBC データノースを将来識別できる詳細を与えます。また、データベース接続資格情報(データ ベースサーバー、ユーザー、パスワード)を入力し、オプションでデータベースを選択します。 5. 「OK」をクリックします。データノースがリスト内に表示されます。

🚳 ODBC Data	a Source A	dministra	ator (32-bit	:)					>	×
User DSN Sy	stem DSN	File DSN	N Drivers	Tracing	Connection	Pooling	About			
<u>S</u> ystem Data	Sources:									
Name		Platform	Driver					A <u>d</u> d.		
my_teradata	_source	32-bit	Teradata D)atabase C	DBC Driver 1	16.10		<u>R</u> emo	ve	
								<u>C</u> onfigu	re	
<							>			
Ar A	n ODBC Sy System dat	stem data a source i	source stor s visible to a	es informat Ill users of	ion about hov this computer	w to conn r, includin	nect to th g NT se	ne indicated data ervices.	a provider.	
					OK	Cano	el	Apply	Help	

- 6. MapForce を作動し、<u>データベース接続ウィザードを開始します</u>。
 7. 「ODBC 接続」をクリックします。

	IS
System DSN	O Build a connection string
O User DSN	
◯ File DSN	
Data Source Name	Driver
my_teradata_source	Teradata Database ODBC Driver 16.10
t, * ⊘ *	

- 8. 「System DSN」をクリックし、前に作成されているデータノースを選択し、「接続」をクリックします。
- メモ 次のエラーが発生した場合:「ドライバーが無効として返されました(おけば返しませんでした) SQL_DRIVER_ODBC_VER: 03.80」「ドライバーが無効と返されました」。ODBC クライアナトへの なが使用中のシステムのPATH 環境変数内に存在する ことを確認してくたさい (例えば このロケーションニインストールされている場合、C:\Program Files\Teradata\Client\16.10\bin)。この なが不足している場合、手動で追加します。

6.2.2 データベースマッピングの概要

このセクションはMapForce内出データベースと作業する方法に関する概要です。接続の確立に成功した後、どのようしてデータベースと 作業するかについて説明されています(次を参照:<u>データベースへの接続</u>)。これには、マンピングに表示されるデータベースオブシェクト、デー タベースルーションシップの処理、マンピングプロセスに適用することのできるデータベースの構成などが含まれます。データベースエンポーネント から、おけはデータベースエンポーネントにデータをマンピングする際の特定の目的を達成する方法もこのセクションの例に含まれています。

6.2.2.1 マッピングにデータベースを追加する

マピングにデータベースを追加する前に、データベースマピングがサポートされている変換言語が選択されていることを確認してくたさい。これ は、BUILT-IN 変換言語、おけば次の言語であることができます: C++、C#、Java (次も参照してくたさい) 変換言語の選択)。マピ ングを FlowForce Server にデプロイ、おけは MapForce Server で実行、おけは バレク転送とストプドプロシージャなどの機能を使用 する場合、BUILT-IN が変換言語として選択される必要があります。

希望する変換言語が選択されると、データベースをマピングに次の方法で追加することができます。

- 「挿入」メニューカジデータベース」をクリックします。
- 「データベースの挿入」(1) ツール・・ポタンをクトックします。

これらのアクションを取ると、データベースへの接続に必要なステップを導く、データベース接続ウィザードが表示されます。

🕑 データベースを選択	R
送 接続ウィザード	接続ウィザード
	ソース データベースを選択し、 [次へ]をクリックしてください。
() () () () () () () () () () () () () (使用したいデータベースのベンダーが下のリストにない場合、ADOまたは ODBCを使用してコネクションを作成してください。
	Microsoft Access (ADO)(A) Progress (ODBC)
	Microsoft SQL Server (ADO)(S)
ADO接続	Oracle (ODBC/JDBC)(0)
	MySQL (ODBC)(M)
	◯ IBM DB2 (ODBC/JDBC)(D)
ODBC 接続	◯ IBM Informix (ODBC / JDBC)(I)
	⊙ Sybase (ODBC)(Y)
	PostgreSQL
JDBC 接続	Firebird (ODBC / JDBC)(F)
	─ SQLite(L)
COUNT ADO.NET 接続	
グローバル リソー ス	ウィザードの構成をスキップ
000 PostgreSQL 接続	次へ(N) > 閉じる(L) //.

データベース接続ウィザード

メモ 高度なシナリオの一部では、データベースに変数として追加することができます(次を参照: 変数の使用)。データベース構成を変数として追加する場合は、上記と同じデータベース接続ウイザードが表示されます。

このウィザードを使用してMapForce によりサポートされているデータベースへの接続の方法に関しては、以下を参照してくたさい、データベー スへ接続。

<u>Sybase への接続(JDBC)</u>

データベースへの接続に成功すると、マピングに表示されるデータベースオブジェクトを選択するようにプロンプトされます。詳細に関しては、以下を参照してくたさい、データベースオブジェクトの追加、編集、および削除。

6.2.2.2 例: "altova.mdb" データベースをマッピングに追加する

この例は、Microsoft Access データベースをマピングに追加する方法を説明します。サンプルのデータベースは altova.mdb と呼ばれ、 <マイドキュメント > ¥Altova Wap Force2021 Wap ForceExamples ¥ フォルダー内にあります。 altova.mdb データベースは、 このドキュメント 内で説明されているデータベースは関連したアグションと概念をサポートします。

altova.mdb データベースをマッピングに追加する

1. 「挿入」メニューカジデータベース」をクリックします。おけよ 「データベースの挿入」(🏮) ツールボタンをクリックします。

🕑 データベースを選抜	R
後に 接続ウィザード	接続ウィザード
	ソースデータベースを選択し、「次へ」をクリックしてください。
	使用したいデータベースのベンダーが下のリストにない場合、ADOまたは ODBCを使用してコネクションを作成してください。
既存の接続	Microsoft Access (AD0)(A) Progress (0DBC) Microsoft SQL Server (AD0)(S)
ADO接続	 Oracle (ODBC/JDBC)(O) MySQL (ODBC)(M)
ODBC 接続	 IBM DB2 (ODBC/JDBC)(D) IBM Informix (ODBC / JDBC)(I) Sybase (ODBC)(Y)
) JDBC 接続	 PostgreSQL Firebird (ODBC / JDBC)(F) SQLite(L)
COUNT ADO.NET 接続	
グローバル リソー ス	ウィザードの構成をスキップ
PostgreSQL 接続	次へ(N) > 閉じる(L) //.

2. Microsoft Access (ADO)をクルクして、「次へ」をクルクします。

🕑 データベースを選択	5
接続ウィザード	MS Access <> 接続
既存の接続	「参照」をクリックして、Microsoft Access データベースを選択してください。その後、「次へ」をクリックしてデータ ベースに接続してください。
へ ADO接続	Documents¥Altova¥MapForce2016¥MapForceExamples¥Tutorial¥altova.mdb 参照
ODBC 接続	
) JDBC 接続	
ジ ローバル リソー	
ス	<前へ(B) 接続(C) 閉じる(L)

- 3. <マイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内のaltova.mdb ファイルを参照 し、「接続」をクトックします。
- 4. データベースオブシェクトを選択するようにプロレプトされると、「ユーザーテーブル」を選択します。

。 「「ロデータベース オブジェクトを挿入	×
テーブル、ビュー、ストアドプロシージャを選択するか、独自の SOL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SOL SELECT ステートメン トはローカルの mfd ファイルに保存されます。	
ג-ע	
▼ ▼ ◆ SELECT ステートメントの追加/編集… リレーションの追加/編集 (0)… レコードセット構造の追加/編集	
HEW ■ altova HEW ■ altova HEW ■ テーブル HEW ■ Address HEW ■ Add	
プレビューを表示(P) >> OK	キャンセル
プレビューを表示(P) >> OK	キャンセル

6.2.2.3 データベースオブジェクトを追加、編集、削除する

データベースの一部は、(スキーマ、テーブル、ビューなどの多数のオブジェクトをもつことができます。このトピックは、マッピングの目的に必要な データベースオブジェクトのみをマッピングする方法について説明します。下では、サンプルAccess データベースが使用されています。データベ ース型のナッシの手順と類似しています。

- 1. 「挿入」メニューから「データベース」をクルクします。
- 2. 「接続ウィザード」をクリックして、Microsoft Access (ADO) をクリックします。
- 3. 「次へ」をクリックし、<マイドキュメント> Altova Map Force 2021 Map Force Examples ¥ フォルダー内の altova.mdb を参照します。

マピングに含まれるデータベースオブジェクトを選択することのできるアケションダイアログボックスが表示されます。

。 「自データベース オブジェクトを挿入	×
テーブル、ビュー、ストアドプロシージャを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SQL SELECT ステートメント はローカルの mtd ファイルに保存されます。	
y-ス ▼ ▼ ◆ SELECT ステートメントの追加/編集 リレーションの追加/編集 (0) レコードセット構造の追加/編集 □ □ altova □ □ □ = サーブル □ □ □ Department □ □ □ Department □ □ Person □ □ ● システム テーブル	
プレビューを表示(P) >> OK	キャンセル

構造内のトップノード 🔢 はデータベース接続を表します。その後の構造はデータベースの種類にお異なります。例えば、 Oracle とIBM DB2 データベースコは接続ノードの下にスキーマノード 🛣 が存在しますが他のデータベースの種類にはカタログ(データベース) 🏮 ノードが 存在します。太線はデフォリトのカタログ(データベース)、ませまスキーマを示しています。

データベースユーザーアカウトがサーバー上に複数のデータベースおけはスキーマを持つ場合、 S アイコンをクリックして必要なものに切り替えることができます。 下のサンプルは SQL Server の例を表示しています。 構造は地のデータベースの型では異なる場合があります。



データベースオブジェクトを含むには、(例えば、テーブル)マッピングの横のチェックボックスをクトックしてくたさい。この例の目的のために、「ユーザーテーブル」の横のチェックボックスをクトックします。

「オブジェクトロケーター」ボタン(しま))を使用して特有のデータベースアイテムを検索することができます。ダイアログボックスの下の部分に 表示される、コンポボックス内から特定のオブジェクトを選択(おけよ 名前を入力)します。 「フィルター」ボタン(1211) によりオブジェクトを名前別にフィルターすることができます。1度フィルターボタンをクトックすると、フィルターアイコン がフィルタレングをサポートするオブジェクトの横に表示されます(この例では、 "Tables")。 フィルターアイコンをクトックして、オブジェクト名が検 索テキストと終わる、同等、まちまは、含むかを選択してくたさい。フィルターの横に検索テキストを入力することができます(この場合、 "A"):

📮 🔳 🔃 altova
🗄 🖵 🔳 📁 altova
🕞 🔳 🛅 Tables 🛛 🔽 🗛
🕀 🗹 💼 User Tables
Address
🛛 🗹 🔟 Altova
🛄 🗋 🛅 System Tables
📄 🛅 SELECT Statements

「チェックされたオブジェクトのみ表示」ボタン(🔽)は、チェックボックスがアクティブな箇所のアイテムを表示します。

「SELECT ステートメントの追加/編集」ボジンを使用して、現在のデータベースのかがにカスタムSELECT ステートメントを追加、おこ は編集することができます。このようなステートメントにより返されたデータは、この後マンピングノースとして使用することができます。詳細に関し ては、次を参照してくたさい、仮想テーブルのSQL SELECT ステートメント。

「リレーションの追加/編集」ボタンを使用して、既存のものに加え、データベース内のフィールド間のローカルな主キーと外部キー・リレーショ ンシップを定義することができます。詳細に関しては、次を参照してください、ローカルリルーションシップの定義。

「レコードセル構造の追加/編集」ボタノはストアドプロシージャをサポートするデータベースに適用されます。ストアドプロシージャがデータベースツレーから現在選択されている場合のみ有効化することができます。詳細に関しては、次を参照してくださ、シストアドプロシージャ。

「プレビューの表示」ボタイこより、現在選択されているテーブルから、おイコビューから素早くデータをプレビューすることができます。 データベー スプラウザーを使用して、マメピング処理とは別にデータベースを参照し、 クロノすることもできます。 詳細に関しては、 次を参照してくたさい、<u>デー</u> タベースの参照とクロリ。

「テーブル名からスキームを削除する」チェックボックスがでオタベースのオータに有効化されます。この機能は後に異なるデータベースに切り替える際に役に立ちます。詳細に関してはデータベースとスキーマの切り替えを参照してください。

マピングへデータベースオブジェクトを追加する準備が整うと、「OK」をクリックします。選択されたテーブル、ビュー、のみがデータベースコンポ ーネント上に表示され、マピング接続をMapForceの標準の方法で描くことができます。

🖯 altova	
⊡Address	
attova ∰ 🗄 🗄	
≥⊞ III Department	
}⊞ ⊞Office	
}⊞ III Person	
1	

データベースオブジェクトを変更するコは、コンポーネントを右クトックして、「データベースオブジェクトの追加/削除/編集」を選択します。
🖯 altova	_			
		追力	ロ/削除テーフ	ブル①
DE Altova		更新	Ͳ(<u>Ε</u>)	
PrimaryKey	M	検索	≂ ≅(<u>F</u>)	Ctrl+F
¦a ⊷⊞ ⊡Office Foreig	Ж	切り	取り(<u>T</u>)	Ctrl+X
p⊞ III Department b⊞ III Office		⊐Ľ	-(<u>C</u>)	Ctrl+C
≥⊞ III Person	e	貼り	付け(型)	Ctrl+V
	\mathbf{x}	削除	亲(<u>D</u>)	Delete
	~	9 <u>9</u> ,	ーを左揃えに	.する(L)
		·995	ーを右揃えに	.する(<u>R</u>)
		ナロ	パティ(<u>0</u>)	

6.2.2.4 データベースとスキーマの切り替え

新規のデータベース接続を使用するようにマンピング上のデータベースエンポーネントを変更するコよ、データベースエンポーネント設定を開き、「変更」をクリックし、新規のデータベースへ接続するオンタニンケザーとのステップに従います。データベースオブジェントのみを変更する場合、デ ータベースエンポーネントのタイトルレーを右クリックして、「データベースオブジェクトの追加/編集/削除」をエンテキストメニューから選択します。データベースオブジェントの追加、編集、および削除も参照してください。

デフォルトではデータベースの変更はすべての既存のマッピング接続をデータベース接続に、ませま、データベース接続から再マップしないことに 注意してくたさい。このトピック内で説明される準備ステップに従うとデータベースの型の多数はデータベースの変更後にマッピング接続を保持 することができます。

<u>データベースオブシェクトの追加、編集、および削除</u>内で説明されている通りデータベースオブシェクトを選択すると、任意でデータベースオブ ジェクト名がデフォルトのスキーマイニマれして相対的に考慮されるかを選択することができます。データベースのエレテキストでは、「スキーマ」は通 常パトミッションの割り当てなどセキュリティの目的のナックのデータベースオブシェクトのエンカルなグループ分類です。スキーマイはデータベースの型 により若干異なる振る舞いを取ります。

MapForce 内では、マピングを異なるデータベースコ後に変更する予定がある場合データベースオブジェクト名をスキーマーン対して相対的に 扱うことは需要です(例えば、生産データベースニ変更する場合など)。これはサーイ、上でデータベーススキーマース名前が付けられており、マッ ピングを必要に応じて更新する場合に役に立ちます。スキーマがマピングデザイン時に使用されたものと同じ構造を持つ場合、マピング接 続を手動で変更することなく切り替えることができます。上記のとおりオブジェクト名がデフォルトのスキーマーマンガして相対的になっている場合の みこれは可能です。

以下の点に注意してくたさい

- デフォルトのスキーマーマはして相対的なオブジェクト名の使用はスキーマをサポートするデータベース型のために使用することができます: IBM DB2、IBM Informix, IBM Db2 for i (iSeries)、Oracle、PostgreSQL、Progress OpenEdge、SQL Server とSybase。
- この機能はMapForce バージョン 2020 からサポートされています。
- <u>ローカルルーションシップ</u> おまな<u>仮想テーブルとてのSELECT ステートメト</u>. がデータベースエンポーネトに含まれる場合相対的な名前を使用することはできません。

データベースオブジェクト名をデフォルトのスキーマイマ対して相対的に扱う方法

- 1. マッピング上のデータベースコンポーネントのタイトルドーを右クトックし、「データベースオブジェクトの追加/削除/編集」をコンテキストメニューから選択します。
- デフォルトのスキーマは属する1つまけは複数のオブジェクト、まけは、デフォルトのカタログ(データベース)とスキーマを選択します。 デフォルトのデータベースとスキーマは太字で表示されています。下のサンプルでは、デフォルトのカタログは「Sandbox」で、デフォ ルトのスキーマは「user」です。この構造はSQL Server データベースは固有で、他のデータベース型とは異なる可能性があり ます。表示される構造に関しての詳細は、データベースオブジェクトの追加、編集、および削除を参照してくたさい。

☞ データベースオブジェクトの追加/削除/編集	×
テーブル、ビュー、ストアドブロシージャを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SQL SELECT ステートメントはローカルの mfd ファイルに保存されます。	
ソース ▼	
Connection1 Sandbox Connection1 Sandbox Connection1 Sandbox Connection1 Sandbox Connection1 Sandbox Connection1 Connec	
フレビューを表示(P)>> 「デフォルトのスキーマに相対するオブジェクト名を使用してください OK	キャンセル

3. 「デフォルトのスキーマーズ対して相対的なオブジェクト名を使用する」チェックボックスを選択します。データベースが相対的なオブジェクト名をサポートしない場合チェックボックスは灰色で表示されることに注意してくたさい。

マピング上で必要なオブジェクトがデフォルトではなく異なるスキーマ内に存在する場合以下の代替が存在します

- 必要とされるデフォルトのスキーマへのアクセスを持つ他のデータベースユーザーとして接続します。
- 必要とされる特権を有することが前提で既存のデータベースユーザーのデフォルトのスキーマを変更するためにデータベースサーバーを 再構成します。

SQL Serverを例とて、データベースの管理者は現在のカタログが「サイボックス」でユーザーとスキーマか既存であることを想定して)以下 に示されるステートメントに類似したステートメントを使用してデータベースユーザーのデフォルトのスキーマを変更することができます。

```
USE [Sandbox]
GO
ALTER USER [test_user] WITH DEFAULT_SCHEMA=[test_schema]
GO
```

デフォルトのスキーマーン相対的なデータベースオブジェクトはマッピング上で相対的な名前と共に表示されています。次の2つのデータベースコンポーネント間の差分を考慮してください

Connection		Connection1	
⊳⊡ III table	₽	þ ⊡ test.table þ	۶
¢ [1 ¶id	Þ	¢ ⊡¶id ¢	۶
> 🔲 val	Þ	¢—∎val ¢	۶
🔉 🛄 user_data	Þ	🖕 📖 user_data 🛛 🛱	۶

左側のエレポーネントでは、「デフォルトのスキーマニ相対的なオブジェクト名を使用する」、チェックボックスが選択されており、このためにテ ーブルは相対的な名前と共に表示されています。右側のエレポーネントでは、チェックボックスは選択されておらず、テーブルコはスキーマ名が含まれています。

データベースオブジェクトが相対的な方法で参照されている場合、マッピング接続を失うことなしにマッピングを新規のデータベースまけはスキ ーマイニ以下のように変更することができます:

- データベースコンポーネント設定を開き、「変更」をクリックします。ウィザーとステップゴ従し、新規のデータベースに新しいユーザーとして接続します。オブジェクトを相対的な名前で修飾した場合、新規のデータベースは同じ構造を持ちます。そして、マッピング上のすべての接続は自動的に更新されます。具体的には、新規のデータベースユーザーのデフォノムトカタログとスキーマにマッチします。
- 切り替えを再帰的なベースで行う必要かある場合、データベース接続をグロー、シリノースとして定義することが奨励されます。例えば、グロー、シリノースは2つの構成を持つことができます。開発データベースのすめのデフォルトの構成と生産のすめの構成を持つことができます。両方のデータベースが同じ構成を持つことを前提とし、ドロップダウノリストから必要とされる構成を変更してデータベース間を切り替えることができます。

```
Default 🔹 🛃 🖕
```

このシナオの詳細に関しては物データベースの切り替えを参照してくたさい。

```
データベースオブジェクトが引い替え後赤い色で表示されます。これは新規のデータベーススキーマではこれらのデータベースオブジェクトが存在しないことを指します。
```

コード生成

「デフォルトのスキーマース対して相対的にオブジェクト名を使用する」 チェックボックスは生成された-C#、C++、おけま Java プログラムコードに影響を与えます。このチェックボックスが選択されていると、全てのデータベースレファレンスが生成されたコード内で相対的にないます。次のサンプルはチェックボックスが選択されていない 生成された(C#) SQL ステート メントのサンプルです。 テーブル名にスキーマ名が含まれていることに注意してくたさい。

var3_NewStatement = (Altova.Db.Dbs.NewStatement(closure.var1_Connection_Instance, 1, "SELECT [id], [val], [user_data] FROM [user].[table]"));

次はチェックボックスが選択された生成された SQL ステートメント (C#)のサンプルです:

```
var3_NewStatement = (Altova.Db.Dbs.NewStatement(closure.var1_Connection_Instance, 1,
"SELECT [id], [val], [user_data] FROM [table]"));
```

メモ スキーマ名を削除するすみはレガシーな方法が存在します。「テーブル名からスキーマ名を削除する」 チェックボックスを <u>データベ</u> <u>ースエンポーネト 設定</u>をから選択します。 レガシーなアプローチは下位互換性のみのすみはこサポートされますが、 回避されるへきで す。

6.2.2.5 データベースリレーションシップの処理

リレーショナルデータベースは、その名前か示すように、テーブル間にフレーションシップを持ちます。フォルダーくマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\にあるaltova.mdbデータベースを例にしてみると、複数のルー ションシップが存在することに気がっきます:

- サンプルの会社(テーブル「Altova」に対応する1つまけは複数のオフィスから構成されています(例えば、1つは、東京に、もう1つは名古屋に存在するとます)。データベース用語では、「Altova」と「Office」テーブル間に1つ対複数のすなわち、「Altova」テーブル内の各「PrimaryKey」レコードのために、「Office」テーブル内で複数の「ForeignKey」レコードが存在することができます。「ForeignKey」値が、「Altova」内の「PrimaryKey」値と対応する「Office」の箇所は、「Altova」のオフィスとして考えされます。
- 各オフィスは1つまけは複数の部署(例えば、「Marketing」、「IT」、「Development」)から構成されています。「Office」と「Department」テーブル間につと複数のルーションシップが存在します。
- 最後に、各部署は、複数の社員から構成されています。ですから、1つ対複数のリレーションシップは「Department」と「Person」テーブル間のルーションシップです。



altova.mdb データベース内のテーブルリレーションシップ (Microsoft Access "リレーションシップ"ビュー)

データベーステーブル間のルーションシップは、マピングの目的にとって重要です。MapForce は、データベースをマピングに追加する際 に、このようなデータベースルーションシップの個々の変更を記録します。これにより、データをデータベースにマピングする際、まけはデータベー スにデータをマピングする際、データベースルーションシップを保持することができます。この概念をより良く理解するけっかに(挿入|データベ ースメニューコマイドを使用して) altova.mdb データベースをマンピングに追加します。「ルート」テーブルの下のテーブルを呼び出します:



「ルート」テーブル

「ルート」テーブルを展開すると、ツレービュート内に存在する、すべての関連するテーブルが表示されます。例えば、Office テーブルを展開すると、関連したテーブルの階層が表示されます:

- テーブルの前の左矢印(←)は、このテーブルが子テーブルであることを表示しています。例えば、Addressは、Officeの子です。DepartmentもOfficeの子であり、おこ、Addressの兄弟テーブルであり、同じインデトのレベルに存在します。表示されているように、マンピング上のリレーションシップは、上の「リレーションシップ」ダイアグラムに対応しています。
- テーブルの前の右矢印(→)はこのテーブルが親テーブルであることを表示しています。例えば、Altovaは Officeの親です。

🏮 altova	
j⊞ III Address	;
p⊞ III Altova	;
¦a⊞ III Department [a	¢
¦¢⊟ III Office	,
🗘 🔤 PrimaryKey 🗘	,
🖒 🔤 🖫 ForeignKey 🖒	
ပုံ Desc ပုံ	•
ပုံ EMail ပုံ	•
C Established	•
β Π Fax β	•
ppiname ppina	•
ppinne p	•
Address ForeignKey	•
C IV PrimaryKey	
ForeignKey = Office.PrimaryKey	
C City C	
▷ □ state ▷	
S Street S	
← ·	:
Grade Strategy → III Altova ForeignKey	:
⊇⊞ III Person	;

MapForce 内のテーブルリレーションシップ (altova.mdb データベース)

テーブルの階層の表示は、マピングがデータベースから読み取る、おさボータベースへ書き込む際に既存のデータベースルーションシップを 保存する手助けをします。例えば、Person テーブルから全てのレコードを取得し、XML ファイルに各個人を部署にブループ別に入力する とします。具体的にさよ、XML ファイルは、この別で使用されている altova.mdb データベースのようこ、各個人を部署にレンします:

					Departi	ment						—	×
Z		Pri	imaryKey 👻 Fo	reignKey 👻	Name 🚽 👻								
	P		1	1 Ad	ministratior								
	L	Z	PrimaryKey 👻	EMail 👻	First	*	Last	Ŧ	PhoneExt	÷	Title	-	
			1	v.callaby@nan	Vernon	Call	aby			582	Office Mana	age	
			2	f.further@nan	Frank	Furt	her			471	Accounts Re	ece	
			3	l.matise@nand	Loby	Mat	ise			963	Accounting	Ma	
		*											
	P		2	1 Ma	rketing								
	L	4	PrimaryKey 👻	EMail 👻	First	-	Last	*	PhoneExt	-	Title	-	
			4	j.firstbread@n	Joe	First	bread			621	Marketing N	Vlar	
			5	s.sanna@nano	Susi	Sani	na			753	Art Director	r	
		*											
	무		3	1 Eng	gineering								
		4	PrimaryKey 👻	EMail 👻	First	•	Last	Ŧ	PhoneExt	Ŧ	Title	-	
			6	f.landis@nano	Fred	Land	dis			951	Program Ma	ana	
			7	m.landis@nan	Michelle	Butl	er			654	Software Er	ngir	
			8	t.little@nanon	Ted	Littl	e			852	Software Er	ngir	
			9	a.way@nanon	Ann	Way	1			951	Technical W	/rit	
			10	l.gardner@nar	Liz	Gard	Iner			753	Software Er	ngir	
			11	p.smith@nanc	Paul	Smit	th			334	Software Er	ngir	
		*											
	P		4	1 IT 8	& Technical	_		_				_	_
		4	PrimaryKey 👻	EMail 👻	First	*	Last	*	PhoneExt	*	Title	-	
			12	a.martin@nan	Alex	Mar	tin			778	IT Manager		
			13	g.hammer@na	George	Ham	mer			223	Web Devel	ope	
			14	j.band@nanon	Jessica	Ban	der			241	Support Eng	gine	
			15	I.king@nanon	Lui	King	5			345	Support Eng	gin€	-
Re	0	米	4 1 of 3	N NS V. No Fil	ter Search								
Re	con	an i		XIVII	Jearch								

上記のように、「Administration」部署は、3人の社員、「Marketing」部署には、2人の社員、「Engineering」部署 には6人の社員が所属しています。

この データベースからデータをマッピングする場合、各社員が正確な部署に割り当てられる場合、「ルート」 テーブ ルとして、Department を使用することは重要で、Department の子である Person テーブルからマップします:



上のマッピングは、 <マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥ フォルダー内の DB_Altova_Hierarchical.mfd ファイルに変更を加えたマッピングです。マッピングをプレビューする際には、各個 人の結果は、意図とされる振る舞いであるとおり、部署別にグループ化されます。「Administration」 によ3人、 「Marketing」 には2人、「Engineering」 にお人、などの結果が得られます。

1	xml version="1.0" encoding="UTF-8"?
2	Altova xsi:noNamespaceSchemaLocation="file:///C
3	 <office></office>
4	Oppartment>
5	<name>Administration</name>
6	Person>
12	Person>
18	Person>
24	
25	Operation of the second sec
26	<name>Marketing</name>
27	Person>
33	Person>
39	
40	Oppartment>
41	<name>Engineering</name>
42	<person> </person>
48	Person>
54	Person>
60	Person>
66	Person>
72	Person>
78	-
79	Operation of the second sec
106	Operation of the second sec
121	Operation of the second sec
130	Operation of the second sec
151	
152	└

若干修正の加えられた下のマピングを確認してくたさい。Department とPersonの両方に接続か意図的に描かれているため両方が「ルート」テーブルとなっています。



マルングをプレビューすると、(ソースの部署に関係なく)すべての個人は、各ターゲメト部署の下にグループ別に表示されています。これは 意図とされた振る舞いではありません。つまり、「Administration」 に21人の社員が存在し、「Marketing」 に21人の社員が存在 し、「Engineering」 に21人の社員が存在するなどという結果を指します。

1	xml version="1.0" encoding="UTF-8"?
2	Altova xsi:noNamespaceSchemaLocation="file:///C:/l
3	
4	Operation of the second sec
5	<name>Administration</name>
6	erson>
12	erson>
18	erson>
24	erson>
30	erson>
36	erson>
42	erson>
48	<person> </person>
54	<person> </person>
60	<person> </person>
66	<person> </person>
72	<person> </person>
78	erson>
84	<person> </person>
90	<person> </person>
96	erson>
102	erson>
108	erson>
114	erson>
120	erson>
126	erson>
132	-
133	Operation of the second sec
262	Operation of the second sec
391	Operation of the second sec
520	Operation of the second sec
649	Operation of the second sec
778	Operation of the second sec
907	
908	└

第2番目の例では、データベースルーションシップは接続の方法のため無視されています。

ですから、データベースレーションシップを保持する場合、リレーションシップを保持する子テーブルを含むことを、おけま同じ「ルート」テーブルから接続が描かれていることを確認してください。これは、ソースとターゲートデータベースと同じようご作動します。リレーションシップを保持するデ ータベースマンピングの例に関しては、〈マイドキュメント〉Witova Map Force 2021 Map Force Examples ¥ フォルダー内にある DB_Altova_Hierarchical.mfd と Altova_Hierarchical_DB.mfd ファイルを参照してくたさい。(次も参照してくたさい、データ を複数にことったれているテーブルに挿入する。

データベースルーションシップを保持しない場合もあります。例えば、altova.mdb データベースからの全てのデータを、SQL/XML 仕様 (Structured Query Language (SQL) 仕様の ペト 14 に準ずる) フラナな XML ファイルロエクスポートすると想定します。この種類 のマナングは、 〈マイドキュメント〉 ¥Altova ¥Map Force 2021 ¥Map Force Examples ¥ フォルダ内の DB_Altova_SQLXML.mfd サンプルにより説明されています。 マナングの目的は、フラナな XML ファイルとしてデータベースデータを 取得することです。 ターゲナト SQL/XML スキーマは、XMLSpyで「変換 | DB 構造から XML スキーマを作成する」 メニューコマナド を使用して生成することができます。



DB_Altova_SQLXML.mfd

上記のように、各データベーステーブルコよ、対応する要素がターゲット XML 内に存在します。マンピングの結果をルビューすると、実際のデ ータベースの行は、ターゲット内の「行」要素に書き込まれていることが確認されます。

2 - (altova xsinoNamespaceSchemaLocation="file:/// 3 - (altova xsinoNamespaceSchemaLocation="file:/// 4 - (Address> 5 - (Address> 21 - (Address> 22 - (Altova> 23 - (Address> 24 - (Address> 25 - (Altova> 26 - (Altova> 27 - (Altova> 28 - Oppartment> 29 - (Altova> 39 - (row> - (row> 44 - (row> - (row> 45 - (Oppartment> 65 - (Office> 66 - (row> - (row> 61 - (Opfice> 62 - (row> - (row> 76 - (row> - (row> 77 - (row> - (row> 88 - (row> - (row> 97 - (row> - (row> 1	1	<2vml version="1.0" encoding="UTE 8"2>
3 <altova> 4 <address> 5 <row> </row> 13 <row> </row> 13 21 23 <row> 23 <row> 24 25 26 27 28 28 39 39 <row></row></row></row></address></altova>		Sexini version= 1.0 encoung= 011-0 ?>
3 G <altova> 4 G <altova> 5 G <row> </row> 13 G <row> </row> 21 </altova> 23 G <row> </row> 24 Cow> 25 G <department> 26 <department> 27 28 Cow> 29 G <row> 29 G <row> 29 G <row> 29 G <row> 29 G <row> 39 G <row> 39 G <row> 44 G <row> 44 G <row> 54 G <row></row></row></row></row></row></row></row></row></row></row></department></department></altova>	2	<pre>attova xsi:nonamespaceSchemaLocation="file:///</pre>
4 <address> 5 <row> 13 <row> 13 <row> 21 22 < 23 <row> 23 <row> 23 <row> 24 29 <row> 34 <row> 39 <row> <</row></row></row></row></row></row></row></row></row></address>	3	⊖ <altova></altova>
5 	4	
13 < <<<<<<li< th=""><th>5</th><th>row></th></li<>	5	row>
21 22 23 <row> 27 28 Opepartment> 29 <row> 34 <row> 39 <row> 39 <row> 44 <row> 49 <row> 54 <row> 59 <row> 64 55 Office> 66 <row> 76 <row> 76 <row> 76 <row> 76 <row> <</row></row></row></row></row></row></row></row></row></row></row></row></row></row>	13	row>
22 <itowa> 23 <rowa> 27 <itowa> 28 Oppartment> 29 <rowa> 34 <rowa> 39 <rowa> 39 <rowa> 44 <rowa> 45 <rowa> 54 <rowa> 54 <rowa> 55 66 <rowa> 66 <rowa> 76 87 97 <</rowa></rowa></rowa></rowa></rowa></rowa></rowa></rowa></rowa></rowa></itowa></rowa></itowa>	21	-
23 <row> </row> 27 28 Obepartment> 29 <row> </row> 34 <row> </row> 39 <row> </row> 44 <row> </row> 44 <row> </row> 44 <row> </row> 54 <row> </row> 54 <row> </row> 64 65 66 <row> </row> 67 76 <row> </row> 77 88 <row> 97 <row> 115 <row> 124 <row> 133 <row> 160 <tow> <th>22</th><th></th></tow></row></row></row></row></row>	22	
27 28 29 <row> 34 39 <row> 34 39 <row> 44 <row> 49 <row> 44 <row> 54 <row> 59 <row> 64 76 <row> 66 <row> 76 76 <row> 76 <row> 76 <row> 77 78 <row> 79 <row> 70 715 <row> 724 733 74 <row> 75 <row> 76 <row> 77 <row> 78 <row> 79 <row> 70 <row> 715 <row> 716 <row> 717 <!--</th--><th>23</th><th>row></th></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row>	23	row>
28 <department> 29 <row> 34 <row> 39 <row> 39 <row> 44 <row> 49 <row> 44 <row> 49 <row> 59 <row> 64 65 66 <row> 76 <row> 76 <row> 76 <row> 76 <row> 76 <row> 76 <row> 77 <row> 78 <row> 797 <row> 115 <row> 124 <row> <row> 133 <row> <row> 142 <row> <row> <</row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></department>	27	-
29 	28	Operation of the second sec
34 <row> 39 <row> 44 <row> 49 <row> 54 <row> 59 <row> 64 65 66 <row> 76 <row> 76 86 97 <row> 97 <row></row></row></row></row></row></row></row></row></row></row>	29	<pre>def sectors and sectors a</pre>
39	34	<pre>crow></pre>
44 • <row> </row> 49 • <row> </row> 54 • <row> </row> 59 • <row> </row> 64 • 65 • 66 • <row> 76 • <row> 86 97 • <row> 97 • <row> 106 • <row> 115 • <row></row></row></row></row></row></row>	39	<pre>crow></pre>
49 •	44	<pre>crow> </pre> /row>
54 • <row> 59 • <row> 64 • 65 • 66 • <row> 66 • <row> 76 • <row> 86 - 76 • <row> 86 - 97 • <row></row></row></row></row></row></row></row>	49	A <rows <="" rows<="" th=""></rows>
54 59 64 65 66 67 68 76 76 76 76 77 88 97 133	54	
59 	50	
04 65 66 76 86 87 97 97 106 115 124 133 142	55	
05 66 <row> 76 <row> 86 87 <person> 88 <row> 97 <row> 106 <row> 115 <row> 124 <row> 133 <row> 142 <row> 151 <row> 160 <row> 178 <row> 187 <row> 187 <row> 187 <row> 205 <row> 214 <row> 223 <row> 224 <row> 250 <row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></row></person></row></row>	04	
00 (row) (row) 76 (row) (row) 86 (Office) 87 (Person) 88 (row) (row) 97 (row) (row) 106 (row) (row) 115 (row) (row) 124 (row) (row) 133 (row) (row) 142 (row) (row) 151 (row) (row) 160 (row) (row) 178 (row) (row) 187 (row) (row) 187 (row) (row) 187 (row) (row) 205 (row) (row) 214 (row) (row) 223 (row) (row) 232 (row) (row) 234 (row) (row) 235 (row) (row) 241 (row) (row) 250 (row) (row) 268 (row) (row) </th <th>00</th> <th></th>	00	
76 (-rows) 86 87 97 106 115 124 133 142 151 160 142 151 160 160 160 178 187 205 214 223 <td< th=""><th>00</th><th><row> </row></th></td<>	00	<row> </row>
86 87 88 97 106 115 124 133 142 151 151 160 178 178 178 187 187 187 187 187 187 187 205 214 214 214 214 214 216	/6	<pre></pre>
87 88 <row> </row> 97 <row> 106 <row> 115 <row> 124 <row> 133 <row> </row></row></row></row></row>	86	
88 e <row> </row> 97 e <row> </row> 106 e <row> </row> 115 e <row> </row> 124 e <row> </row> 133 e <row> </row> 134 e <row> </row> 135 e <row> </row> 136 <row> </row> 142 e <row> </row> 151 e <row> </row> 160 e <row> </row> 160 e <row> 178 e <row> 187 e <row> 187 e <row> 196 e <row> 205 e <row> 214 e <row> 223 e <row> 2241 e <row> e <row> 250 e <row> e <row> </row></row></row></row></row></row></row></row></row></row></row></row>	87	⊖ <person></person>
97 (************************************	88	<row></row>
106 @ <row> </row> 115 @ <row> </row> 124 @ <row> </row> 133 @ <row> </row> 133 @ <row> </row> 142 @ <row> </row> 151 @ <row> </row> 160 @ <row> </row> 160 @ <row> </row> 160 @ <row> </row> 178 @ <row> </row> 187 @ <row> 187 @ <row> 187 @ <row> 187 @ <row> 196 @ <row> 205 @ <row> 214 @ <row> 223 @ <row> 2241 @ <row> 9 @ <row> 250 @ <row> 277 _ 28 @ <row> 277 <t< th=""><th>97</th><th><pre>erow> </pre></th></t<></row></row></row></row></row></row></row></row></row></row></row></row>	97	<pre>erow> </pre>
115 (row) (row) 124 (row) (row) 133 (row) (row) 133 (row) (row) 142 (row) (row) 142 (row) (row) 151 (row) (row) 160 (row) (row) 160 (row) (row) 178 (row) (row) 178 (row) (row) 187 (row) (row) 187 (row) (row) 187 (row) (row) 205 (row) (row) 214 (row) (row) 223 (row) (row) 232 (row) (row) 241 (row) (row) 250 (row) (row) 259 (row) (row) 277 (/Person) (/row) 278 (/altova) (/altova)	106	<row></row>
124 Image: style="text-align: center;">(row > 133 Image: style="text-align: center;">(row > 133 Image: style="text-align: center;">(row > 142 Image: style="text-align: center;">(row > 142 Image: style="text-align: center;">(row > 151 Image: style="text-align: center;">(row > 151 Image: style="text-align: center;">(row > 160 Image: style="text-align: center;">(row > 178 Image: style="text-align: center;">(row > 178 Image: style="text-align: center;">(row > 187 Image: style="text-align: center;">(row > 187 Image: style="text-align: center;">(row > 187 Image: style="text-align: center;">(row > 205 Image: style="text-align: center;">(row > 214 Image: style="text-align: center;">(row > 223 Image: style="text-align: center;">(row > 232 Image: style="text-align: center;">(row > 250	115	<row></row>
133 	124	<row></row>
142 • <row> </row> 151 • <row> </row> 160 • <row> </row> 169 • <row> </row> 178 • <row> </row> 178 • <row> </row> 187 • <row> </row> 196 • <row> </row> 205 • <row> </row> 214 • <row> </row> 223 • <row> </row> 232 • <row> </row> 250 • <row> </row> 250 • <row> 250 • <row> 250 • <row> 250 • <row> 250 • <row> 277 • 278 • */altova></row></row></row></row></row>	133	<row> </row>
151 Image: constraint of the second seco	142	<row> </row>
160 • <row> </row> 169 • <row> </row> 178 • <row> </row> 187 • <row> </row> 187 • <row> </row> 196 • <row> </row> 205 • <row> </row> 214 • <row> </row> 223 • <row> </row> 232 • <row> </row> 232 • <row> </row> 232 • <row> </row> 250 • <row> </row> 250 • <row> 250 • <row> 250 • <row> 250 • <row> 276 277 - 278 279</row></row></row></row>	151	row>
169 <row></row> 178 <row></row> 187 <row></row> 187 <row></row> 196 <row></row> 205 <row> 214 <row> 223 <row> 2241 <row> 250 <row> 251 <row> 252 <row> 268 <row> 277 278 279</row></row></row></row></row></row></row></row>	160	row>
178 • <row> </row> 187 • <row> </row> 196 • <row> </row> 205 • <row> </row> 214 • <row> </row> 223 • <row> </row> 232 • <row> </row> 241 • <row> </row> 250 • <row> </row> 259 • <row> </row> 268 • <row> </row> 277 - 278 - 279 -	169	row>
187 • <row> </row> 196 • <row> </row> 205 • <row> </row> 214 • <row> </row> 223 • <row> </row> 232 • <row> </row> 232 • <row> </row> 241 • <row> </row> 250 • <row> </row> 250 • <row> </row> 250 • <row> </row> 250 • <row> </row> 268 • <row> </row> 277 - 278 - 279 -	178	row>
196 (row) (row) 205 (row) (row) 214 (row) (row) 223 (row) (row) 223 (row) (row) 223 (row) (row) 232 (row) (row) 241 (row) (row) 250 (row) (row) 259 (row) (row) 268 (row) (row) 277 (row) (row) 278 (row) (row) 279 (row) (row)	187	<row></row>
205 Image: constraint of the second seco	196	row>
214 Image: style="text-align: center;">Image: style="text-align: center;">Image: style="text-align: center;">Image: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: center;">Image: style="text-align: style="text-align: style="text-align: style="text-align: center;">Image: style="text-align: style="	205	row>
223 Image: style="text-align: center;">Image: style="text-align: center;">Image: style="text-align: center;">Image: style="text-align: style="text-align: center;"/> 223 t	214	even with the second
232 Image: style="text-align: center;">(row> 241 Image: style="text-align: center;">(row> 250 Image: style="text-align: center;">(row> 250 Image: style="text-align: center;">(row> 259 Image: style="text-align: center;">(row> 268 Image: style="text-align: center;">(row> 277 (row> 278 (row> 279 (row>	223	<pre>crow></pre>
241 Image: constraint of the second seco	232	<pre>crow></pre>
250 Image: constraint of the second seco	241	<pre>crow> </pre> /row>
259 Image: style="text-align: center;">Image: style="text-align: center;"/>Image: style="text-align: center;"/>Image: style="text-align: center;"/>Image: style="text-align: center;"/>Image: style="text-align: center;"/>Image: style="text-align: center;"/>Image: style="text-align: center;"//// Image: style="text-align: center;"////// Image: style="text-a	250	<row></row>
268 <row> 277 </row> 278 279	259	<pre>srow>llllllllllllllllllllllllllllllllllll</pre>
277	268	<pre>crow> </pre> /row>
278 279	277	-
279	278	-
210 Statovar	279	
	213	State TO P

XML 出力が示すように、XML 要素間には、階層は存在しません。これはフラオな SQL/XML 構造です。複数の「ルート」 テーブルか らのデータが内部でマップされたちがに、データベースルーションシップは、無視されます。

6.2.2.6 ローカルリレーションシップの定義

データベーステーブルが明示的にリレーションシップを定義していない場合、MapForce内でそのようなリレーションシップをローカルに定義す ることができます。特に、MapForce内から、異なるテーブルの別の間の主キーと外部キーリレーションシップをデータベースに影響を与えるこ どな作成することができます。すべてのデータベース列を主キーまけよ外部キーを作成するために使用することができます。また、データベー

ス内に既存のルーションに加え、新規のルーションを素早く作成することができます。ローカルで定義されたリルーションシップはマッピングと 共に保存されます。

これらの素早く作成されたルーションシップは、MapForce内ではローカルルレーションと呼ばれます。次のテーブルはローカルルレーションを定義することのできるすべての可能なフィールドをリストしています(例えば、ストアドプロシージャの出力をデータベース列にマッピングするなど。・ルーションシップ内の一部のフィールドは、同じ、まけよ互換性のあるデータ型を持つ必要があります。

主/ユニークキー	外部+
 データベーステーブル おけボューの列 ストアドプロシージャの出力/ デメーターおけま見り値。ストアドプロシージャを参照してくたさい。 ストアドプロシージャイニより返されオシコードセトの列* ユーザー定義 SELECT ステートメストの列。仮想テーブルとしてのSQL SELECT ステートメスト を参照してくたさい。 	 データベーステーブル おゴゼューの列 ストアドプロシージャの入力/ ウメーター ユーザ 定義 SELECT ステートメントの入力/ ウメーター

*(パラメータを持たない) データソースまたは、(入力と出力パラメーターを持つ)関数としてストアドプロシージャが呼び出された場合 適用することができます。 セクションのためにレコード セットが使用 できるようになるには、 レコード セットを取得するためにストアドプロシ ージャを一度実行する必要かあります。

この例で使用されている、altova-no-relation.mdb データベースは、MapForce により与えられる altova.mdb データベースの簡素化された、・ジョンです。Person ならびに Address テーブルや、関連するテーブルが Microsoft Access。 altova-no-relation ツノー こあるテーブルは、全て「ルート」レベルのテーブルで、子テーブルはかりません。 各テーブルのエンテンソは、そのテーブルに含まれるものだけ に限定されています。



明示的なリレーションシップを持たないデータベース構造

この別の目的は、「Altova」のオフィスを表示し、各オフィス内の部署を表示することです。altova-no-relation.mdb 内では、主キーと 外部キーリーションシップは、上記のように、明示的には存在しません。このリーションシップは、論理的に(明示的に)存在し、マノピングの目的を完成するためコーカルで再作成されます。

ローカルルーションは、データベースをマッピングに追加中に定義することができます、おけま既存のデータベースコンポーネントを右クトックして、おけまことテキストメニューから「データベースオブジェクトの追加/削除/編集」を選択して定義することができます。

- 1. 「挿入」メニューから「データベース」をクリックします。
- 2. 「接続ウィザード」内のMicrosoft Access (ADO) をクリックして、「次へ」をクリックします。
- 3. <マイドキュメント>糾ltova糾apForce2021糾apForceExamples¥Tutorial¥フォルダー内にあるaltova-norelation.mdb データベースを参照し、「接続」をクルクします。
- 4. ユーザーテーブルチェックボックスを選択します。

(戸データベース オブジェクトを挿入	×
テーブル、ビュー、ストアドプロシージャを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SQL SELECT ステートメント はローカルの mfd ファイルに 保存されます。	y
ックス ▼ ▼ ◆ SELECT ステートメントの追加/編集… リレーションの追加/編集 (0)… レコードセット構造の追加/編集	
H M 18 altova-no-relation H B V G altova-no-relation 비 B 2 ブル	
■ Community Communis Community Community Community Community Community Co	
プレビューを表示(P) >> OK	キャンセル

- 5. アイエンバーコあるルーションを追加/編集ボタンをクリックします。
- 6. リレーションを追加/編集ダイアログボックスか開かれますので、リレーションを追加ボタンをクトックします。

1 リレーションの追加/編集	×
このダイアログではデータベースにて宣言されたもの加えて、新たなリレーションを宣言することが に対して、主キーオブジェクトと外部キーオブジェクト、そして少なくとも1つのデータタイプの列ベ い。	できます。各リレーション プをを選択してくださ
主キー/一意キーオブジェクト 外部キー オブジェクト □ [テーブル オブジェクト] □ □ [テーブル オブジェクト] □ □ [最初にオブジェクト 選択] □	OK キャンセル
	リレーションを追加 リレーションを削除
	列の追加 列の削除
-	

7. 2 つのエレボド・クスによりテーブルオオオデータベースオブジェクト間のルーションを選択することができます。左側のエレボド・クスが 主キー/ユニークキーのテーブルで、右側が外部キーのテーブルゴンります。2つのドロップダサノストにより、ルーションを作成するテ ーブル、おオオデータベースオブジェクトを選択することができます。「主/一意のキー」を保管する左側のノストは、オブジェクトを指 定し、右側のノストはか部キーオブジェクトを指定します。主キー/一意キーオブジェクトがMapForce における親オブジェクトとな り、外部キーオブジェクトがデータベースエレポーネントの子として表示されます。(次も参照してくたさい、データベースルーションシッ プの処理)。

№ リレーションの追加/編集	×
このダイアログではデータベースにて宣言されたもの加えて、新たなリレーションを宣言することが に対して、主キーオブジェクトと外部キーオブジェクト、そして少なくとも1つのデータタイプの列引い。	りできます。各リレーション ペアをを選択してくださ
主キー/一意キーオブジェクト 外部キー オブジェクト 印皿 Altova ■ 回 Office 1 回 PrimaryKey (INT) ■ ForeignKey (INT)	OK キャンセル
	リレーションを追加 リレーションを削除
	列の追加 列の削除

8. 「OK」ボタンをクリックしてローカルルーションの定義を完了し、「挿入」ボタンをクリックして、データベースをマッピングエアへ挿入します。

この段階では、「Altova」テーブルの「PrimaryKey」列と「Office」テーブルの「ForeignKey」列の間にコーカルルーションシップ を作成します。下に表示されるとおり、「Altova」ルートテーブルは、「Office」テーブルの親コなりました。具体的には、Office テーブル は、Altova テーブルの下の、自身の展開することのできるアイコンを持つ、関連するテーブルとして表示されています。

🖯 altova-no-relation	
>🖵 🛄 Altova	
PrimaryKey	
> III xmlns	
> 🔲 ipo	
> 🔲 xsi	
In schemaLocation	
> 🔤 🔟 Name	
↓ ··· ← 🕀 📶 Office ForeignKey	
] ≥⊞ IIII Department	
}⊞ III Office	

しかしなから、マパングの目的はまだ達成されていません。マパングの目的を完成するために、下に表示されるとおり、Officeと Department テーブル間のルーションシップを作成するメンバと同じメンバを使用します。

💁 リレーションの追加/編集			×
このダイアログではデータベースにてき に対して、主キーオブジェクトと外部 い。	宣言されたもの加えて、新たなリレーションを 3キーオブジェクト、そして少なくとも1つのデー	宣言することが -タタイプの列バ	できます。各リレーション プアをを選択してくださ
主キー/一意キーオブジェクト 日団 Altova 『PrimaryKey (INT) 日団 Office 『PrimaryKey (INT)	外部キーオブジェクト ■ Office ■ ForeignKey (INT) ■ Department ■ ForeignKey (INT)		OK キャンセル リレーションを追加 リレーションを削除
		Ŧ	列の追加 列の削除

「リレーションの追加/編集」ダイアログボックスを再度開くには、データベースコンポーネントを右クトックし、「データベースオブジェクトの追加/削除/編集」コンテキストメニューから選択します。

最後に、ターゲナスキーマをマピングに以下のように追加します

- 1. 「挿入」メニューから「XML スキーマ/ファイルの挿入」をクトックします。
- 2. <マイドキュメント> Altova Map Force2021 Map ForceExamples ¥フォルダー内で使用することのできる Altova_Hierarchical.xsd ファイルを参照します。サンプルXML ファイルを与えるようプロンプトされると、「スキップ」をク リックします。ルート要素を選択するようプロンプトされると、「Altova」を選択します。

ターゲオ XML 内のテーブル間のルーションシップを保持するために、全ての接続は、同じ「ルート」テーブルから、階層的(この場合、「Altova」) 「描かれています。詳細に関しては、次を参照してください、データベースルーションシップの処理。



マピング上記のように、マピングを定義し、「出力」タブをクリックして、結果をプレビューします。マピングの結果は、このマピングの目的のとおり、それぞれのオフィスの下にネスとされた部署の要素を表示しています。



6.2.2.7 データベースを変更するマッピングの実行

(例えば、レコードの挿入、更新、削除など)マレングがデータベースのデーダを変更すると、変更は、マレングを実行したエンジイことデ ータベースに適用されます。マレングを実行するエンジノは、MapForce、(スタンドアロン おけは FlowForce Server 管理下での) MapForce Server、おけは C++、C#、おけは Java のために生成されたコードの実行環境などです。

(「出力」ダブをクトックして、) MapForce 内で直接マッピングの結果をパビューすると、更新スクリプトが表示されます。スクリプトは、このアクションを明示的に実行するまで、データベースに対して実行されません。これはパビューのナックのみに使用することができます。

1	/*
2	The following SQL statements are only for preview and may not be executed in another SQL query too!
3	To execute these statements use function "Run SQL-script" from menu "Output".
4	Connect to database using the following connection-string:
5	Data Source=C:/Users/altova/Documents/Altova/MapForce2016/MapForceExamples/AltovaTarget.mdb;Provider=Microsoft.Jet.OLEDB.4.0
6	*/
7	
8	begin transaction
9	
10	DELETE FROM [Address]
11	
12	DELETE FROM [Person]
13	
14	DELETE FROM [Department]
15	
16	DELETE FROM [Office]
17	
18	DELETE FROM [Altova]
19	
20	SELECT IIF(MAX([Altova].[PrimaryKey]) IS NULL,0,MAX([Altova].[PrimaryKey]))+1 AS [PrimaryKey] FROM [Altova]
21	>>> %PrimaryKey1%
22	
23	INSERT INTO [Altova] ([Name], [PrimaryKey]) VALUES ('Organization Chart', '%PrimaryKey1%')

データベースを変更するマッピングの出力プレビュー(Altova_Hierarchical_DB.mfd)

スクリプトは疑似-SQL ステートメイを情報提供の目的で表示しています。実行されようとているステートメイを直接反映していません。例えば(「Update if... Insert Rest"などの複数のアクションカテーブルについて定義されている場合、プレビュー内では、2番目のアクションは条件付きで実行されるナメ最初のアクションのみが表示されています。

疑似-SQL スクリプトは上記の実行エンジン以外 SQL ツールを使用してデータベースに手動で適用されることはできません。スクリプト は外部 SQL エディターにより理解不可能な値のフォーマナを含んでいる可能性があります。

マピングへの変更をデータベースへ直接 MapForce から適用する場合、「出力」メニュー内で使用することのできる「SQL-スクリプトの実行」コマンドをクリックします。このアグンコンは、実際にデータベースを変更し、効果はすく「こ適用されます。

(スタイアロン おは Flow Force Server 管理下で) Map Force Server でマピングが実行される場合、データベースへの変更はすぐ に適用されます。生成されたコード内でも同様が発生します。コードをコンパイルし実行した際に、データベースへの変更は適用されます(例 えば、 Visual Studio 内で「実行」コマイドをクリックしてなど)。

MapForce のインストールコよ 〈マイドキュメント〉 ¥ltova ¥lap Force2021 ¥lap ForceExamples ¥ フォルダー内に存在 する、複数のサンプルデータベース (Microsoft Access ま) は SQLite ファイル) か含まれます。 MapForce から「SQL-スクリプ トの実行」コマイドを使用して、 MapForce により与えられるサンプルの データベース ふれてデータベースへの変更を適用することが 奨励されません これは、サンプルが使用できなくなる可能性かあるからです。 オリジナルのデータへの上書を回避するけっかく、 〈マイ ドキュメント〉 ¥ltova ¥lap Force2021 ¥lap ForceExamples ¥ フォルダー全体の ジングアップをファイルを更新する前に作成す ることの最善策です。

MapForce 以外の実行環境内でのマピングの実行に関する情報は、以下を参照してくたさい

- <u>FlowForce Server にマンピングをデプロイする</u>
- <u>MapForce Server 実行可能ファイルにマンピングをコンパイルする</u>
- <u>コード ジェネレーター</u>

6.2.2.8 特殊文字の使用

データベースデータを変換する際に、データノースから Enter キー/改行(CR/LF) 文字などの、固有の特別文字を削除する必要かある可能性があります。ソース。これは、MapForce ライブラノ関数char-from-code を使用して行うことができます。

テーブル「Lines」は、2つの列(「ID」と「Description」)か構成されています。Microsoft Access データベースが以下に表示されています。

	Lines			
4	ID 👻	Description 👻	Click to Add	Ŧ
	1	This is our new company policy.		
	2	It will be implemented immediately.		
*	(New)			

目的は、CSV ファイルの詳細(ラインごとの詳細)を抽出することです。ですから、この目的を達成するためのマメピングは以下に類似します:



しかしなから、Access 内の各「Description」行は、意図する結果ではない、改行を含むCR/LF 文字により区切られた複数のラインを 含んでいます:



この問題を解決するけっかに、マンングにMapForceビルトインライブラルらchar-from-codeとreplace関数を追加します (マンング)、関数を追加するも参照してくたさい)。上記で説明されている文字が発生すると、スペース文字で置き換えられるように、すべて の詳細が処理されなくてはないません。

Unicode チャート(<u>http://www.unicode.org/charts/</u>)では、LF とCR 文字は、**hex 0A | dec 10** と**hex 0D | dec 13** 文字 にそれぞれ対応します。ですから、マピングは、更にreplace 関数を使用した処理を行うかに、小数のUnicode 値 13を文字列に 変換するように変更されなければなりません。



マピングをこの時点でプレビューすると、各データベースフィールド内のCRLF文字がスペースと置き換えられていることに注目してくたさい。

1	This is our new company policy.
2	It will be implemented immediately.
3	

6.2.2.9 Null 値の処理

マピングランタイムでデータベースフィールドがnull 値であるかをチェックするコよ、<u>is-null</u>と<u>is-not-null</u> MapForce ライブラノ関数を使用します。MapForce から、テーブル内にnull フィールドが存在するかを確認するコよ、データベースブラウザーを使用してクエルます、次を参照: データベースの参照とクエル。

データベースフィールドをnull に設定するために、<u>set-null</u> 関数を使用します。

データベースのnull 値を文字列と置き換えるコよ。<u>substitute-null</u> 関数を使用します。この機能を説明する DB_ApplicationList.mfd は、<マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥フォルダー内にあります。 す。

データベースを更新するマッピング内のNULL値の比較に関しては、データベーステーブルアグション内のNull値の処理を参照してくたさい。

XMLドキュメトから、マンピングをデータベースにマップする、おけまデータベースからXMLドキュメトにデータをマンピングする際のnull値の処理に関する情報は、次を参照してくたさい<u>Nilの値 / Nillable</u>。

6.2.2.10 連続する、および、一意の値の生成

データを挿入し、データベースを更新するコよ、マップされたノースからの入力がは、データベースフィールドのためコシーケンシャル、まけま一意の値を作成する必要かある場合があります。このようは場合には、次のビルトインMapForce ライブラノ関数を使用します:

- <u>auto-number</u>("core | generator functions" ライブラル内で使用することができます)。この関数は、数値フィールドのための主キーの植を生成するために通常使用されます。
- <u>create-guid</u>("lang | generator functions" ライブラリ内で使用することができます)。この関数は、特定のフィールドの ためのグロー ジルニー意の識別子(16 進でエンコードされた文字列として)を作成するために使用されます。

データベースにより生成された値を使用してデータベースフィールドのための値を書き込むこともできます。このオプションは、データベーステーブ ルアケンョンダイアログボックスで使用することができ、主キーを生成する際にとても役したちます(次を参照: <u>データベーステーブルアケンョン殻</u> 定)。

6.2.2.11 SQL 自動補完の提案

特定のエレテキスト内にSQLステートメートを入力すると、MapForceは、テキストのエートレを自動的に提案する場合があります。自動補完を次のエレテキストで使用することができます:

- SQL Editor (次を参照: <u>データベースの参照とクエリ</u>)
- 「データベーステーブルアケンョン」ダイアログボックス内の「カスタムSQL」テキストボックス(次を参照)
 <u>データベーステーブルアケンョン</u>の設定)
- 「SQL SELECT ステートメトの入力」ダイアログボックズ次を参照 SELECT ステートメトの作成)

1	SEL	
	SELECT	÷

自動補完

「上へ」と「下へ」キーボードキーを使用して、提案のノスト内を移動します。提案されたエイリを選択するコよクルク、おコゴ Enter」を押します。

自動補完の提案の無効化:

- 1. 「ツール」メニューからオプション」をクリックします(おけは Ctrl+Alt+Oを押します)。
- 2. 「データベース」から「SQL エディター」をクリックします。
- 3. 「入力へルペー」から「自動的に開く」チェックボックスをクリアします。

必要に応じて、自動補完の提案を手動で呼び出す

• Ctrl+Space を押します。

6.2.2.12 データベースコンポーネントの設定

データベースコンポーネントをマッピングコリアに追加した後、コンポーネント設定ダイアログボックスから適用することのできる設定を構成することができます。コンポーネント設定 ダイアログボックスを以下の方法で開くことができます:

- コンポーネトを選択して、「コンポーネント」メニューからプロパティ」をクリックします。
- コパーネトをダブルクトックします。
- コンポーネトを右クルクして、「プロパティ」をクルクします。

コンポーネント設定	
- データベース	
データソース:	rce2016¥MapForceExamples¥Accounts.mdb
接続名:	Accounts
データベースの種類:	MS Access (ADO)
接続文字列:	Data Source=C:¥Users¥ [·] ¥Documents¥Altova ¥MapForce2016¥MapForceExamples¥Accounts.mdb; Provider=Microsoft.Jet.OLEDB.4.0
ログイン設定	
ユ <i>ー</i> ザー(U):	
パスワード(P):	
JDBC 設定	
JDBCドライバー(J):	sun.jdbc.odbc.JdbcOdbcDriver
データベース URL(L):	jdbc:odbc:;DRIVER=Microsoft Access Driver (*.mdb);DBQ=
ADO/OLEDB 設定	
データソース(D):	C:¥Users¥ I¥Documents¥Altova¥MapForce2016¥N
カタログ <mark>(A)</mark> :	Accounts
プロバイダ −(∨) :	Microsoft.Jet.OLEDB.4.0
追加オプション <mark>(0)</mark> :	Data Source=C:¥Users¥ ¥Documents¥Altova¥Ma
生成設定 トランザクションを テーブル名からスま	使用(T) キーマ名を除外する(S)
ステートメント実行の	ゆイムアウト
タイムアウト (M):	60 秒 🔲 無限(I)
	ОК キャンセル

データベースコンポーネント設定ダイアログボックス

使用することのできる設定は、以下のとおりです。

データベース

このグループは、データベース接続情報を表示します。「変更」をクルクして異なオニデータベースを選択し、既存のデータベースコンポーネント内のデータベースオブジェクトを際定義します。同じ名前のテーブルへの接続は保持されます。データベースコンポーネントを右クルクして、「追加/削除/データベースオブジェクトの編集」を選択することに以、コンポーネント内のテーブルを変更することができます。

データソース	現在のデータベースの名前を指定します。ファイルベースのデータベースでは、これは ファイルシステム上の みです。
	この設定を使用して相対的まけは絶対的ない やを使用してファイルベースデータベ ースがマルングに追加されるかを決定します。相対的ない やの場合、「データノー ス」にはか やか含まれています。絶対的ない やいコはデータベースファイル名のみか含ま れています。
接続名	接続の名前を指定します。MapForce によしこの名前は自動的に生成されます。通常、データノース名と同じ名前ですが、Altova グロー・シリノース す。通常、データノース名と同じ名前ですが、Altova グロー・シリノノース は接続されている場合、しかかる「エイリアス」内である場合もあります。マナング上で同じ接続名を持つ複数のデータベースコンポーネントが存在する場合、名前は ~~connection1>~、~~connection2>~ などの書式を取ります。
データベースの種類	データベースの型を指定します。
接続文字列	現在のデータベース接続文字列を表示します。この読み取り専用のフィーリドは、 データベース接続の作成時ませま変更時に提供される情報を基に生成されます。

ログイ設定

ログイン設定はすべてのロード生成ターゲオとビルドイン実行エンジンのために使用されます。

<i>ユーザ</i> ー	データベースは接続するさかのユーザー名を変更することができます。 データベースが 接続のさめにユーザー名を必要とする場合必須です。
1&7-+*	データベースは接続するかめの マワードを変更することができます。 データベースが 接続のかりコ マワードを必要とする場合必須です。

JDBC-固有の設定

JDBC 接続が含まれている場合、これらの設定がデータベースに接続するために使用され、生成された Java コード、ませは MapForce Server によし実行されます。

メモ マピングがLinux おさまmacOS マシン上で実行される場合、ADO、ADO.NET、およびODBC 接続はJDBC(およびJDBC 設定が適用されます)に変換されます。次を参照してくたさい異なる実行環境内でのデータベースマピング

JDBC ドライバー	データベースエンポーネントのナダの現在アクティブムドライバーを表示します。デフォ ハトのドライバーは、データベースエンポーネントを定義する際に自動的に入力され ます。必要に応じてこの項目に入力されるドライバーを変更することができます。デ ータベースURL フィールド内のエントリの構文が選択する特定のドライバーご従って いることを確認してくたさい。
データベース URL	現在選択されてるデータベースのURLです。このエイトリはJDBCボライバーフィールド内に入力されてる特定のJDBCドライバー構文にしたからてること

を確認してください。

ADO/OLEDB-固有の設定

マピングにADO 接続が含まれる場合、これらの設定がデータベースは接続するかりに使用されます。生成されたC# おけまC++ コード、おけは indows 上で作動する MapForce Server により実行されます。次を参照してくたさい、<u>多種の実行環境内でのデータベース</u>マピング。データソースとカタログ設定は、ビルトインの実行エンジンでは使用されていません。

データソース	ADO データノースの名前を表示します。
לקאל	ADOカタログの名前を表示します。
<i>プロ/ ビイダ</i> ー	データベースエンポーネントのナックの現在アクティブなプロレイダーを表示します。
追加オプション	追加のデータベースオプノョンを表示します。

生成設定

すべてのロード生成ターゲトとビルドイン実行エンジノに生成の設定は適用することができます。

トランザクションの使用	データベースをターゲトとして使用する場合、 <u>トランザンコン処理</u> を有効化します。エラーが発生すると、ダイアログボックスか開かれ、続行のオプションを選択できます。トランザンシュン処理は、このオプションを選択すると、すべてのデータベースエンポーネントのテーブルのために有効化されます。
テーブル名からスキーマ名を削除する	柔軟性のためにテーブル名だけを残し、データベーススキーマ名を生成されたコード から削除することが可能にないます。
	このオプランにお生成された、MapForce SQL Select ステートメントのみに 対して動作することに注意してくたさい。ユーザー定義 SQL-ステートメントは、仮 想テーブルを作成する際には、変更されません。

ステートメント実行のためのタイムアナ

データベースがターゲトコンポーネトとして使用される場合、実行タイムアナがサーバーの使用状況、交通、長期的に実行されているトリ ガー、および他の要素により発生する場合があります。この設定により、データベースへの接続が閉じられる前のタイムアナの時間を定義する ことができます。設定は、データベースデータをクロリオる際、C#、Java、およびC++ コード内で実行されます。

タイムアオ	実行エンジが、データベースステートメントの実行を中断する前にデータベースのレ スポンスを待つ期間を秒数で定義します。実行タイムアウトのデフォルト設定は60 秒です。
タイムアサトなし	有効化されると、このオプションは実行エンジンノニタイムアナトしないように命令します。

メモ ステートメートの実行のためのタイムアナトはSQLiteデータベースに使用することができません。

データベーストランザクションの扱い

トランザウションの使用	データベースターゲナとしてデータベースを使用する場合トランザクションの処理を有効化します。トランザグションの処理はこのオプションを選択するとデータベースコポ ーネトのすべてのテーブルのナンタン有効化されます。詳細に関してはデータベース 例外の扱いを参照してくたさい。
-------------	---

トレース

トレースレベッレ	ターゲオデータベースエレポーネントに適用可能です。トレーシングが有効化されて いると、データベースコズリして行われるアクションがトレースファイル内でログされます。 詳細に関しては <u>データベーストレースとエラーログ</u> を参照してください。
トレースファイル	データベーストレース情報が書き込まれるファイルを指定します。トレースファイルは XML フォーマナです。トレースファイル構成を参照してくたさい。この、マレキ絶対的 ます」は相対的で、「MFD に対して相対的なすべてのファイルを保存する」ファ イルチェックボックスにより影響されます。

MFD ファイルコ相対的にすべてのファイル マを保存する方法

オプションカ有効化されている場合、MapForce はコンポーネント設定ダイアログ上に表すされているファイルレンを MapForce Design (.mfd) ファイルの場所に対して相対的に保存します。異なるオペレーティングシステム上のMapForce Server マンピングを使用して実行 する場合、相対的ない父を使用してください。コンポーネント上の相対的ない父の使用を参照してください。

6.2.3 データをデータベースにマップする

このセクションは、MapForce によりサポートされるマピングノースからデータを転送する例と方法について説明しています、例えば、XMLファイル)をターゲルデータベースは転送するなど。以下のテーブルは、使用することのできるオプタンのまとがをリストしています。

希望するアクション	このトピックを参照する
マピングには提供されたデーダを基にして、ターゲトデータベース テーブルにデーダを挿入します。	• <u>デーダテーブルご挿入する</u>
主キーの値の作成方法を管理します。	 <u>データをテーブルは挿入する</u> <u>データを複数にしとりされているテーブルは挿入する</u>
「準備中」 SQL ステートメトをテーブルがマピングには変更される前に実行します(例えば、テーブル、おけまカスタム SQL ステートメート内の全てのレコードを削除するなど。)。	 <u>データを複数にレノされているテーブルは挿入する</u> <u>データベーステーブルアケション</u>設定
外部キーによりレクされているテーブル内の階層的なリレーション シップを保持します。	• デーダを複数にレンクされているテーブルに挿入する
条件付きでテーブルを更新します。	 <u>テーブルの更新</u>
条件に従い、レコードをデータベースにマージします(レコードを更新 して、他のレコードを同じテーブルは挿入します)。	 「以下の条件で更新し、残りを挿入」アクション MERGE ステートメント

希望するアクション	このトピックを参照する
外部キー・リレーションシップを使用して他のテーブルコレクされてい るテーブルを更新する際に、データベースの整合性を保持します。	• 親テーブルを更新する際の子テーブルのためのオプション
同じテーブルニオして複数のアクションを定義します(例えば、条件が満たされるとコード」は削除され、それ以外の場合、新規のレコードが挿入されます)	• 「以下の条件で削除…」アクション • 「以下の条件で無視…」アクション
マピングの実行に失敗した場合、データの整合性を保持します。	 トランザブションロール シクの使用
データベーステーブル:複数のレコードをいい/挿入します(複数の INSERT ステートメートを1つのカエリにまとめます)。	・ <u>ハルゆ気送 (MapForce Server)</u>

6.2.3.1 データをテーブルに挿入する

マピングは、他のデータベースを含む MapForce によりサポートされるノースコンポーネントからデータベーステーブルニデーダを挿入することができます。新規に挿入されオノコードの主キーかどのようご作成されるかを柔軟的に構成することができます。例えば、主キーは、データベーステーブル内の既存のキーの値に基づき計算することができます。

この例は、新規のレコードをXML ファイルから既存のデータベーステーブルコ挿入する方法を説明しています。主キーの生成の構成方法についても説明されています。この例は以下のファイルを使用します:

- altova-cmpy.xml データベースコ挿入されるソースデーを含んでいます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲオデータベースです。

全てのファイルはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内にあります。下の説明では、 簡素化のためにこれらのファイルへの完全ない いは省略されています。

この例のマッピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同 じ結果を得ることができます。詳細に関しては、次を参照してくたさい、<u>データベースを変更するマッピングの実行</u>。

マピングの目的は、です。 altova.mdb データベースの「Altova」テーブル内に、新規のレコードとして Altova-cmpy.xml 内の会社 を入力することです。ソース XML ファイルを開くと、「Microtech OrgChart」と呼ばれる会社が「つのみが含まれていることに気がつきま す。ですから、マピングはとしら名前「Microtech OrgChart」の新規のレコードを「Altova」テーブルに追加しなければなりません。また、新規の主キーが生成される必要があります。

マッピングの目的を達成するために、以下のステップを行います。

ステップ 1: ソース XML コンポーネントを挿入する

• 「挿入」メニューカら、「XML スキーマ/ファイル」をクトックして、Altova_Hierarchical.xsd を参照します。 インスタンスファイ ルを与えるようご問われると、Altova-cmpy.xml を参照してくたさい。

ステップ 2: ターゲナデータベースを挿入する

• 「挿入」メニューカら、「データベース」をクリックして、altova.mdb は接続するためにフィザードのステップを行ってくたさい。(次を参照してくたさい、例 ″altova.mdb ″データベースをマッピング」ご自加する。

ステップ 3: 接続を描く

• 下に表示されるとおりマピングの接続を描きます。



メモ 希望しない接続が自動的に子孫アイテムに作成される場合は、「一致する子を自動的に接続する」が有効化されています。この場合、最後のアウションを元に戻すゴは、メニューオブション「編集 | 元に戻す」を選択します。自動接続オプションを 無効化するゴは、「接続 | 一致する子を自動的に接続する」を選択します。

ステップ 4: 挿入アクションを構成する

1. ターゲナコポーネント上に、アクション:挿入(<u>An</u>)) ボタム表示されていることご注意してください。このボタンは、マンピング からの接続がある各テーブルに表示されます(この場合は、「Altova」テーブル)。このボタンをクリックして、実行されるデータベース アグンヨンの詳細を構成してください(この場合は、入力アグンヨン)。データベーステーブルアグンヨン ダイアログボックス か表示され ます。

🕑 データベーステーブルのアクション - Altova	×
- 最初のレコードの前に実行する SQL ステートメント ◎ なし(N) ◎ 全てのレコードを削除(L) □ 全ての子テーブルにある全てのレコードも削除	
◎カスタム SQL(C):	
- 各レコードに対して実行するアクション 以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て真であれば、指定されたアクションが実行されます。	
Action on record 全て挿入 「 Priman Key マックされた値	7ションを追加(A)
マー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	フションを挿入(I)
U Office	
	フションを削除(D)
■ トランザクションを使用(T) ■ バルク転送を使用 (MapForce Server のみ)(B) バッチ サイズ: 1000 レコード バルク転送はデータベース接続によりサポートされて()ません。	
ОК	キャンセル

2. データベーステーブルアクションダイアログボックス内で、「全て挿入」から「PrimaryKey」の横の max() + 1 オプションを 選択します。

Action on record	全て挿入	-
💵 PrimaryKey	max() + 1	•
💷 Name	マップされた 値	•

このリスト内で使用することのできるオプションは以下を意味します

オプション	説明
マップされた植	ソースデータをデータベースフィールドに直接マップすることができ、全てのデータベースフィール ドのナダの通常の設定です。リレーションを定義することことり、ストアドプロシージャを使用 して、キーの値を与えることもできます。次を参照してくたさい、ストアドプロシージャを使用し た主キーの生成。
max() + 1	データベース内の既存のキーをベースして、キーの値を生成します。例えば、テーブルに主 キー1、2、および3を持つ3つのレコードが存在する場合、max()+1は4 になり

	ます。
	この例では、「Altova」テーブルコは、主キー1を持つレコードはつしか存在せず、 max() + 1 は、期待される新規の主キーの値である2 です。
DB-により生成された	データベースは キーの値を生成するため」「Identity 関数」を使用します。

「Name」の横のオプタンマップされた値はこの列がマッピングから値を直接取得することを示しています。テーブルアケションダイアログボックスで使用することのできる他のオプタンについての説明は、次を参照してくたさい、データベーステーブルアグタン設定。

ステップ 5: マッピングをプレビューし、データベースを更新する

マピングをプレビューするすめゴ出カ」タブをクリックします。データベースに対して実行されるアクションを含む、SQL スクレプトが生成されます。スクレプトは、まだ、データベースを変更していません。これはまたプレビューです。

、 以下の sqL ステートメントは、プレビューのためだけに生成されており、他の sqL クエリー ツールで使用することはできません! これらのステートメントを実行するには、[出力] メニューから [sqL スクリプトの実行] 機能を使用ください。 以下の接続時文字列によりデータベースへ接続することができます: 2 З 4 5 Data Source=C:\Users\t.a.yasaka\Documents\Altova\MapForce2019\MapForceExamples/altova.mdb;Password=•••••;Provider=Micr 6 7 8 SELECT IIF(MAX([Altova].[PrimaryKey]) IS NULL,0,MAX([Altova].[PrimaryKey]))+1 AS "PrimaryKey" FROM [Altova] 9 -->>> %PrimaryKey1% 10 INSERT INTO [Altova] ([Name], [PrimaryKey]) VALUES ('Microtech OrgChart', '%PrimaryKey1%') 11 12

データベースコン対してスクレフトを実行するココお以下を行います

- 「出力」メニューから「SQL-スクリプトの実行」をクリックします。
- メモ MapForce から直接 SQL スクレプトを実行することは、データベースを更新する方法のひとつです。次も参照してくたさい、データ ベースを変更するマンピングの実行。

結果を確認するコよ DatabaseSpy または Access 内で altova.mdb データベースを開きます。新規の Microtech OrgChart」 レコードが「Altova」 テーブルに新規の主キー2 と共に、追加されていることに注意してくたさい。 このレコードためのデータは、 入力 XML インスタンスに基づけたものです。

Ε			Altova			×
2		PrimaryKey 👻	Name	Ŧ		
	+	1	Organization Chart			
	÷	2	Microtech OrgChart			
*						
Re	cor	d: I4 4 3 of 3	🕨 🕨 🐺 No Filter	Se	arch	

データベーステーブルニデータを挿入するマピングの作成が完了しました。現在のテーブルと依存する子テーブルの両方にデータを挿入する。 例に関しては、次を参照してくたさい、データを複数の関連するテーブルは挿入する。

6.2.3.2 データを複数にリンクされているテーブルに挿入する

データベーステーブルは、「親」テーブルである可能性があります。つまり、このテーブルはデータベース内の他のテーブルは外部キー・レルーション シップを使用して参照されている可能性があります。このような場合、マンピングに親テーブルはけでなく、依存する子テーブルにもレコードを挿 入するように構成することができます。例えば、新規の「company」「レコードをデータベーステーブルは挿入する場合、レコードをこの会社に関 連したオフィス、オフィスの部署、社員などのためにも挿入することができます。

この例は、データベースルーションシップを保持しつつ、複数のテーブルニデータを挿入する方法について説明しています。この例は前の例であるデーダをテーブルに挿入するを若干高度にしたい、チジョンです:

- Altova_Hierarchical.mfd 実際のマピングファイルです。
- Altova_Hierarchical.xml データベースは挿入されるノースデーを含んでいます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するためご使用されるスキーマです。
- AltovaTarget.mdb 更新されるターゲナデータベースです。

全てのファイルはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内にあます。下の説明では、 簡素化のためにこれらのファイルへの完全ない いは省略されています。

この例のマッピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同 じ結果を得ることができます。詳細に関しては、次を参照してください、データベースを変更するマッピングの実行。

マピングの目的は、ソースからのデータXML ファイルを持つターゲトデータベース(AltovaTarget.mdb)内のデーを置き換えることで す。XML ファイル構成は、データベース内のテーブルの階層構造に大体対応します。これは、以下のようご構成されて、る組織図です:一 番上の要素は、2つのオフィスと含む会社で、各オフィスコン部署が存在し、各部署には社員が存在します。同じ階層は、「Altova」テーブ ルは会社に対応するAltovaTarget.mdb内に主存在します。このテーブルは、外部キー・リレーションシップを介して、「Office」テーブル 内のレコードにコンプされて、ます。同様に、「Office」は、「Department」にコンプされており、「Department」は、「Person」につ クされて、ます。AltovaTarget.mdbデータベースのレーションシップの図を確認するには、Accessのレーションシップビューでファイル を開きます(次も参照してくたさい、データベースルーションシップの扱い)。

ステップ 1: ソース XML コンポーネントを挿入する

• 「挿入」メニューから、「XML スキーマ/ファイル」をクリックして、Altova_Hierarchical.xsd を参照します。 インスタンスファイ ルを与えるよう、間われると、Altova_Hierarchical.xml を参照してくたさい。

ステップ 2: ターゲナデータベースを挿入する

 「挿入」メニューカら、「データベース」をクリックして、AltovaTarget.mdb に接続するためにフィザードのステップを行ってくたさい。データベースへの接続方法は、altova.mdb への接続方法と同じです。(次を参照してくたさい)例: "altova.mdb" データ ベースをマメピングに追加する)。

ステップ 3: 接続を描く

下に表示されるとおりマメピングの接続を描きます。主キーと外部キーがマップされていないてとに注意してください。下に表示されるように素早く生成されます。



メモ 希望しない接続が自動的に子孫アイテムに作成される場合は、「一致する子を自動的に接続する」が有効化されています。この場合、最後のアウションを元に戻す」ゴス、メニューオプション「編集 | 元に戻す」を選択します。自動接続オプションを 無効化するコス 「接続 | 一致する子を自動的に接続する」を選択します。

ステップ 4: 挿入アクションを構成する

1. ターゲナコンポーネント上で、「Altova」テーブルの横のアクション: 挿入(<u>Alm</u>) ボタンをクリックし、下に示されるとおり、主キ ーのmax() + 1 設定を構成します。この設定に関しては、前の例で更に詳しく説明されています。次を参照してくたさい:<u>データ</u> をテーブルニ挿入する。

🕑 データベーステーブルのア	ックション - Altova			—
最初のレコードの前に実行する のなし(N) ④全てのレコードを削除(L) のカスタム SQL(C):	SQL ステートメント 一全ての子テーブルにある全	てのレコードも削除		_
 各レコードに対して実行するアク 以下のオペレーションを使用し 比較の結果が全て真であれば	ジョン て全ての人力データは DB デー: 、指定されたアクションが実行さ	ブルのデータと比較さ されます。	5れます。	
入力データアクション	全て挿入	•	*	アクションを追加(A)
III Name	max() + 1 マップされた 値	•	-	アクションを挿入(I)
子テーブルのデータ削除 入力子データを無視			*	
 ■トランザクションを使用(T) ■バルク転送を使用(MapFc バルク転送はデータベース接新 	rce Server のみ)(B) パ: 売によりサポートされていません。	ッチ サイズ:	▼ 1000 V⊐-	<u>アクションを肖小除(D)</u>
				ок キャンセル

また、「全てのレコードの削除」オプションが有効化されていること注意してくたさい。これは、この例で希望される振る舞いとおり、新規のレコードが挿入される前に、テーブルからの既存のレコード全てをクリアします。このオプションを無効化すると、(新規の注 キーを持つ)新規のレコードは、既存のレコードに加え、マノピングを実行する都度、既存のレコードに加え、データベースに追加されます。

この例のスコープのために、「子テーブルから全てのレコードの削除」オプシンも有効化されています。このオプションにより、 「Altova」テーブルからレコードが削除されるだけではなく、外部キー・リレーションシップを使用して「Altova」テーブルコレクされて いるすべてのテーブル内のレコードも削除します。子テーブルに子が存在する場合、していたの子テーブルは削除され、依存関係ツリ 一内の最後のテーブルまで削除されます。ルート「Altova」テーブルからレコードを削除することは、データベースの整合性の違反 であり、マメニングの実行は、失敗します。

データベーステーブルアケンョンダイアログボックスで使用することのできる他のオプションに関しての詳細は、次を参照してくたさい、 データベーステーブルアケンョン設定。

- 2. ダイアログボックスを閉じるけっか」「OK」をクリックします。マッピングエリアで、ボタンの外観が DEL, A:In に変更されたことに注意してくがきい。これは、 "Delete" ステートメントが「挿入」 アグションの前に実行されることか構成されていることを示します。
- 3. 「Office」テーブルの横の All ボシンをクルクし、主キーの max() + 1 設定を構成します。

) データベーフテーブルの				
テータベーステーノルの	アクション - Office			
各レコードに対して実行する	アクション			
山下のオペレーションを使用	ᆘᅎᅀᅎᇭᆘᆂᇎᆈᇉᇏᇕ	」」の「ジュークレトレーディ	カキオ	
以下のオハレーンヨンを使用 比較の結果が全て直であれ	して主ての人力テーダは DBテー しば 指定されたマカションが実行	-ノルのナニタと16#X3 Fなれます。	MUE 9.	
	NG(18/2010/00/00/10/00/10/10/10/10/10/10/10/10/	10100 28		
入力データアクション	全て挿入	•	A (アクションを追加(A)
📲 PrimaryKey	max() + 1		l	
🕫 ForeignKey	外部キー			アクションを挿入(I)
🗆 Desc	マップされた 値	_	, i	
EMail	マップされた 値	_		
Established	マップされた値	_		
III Fax	マップされた 値	-		
Name Name	マップされた値			
Phone Phone	マップされた 1値		T	
子テーブルのデータ削除				
入力子データを無視				
Address				
🔲 Department				
			1	フロションた当地会(の)
J			T	アクションを削除(D)
トランザクションを使用()	Γ)			
		(ッチ サイブ・ 「	1000	
□/ハルク転达を1定用 (Ma	peorce Server のか)(B) /			
バルク転送はデータベース打	妾続によりサポートされていません	0		
			OK	キャンセル

4. 「Office」から降順し、各テーブルのためにステップ3を行います。具体的には、「Address」、「Department」、および 「Person」です。これらすべてのテーブルがレト「Altova」テーブルの直接の子孫であることを確認してくたさい、「ルート」テー ブルの詳細とその重要性に関しては、次を参照してくたさい、データベースルーションシップの扱い。

ステップ 5: マピングをプレビューし、データベースを更新する

マルングをプレビューするけっか」ゴ出カ」ダをクリックします。データベースに対して実行されるアクションを含む、 SQL スクリプトが生成されます。 スクリプトは、おビ、データベースを変更していません。 これはまけこプレビューです。

データベースコン対してスクリプトを実行するココお以下を行います

- 「出力」メニューから「SQL-スクリプトの実行」をクトックします。
- メモ MapForce から直接 SQL スクリプトを実行することは、データベースを更新する方法のひとつです。次も参照してくたさい、データ ベースを変更するマンピングの実行。

結果を確認するコよ、Microsoft Accessで「Altova」テーブルを開き、「Altova」テーブルから「Person」テーブルに、XML ファイル からのルーションシップかどのようにデータベースコム注意されているかを確認してくたきい。

4		Dri	ma	arv	Κo	v -		N	ame	2																_
ï		- 11	IIIG	лу	ĸe	у т	0		anne		- bart															
	Īr		Dimension Chail Establis										had		Fax		Mana			Dhana			_			
		4	_	Pr	птагукеу 🔻					esc	•	★ EIV		*	Establis	iea 🔻	Fax	гах	*	Name	2 *		Phone	•	•	
			٦r					11	he o	om	pany v	office@		nanoni	1992-04-	-01	+1 (321)	321) 5	55 51	Nanonul	I, Inc.	+1 (321)		555 51		
			-			Pr	maryKey 👻				Name	*														
					P				1	Ad	minist	ration														
						4	Prir	nar	yKey	-	EN	lail	Ŧ	Fir	st 🔹 👻	Li	ast	-	Pho	neExt 👻	Ti	tle	*			
								1	v.calla	aby@nan		Verno	n	Callal	у			582	Office	e Ma	nage					
										2	f.furth	ner@n	anı	Frank		Furth	er			471	Accou	unts	Recei			
										3	l.mati	se@na	anc	Loby		Matis	e			963	Accou	untir	ng Ma			
						*																				
					+				2	Ma	rketin	g														
					+				3	En	gineer	ing														
					+				4	IT	& Tech	nical !														
				*																						
		_	Ŧ		-			20	n N	larc	h 1ct 3	nevto	offi	no@no	2001-02	.01	+1 (927) 9	55 19	Nanonul	Dartr	+1	(927) 5	55 0		
		*	-					2 011 Marc			11 13t, 2 HEALO		////	ce@na	2001-05	01	.1(527 3	555 10	b- Inditoriu	rraiti	1 71 1	(527)3	555 00:	0.	
		т [.]	_					_																		
1				_																						_

テーブルの整合性のルーションシップを保持しつつ、複数のデータベーステーブルヘデータを挿入するマピングの作成が完了しまた。

6.2.3.3 テーブルの更新

この例は、XML ソースからデータか与えられる既存のデータベーステーブルの更新方法について説明しています。この例は以下のファイルを使用します。

- altova-cmpy.xml データベース 挿入されるソースデー 好含んで、ます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲナデータベースです。

全てのファイルはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内にあます。下の説明では簡素化のために、これらのファイルへの完全ない、スは省略されています。

この例のマピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同じ結果を得ることができます。詳細に関しては、次を参照してくたさい、データベースを変更するマレピングの実行。

マピングの目的は、XMLドキュメトカらの「Person」インスタンズを持つ「Person」テーブル内のレコードを更新することです。XMLファ イル内の各個人は、PrimaryKey 子要素を持ちます。「Person」テーブル内の各個人には、PrimaryKey 列が存在します。デー タベース内にPrimaryKey を持つ個人とXMLファイル内での個人のPrimaryKey か対応するレコードのみか更新される必要かありま す。

マンピングの目的を達成するために、以下のステップを行います。

ステップ 1: ソース XML コンポーネントを挿入する

• 「挿入」メニューから、「XML スキーマ/ファイル」をクリックして、Altova_Hierarchical.xsd を参照します。インスタンスファ イルを与えるように問われると、altova-cmpy.xml を参照してくたさい。

ステップ 2: ターゲナデータベースを挿入する

• 「挿入」メニューから「データベース」をクリックして、altova.mdb に接続するためにウィザードのステップを行ってくたさい。(次を参照してくたさい)<u>例: "altova.mdb" データベースをマッピングに追加する。</u>

ステップ 3: 接続を描く

• 下に表示されるとおりマピングの接続を描きます。



ステップ 4: 更新アクションの構成

1. ターゲナコポーネトで、「Person」テーブルの横のアクション:挿入(<u>Alu</u>)ボタンをクルクします。
- 2. 「レコードに対するアクション」の横の「以下の条件で更新する…」、を選択します。これにより、データベーステーブルアクションを、条件付きの更新アクションで変更します。つまり、条件が満たされる場合のみ、現在のレコードは更新されます(次のステップを参照してくたさい)。
- 3. 「PrimaryKey」値の横にあるオプンコンを、以下に表示されるよう」「equal」に選択します。これは更新の条件を定義します。つまり、データベースの記録は、「PrimaryKey」値が、マンピングがみる「PrimaryKey」値に等し、場合のみ、更新されます。

🕑 データベーステーブルの)アクション - Person			×
最初のレコードの前に実行す ●なし(N) ●全てのレコードを削除(L) ●カスタム SQL(C):	る SQL ステートメント 一全ての子テーブルに	ある全てのレコート	"も削除	
←各レコードに対して実行する 以下のオペレーションを使用 比較の結果が全て真であれ	アクション して全ての入力データは 」ば、指定されたアクション:	DB デーブルのデー が実行されます。	タと比較され;	ます。
入力データ アクション 『PrimaryKey 『ForeignKey EMail EMail Last ()	<u>以下の条件で更新</u> 一致		ج ج ج	ウションを追加(A) クションを挿入(I)
 ■トランザクションを使用(T ■バルク転送を使用 (Map バルク転送はデータペース招 	つ Force Serverのみ)(B) き続によりサポートされてい	ಗ್ರೊಕ ಈಗಸ ತಿರಸಿಂ	; 10	00 VI-K
			ОК	キャンセル

この例では、等値演算子が「PrimaryKey」フィールドに適用されています。これはデータベースを更新する場合、よちるシナ リオです。型のフィールドに対して定義することのできる定義は、必ずしも主キーではおりません。例えば、「最初」と「最後」の フィールドの横のオプションに「equal」を選択することにより、ソースXML内で、最初と最後の名前が同じてあるレコードのみを 更新します。

任意で、NULL equal チェックボックスを選択します。このサンプルでは、ソースXML内のPrimaryKey フィールド、おけはデータベース 内のPrimaryKey フィールドに null 値が含まれていない ケータ、このチェックボックスは 想連性はおりません。 しかしながら、データに null 値が含まれており、このような値を等価として扱うはは、場合、このチェックボックスをチェックする必要があます。 それ以外の場合は、希望しない 結果が生成される場合があります。 詳細に関しては、データベーステーブルアクション内の Null 値の処理を参照してくたさい。

4. ダイアログボックスを閉じるけっとしてくけっという。マンピングに戻り、アクション:挿入 All ポタン かアクション:更新 (All)) ポタイ 変更されたことに注目してくけっとい。これによのテーブルのたっとに更新アクションか構成されていることを示しています。

ステップ 5: マピングをプレビューし、データベースを更新する

マピングをプビューするためゴ出力」タブをクリックします。データベーススプレて実行されるアクションを含む、 SQL スクリプトが生成されます。 スクリプトは、まだ、データベースを変更していません。 これはまだプレビューです。

1	/*
2	The following SQL statements were executed during "Generate output" function.
3	Every single result is written right to the ">>" string.
4	These statements are only for preview and may not be executed in another SQL query too!
5	The database was connected using the following connection-string:
6	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update
	if\altova.mdb;Provider=Microsoft.Jet.OLEDB.4.0
7	*/
8	
9	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Albert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [little] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
10	>>> OK. 1 row(s).
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander', [PhoneExt] = 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
13	>>> OK. 1 row(s).
14	
15	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis', [PhoneExt] = 963, [Title] = 'Accounting Manager' WHERE ([Person].[PrimaryKey] = 3)
16	>>> OK. 1 row(s).
17	
18	UPDATE [Person] SET [ForeignKey] = 2, [EMail] = 'd.Durnell@microtech.com', [First] = 'Dave', [Last] = 'Durnell',
	[PhoneExt] = 621, [Title] = 'Marketing Manager Europe' WHERE ([Person].[PrimaryKey] = 4)
19	>>> OK. 1 row(s).

データベースコン対してスクリフトを実行するココお以下を行います

- 「出力」メニューから「SQL-スクリプトの実行」をクトックします。
- メモ MapForce から直接 SQL スクリプトを実行することは、データベースを更新する方法のひとつです。次も参照してください、データベースを変更するマレングの実行。

6.2.3.4 「以下の条件で更新し、残りを挿入」 アクション

既存のレコードを更新するだけでなく、同じデータベーステーブルに新規のレコードを挿入する必要がある場合があります。このような場合には、MapForceでは、「以下の条件で更新し、残りを挿入」アクション使用することができ、これは以下のように動作します。

- 「以下の条件で更新する」条件がtrueの場合、既存のデータベースの記録は、マピングからのデータによし更新されます。
- 「以下の条件で更新する」条件がfalseの場合、おこ、「残りを挿入」条件が存在する場合、新規のレコードが挿入されます。
- ソースファイル内に対応するエトリが存在しないデータベース内にユコードが存在する場合、これらのレコードは変更されません。

MySQL / MariaDB ODBC メモ

ターゲナデータベースがODBCを使用したMySQL おけまMariaDB の場合、オプションは「影響された行の代わりに一致した 行を返す」がMySQL ODBC コネクタのカーソル、結果 タブで有効化されている必要があります。まけま、データベース接続ウィザー ドを使用して接続文字列を手動で入力する場合Option=2を接続文字列に追加します: Dsn=mydsn; Option=2; MySQL ODBC コネクタからこのオプションを有効化する方法

- 1. Windows キーを押して、"ODBC"を入力します。
- 2. (インストールされた MySQL ODBC コネクタにより32-ビナ、おは64-ビナの) ODBC データノース管理者を実行します
- 3. MapForce マピングによ使用されるデータノース名 (DSN)をクリックし、「構成」をクリックします。

🐺 ODBC Data Source /	ODBC Data Source Administrator (32-bit)			
User DSN System DSN	File DSI	N Drivers Tracing Connection Pooling About		
User Data Sources:				
Name	Platform	Driver Add		
Excel Files MS Access Database	32-bit 32-bit 22.5#	Microsoft Excel Driver (*xls, *xlsx, *xlsm, *xlsb) Microsoft Access Driver (*.mdb, *.accdb) MicSOL ODBC 5 2 Unicedo Driver		
mysqi_dsn	32-01	<u>Configure</u>]	
<		>		
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.				
		OK Cancel Apply Help		

- 4. 使用することのできる高度なオプションを使用するため」「Details >>」をクリックします。
- 5. **Cursors/Results** タブをクリックし、チェックボックス「Return matched rows instead of affected rows」を選択 します。

サンプル

次の例は、XML ソースからのデータをデータベーステーブルゴ更新と挿入して)マージする方法を説明しています。この例は以下のファイルを 使用します:

- altova-cmpy-extra.xml データベースに挿入されるソースデーを含んでいます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲナデータベースです。

全てのファイルはフォルダーくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥内にあります。下の説明では、 簡素化のためにこれらのファイルへの完全ない いば省略されています。

この例のマピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同じ結果を得ることができます。詳細に関しては、次を参照してくたさい、<u>データベースを変更するマピングの実行</u>。 マルングの目的は、ターゲット「Person」テーブルコノースXMLドキュメトカらのすべてのレコードをマージすることです。具体的には、ソースXMLデータ内の各レコードのナックに、マルングは以下を行う必要が決ます。

- XML ファイル内の個人 の PrimaryKey が、データベース内の個人 の PrimaryKey に対応する場合、レコードは更新されます。
- 上記の条件を満たされ、Person テーブル内の既存のレコードは影響を受けません。
- XML ファイル内の個人のPrimaryKey が、ターゲナデータベーステーブルニー致するものが悪い場合、新規のレコードをデ ータベーステーブルに追加してくたさい。

マパングの目的を達成するために、以下のステップを行います。

ステップ 1: ソース XML コンポーネントを挿入する

• 「挿入」メニューから、「XML スキーマ/ファイル」をクルクして、Altova_Hierarchical.xsd を参照します。インスタンスファイ ルを与えるように問われると、altova-cmpy-extra.xml を参照してくたさい。

ステップ 2: ターゲトデータベースを挿入する

• 「挿入」メニューカら、「データベース」をクリックして、altova.mdb は接続するためにフィザードのステップを行ってくたさい。(次を参照してくたさい、例: "altova.mdb" データベースをマソピングに追加する。

ステップ 3: 接続を描く

• 下に表示されるとおりマッピングの接続を描きます。



ステップ 4:「以下の条件で更新し、残りを挿入」アクションを構成する

- 1. ターゲオンポーネイで、「Person」テーブルの横のアクション: 挿入 (All) ボタンをクルクします。
- 2. 「レコードに対するアクション」の横の、「以下の条件で更新する…」を選択します。これはデータベーステーブルアグションを条件 付きの更新アクションで変更します。つまり、条件が満たされる場合のみ、現在のレコードは更新されます(次のステップを参照して くたさい)。
- 3. 「PrimaryKey」値の横にあるオプションを、以下に表示されるよう」「equal」に選択します。つまり、データベースの記録は、「PrimaryKey」値がマンプから与えられるの「PrimaryKey」値に等し、場合のみ更新されます。

ミレコードに対して実行するアクション					
以下のオペレーションを使用し	以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。				
比較の結果が全て真であれば	、指定されたアクションが実行される	₹ 7 .			
入力データアクション	以下の条件で更新		アクションを追加(A)		
🛛 🕅 PrimaryKey	一致		/ / / / / / / / / / / / / / / / / / /		
💵 ForeignKey		=	アクションを挿入(1)		
💷 EMail					
III First					
Last		Ŧ			
< Ⅲ		•	アクションを削除(D)		

- 任意で、NULL equal チェックボックスを選択します。このサンプルでは、ソースXML内のPrimaryKey フィールド、おけまデ ータベース内のPrimaryKey フィールドに null 値が含まれていないため、このチェックボックスはは関連性はありません。しかしな から、データ c null 値が含まれており、このような値を等価として扱うはは、場合、このチェックボックスをチェックする必要があます。そ れ以外の場合は、希望しない結果が生成される場合があります。詳細に関しては、データベーステーブルアグション内の Null 値の 処理を参照してくたさい。
- 5. 「アケションの追加」をクトックします。これは、新規のアケションを既存の「以下の条件で更新する」アクションの右に追加します。新規のアクションを「残りを挿入」として構成します:

各レコードに対して実行するアクション				
以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。				
比較の結果が全て真であれ	1ば、指定されたアクションが実行	されます。		
Action on moord		- びまりた t手 3		
				アクションを追加(A)
ForeignKey		マップされた 値		アカションを挿入の
III EMail		マップされた 値))) 4) C1+/((1)
III First		👤 マップされた 値		
💷 Last		💌 マップされた 値		
Phone Ext		💌 マップされた 値		
🛄 Title		💌 マップされた 値		
			•	
◀	111		•	アクションを削除(D)

上のイメージでは、データベーステーブルアグションは、マピングの目的に従い構成されています。つまり、「以下の条件で更新する」… 条件が満たされる場合のチルコードは更新されます。それ以外の場合、挿入されます。オプション「マップされた値」は、マピングからの値がレコードの全てのフィールドを作成するために使用されることを指定します。

(この例では必要みませんが2つ以上のアウションを同じデータベーステーブルに対して定義することもできます。マノピングランタイムでは、アウションは左から右へ実行されます。最後の「挿入」アウションは最後と考えられ、このアウションの後のアウションは無視されます。

ダイアログボックスの「アクションの追加」 ポタイは、選択されたアクションの後に新規にアクションを追加します。「アクションの挿入」は、新規のアクションを選択されたアクションの前に追加します。既存のアクションを削除するには、アクションをクリックし、を「アクションの削除」 クリックします。

6. ダイアログボックスを閉じるためは「OK」をクトックします。マッピングでは、アクション:挿入 All ボタンが、アクション:更新;挿 入(Allphi) これは、更新と挿入アクションが、このテーブルのために構成されることを示します。

ステップ 5: マッピングをプレビューし、データベースを更新する

マピングをプレビューするオーカリゴ出カ」タブをクリックします。データベースに対して実行されるアクションを含む、 SQL スクリプトが生成されます。 スクリプトは、おミ、データベースを変更していません。 これはおビプレビューです。

1	
2	The following SQL statements are only for preview and may not be executed in another SQL query too!
3	To execute these statements use function "Run SQL-script" from menu "Output".
4	Connect to database using the following connection-string:
5	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update if Insert
	Rest\altova.mdb;Provider=Microsoft.Jet.OLEDB.4.0
6	*/
7	
8	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Albert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [Title] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
9	
10	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander',
	[PhoneExt] = 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis',
	[PhoneExt] = 963, [Title] = 'Accounting Manager' WHERE ([Person].[PrimaryKey] = 3)
13	
14	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.Cicada@microtech.com', [First] = 'Camilla ', [Last] = 'Cicada',
	[PhoneExt] = 765, [Title] = 'HR' WHERE ([Person].[PrimaryKey] = 30)

データベース更新前のSQL スクリプト(部分的なビュー)

INSERT ステートメトがプレビューのスクリプト内で表示されていたいことに気がついているかもしれません。これはレコードは条件付きで挿入され、INSERT ステートメトは(マレングが実行される前は未知である「以下の条件で更新する」アクションの結果に依存するため、通常の振る舞いです。

メモ 特定のデータベース型のために、MapForceは、MERGEステートメントをUPDATEステートメントの代わりに作成します。 詳細に関しては次を参照してくたさい<u>MERGEステートメント</u>.

データベースコンオしてスクリフトを実行するココお以下を行います

• 「出力」メニューから「SQL-スクリプトの実行」をクトックします。

マピングが実行され、スクレプトがデータベースに適用されまた。「出力」タブ内でINSERTステートメイカ表示されていることに注意してくたさい。

1	/*
2	The following SQL statements were executed during "Generate output" function.
3	Every single result is written right to the ">>>" string.
4	These statements are only for preview and may not be executed in another SQL query tool
5	The database was connected using the following connection-string:
6	Data Source=\\viepfs06\Documentation\Public\ExampleFiles\EN\MapForce\DB Update if Insert
	Rest\altova.mdb:Provider=Microsoft.Jet.OLEDB.4.0
7	*/
8	
9	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'A.Aldrich@microtech.com', [First] = 'Aldert', [Last] = 'Aldrich',
	[PhoneExt] = 582, [Title] = 'Manager' WHERE ([Person].[PrimaryKey] = 1)
10	>>> OK. 1 row(s).
11	
12	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'b.bander@microtech.com', [First] = 'Bert', [Last] = 'Bander', [PhoneExt]
	= 471, [Title] = 'Accounts Receivable' WHERE ([Person].[PrimaryKey] = 2)
13	>>> OK. 1 row(s).
14	
15	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.clovis@microtech.com', [First] = 'Clive', [Last] = 'Clovis', [PhoneExt] =
	963, [Title] = 'Accounting Manager' WHERE ([Person].[PrimaryKey] = 3)
16	>>> OK. 1 row(s).
17	
18	UPDATE [Person] SET [ForeignKey] = 1, [EMail] = 'c.Cicada@microtech.com', [First] = 'Camilla ', [Last] = 'Cicada',
	[PhoneExt] = 765, [Title] = 'HR' WHERE ([Person].[PrimaryKey] = 30)
19	>>> OK. 0 row(s).
20	
21	INSERT INTO [Person] ([PrimaryKey], [ForeignKey], [EMail], [First], [Last], [PhoneExt], [Title]) VALUES (30, 1,
	'c.Cicada@microtech.com', 'Camilla ', 'Cicada', 765, 'HR')
22	>>> OK. 1 row(s).

データベース更新後のSQL スクリプト (部分的なビュー)

メモ MapForce から直接 SQL スクリプトを実行することは、データベースを更新する方法のひとつです。次も参照してくたさい、データベースを変更するマンピングの実行。

MapForce 内のDB ケエタブ内の「Person」テーブルを開くと(次を参照してくたさい、データベースの参照とケエル)、マンピングの結果は以下の通りです。

- 対応する主キーin XML ファイル内で、対応する主キーを持つすべてのデータベースの記録は更新されました。例は、主キー1、
 2、3、4、および5を持つレコードです。
- XML ファイル内に対応するキーをもたない全てのデータベースの記録は、影響を受けず、変更されていません。例は、主キー6、 7、8、および9を持つレコードです。
- 新規のレコードが(データベース内でキーか存在していない箇所である)「Person」テーブルは挿入されました。例は、主キー 30と31を持つレコードです。

	PrimaryKey •	ForeignKey •	EMail •	First •	Last •	PhoneExt •	Title •
1	1	1	A.Aldrich@microtech.com	Albert	Aldrich	582	Manager
2	2	1	b.bander@microtech.com	Bert	Bander	471	Accounts Receivable
3	3	1	c.clovis@microtech.com	Clive	Clovis	963	Accounting Manager
4	4	2	d.Durnell@microtech.com	Dave	Durnell	621	Marketing Manager Europe
5	5	2	e.ellas@microtech.com	Eve	Ellas	753	Art Director
6	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
7	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
8	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
9	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
10	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
11	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
12	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
13	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
14	14	4	n.newbury@microtech.com	Nira	Newbury	241	Support Engineer
15	15	4	o.origone@microtech.com	Olanda	Origone	345	Support Engineer
16	16	5	s.meier@microtech.com	Paul	Ponetti	114	Office Manager
17	17	5	t.bone@microtech.com	Quang	Qui	331	Accounts Receivable
18	18	6	m.nafta@microtech.com	Rita	Rotter	122	PR & Marketing Manager US
19	19	7	v.bass@microtech.com	Silla	Semua	716	IT Manager
20	20	7	c.franken@microtech.com	Tom	Tinker	147	Support Engineer
21	21	7	m.redgreen@microtech.com	Uri	Ugoner	152	Support Engineer
22	30	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
23	31	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

データベースを更新後の「Person」 テーブル

6.2.3.5 MERGE ステートメント

データベーステーブルを更新し、データを挿入する特定のマンピングは関しては、MapForceは、データベースに対して実行されるMERGE ステートメントをマンピングランタイムも生成する可能性があります。(次も参照してくたさい、「以下の条件で更新し、残りを挿入」アグショ ン。実行エンジンがMapForceである必要は必ずしもありません。次を参照してくたさい、データベースを変更するマンピングの実行。

MERGE ステートメート は 次のデータベース型によりサポートされています:

- SQL Server 2008 おび以降
- Oracle
- DB2
- Firebird

INSERT とUPDATE ステートメイを1つにませるけっか、MERGE ステートメイトは、データベースサーバーの呼び出しの回数を減らします。おこ、MERGE ステートメイの場合、整合性は、データベースによりチェックされます。データベース型を検出すると、MapForce によるMERGE ステートメイク生成を手動で影響することはできません。

マピングがラクイムニデータベースMERGE ステートメートを実行するかを確認します(INSERT とUPDATE ステートメートの組み合わせの適用に対して):

- 1. 「以下の条件で更新する...」と「残りを挿入」アウションを使用するマンピングを作成します。例えば、次を参照してくたさい、 「以下の条件で更新し、残りを挿入」アクション。
- 2. マピングをプレビューするコは「出力」ダブをクリックします。

MERGE がデータベース型によりサポートされている場合、生成されたSQL スクレプトは、MERGE ステートメートを含みます。例

```
1
        /*
        .
The following SQL statements are only for preview and may not be executed in another SQL query tool!
To execute these statements use function "Run SQL-script" from menu "Output".
 2
 з
        Connect to database using the following connection-string:
 4
 5
        Data Source=
                                                 ;Initial
        Catalog=DB MERGE;MultipleActiveResultSets=True;Password=•••••;Persist Security Info=True;User
        ID=
 6
        */
 7
        SET QUOTED IDENTIFIER ON
 8
 9
        UPDATE [dbo].[Users] SET [FirstName] = (CAST('Despine' AS nvarchar(50))), [LastName] =
10
        (CAST('Buttler' AS nvarchar(50))) WHERE ([dbo].[Users].[UserID] = 1)
11
12
        SELECT [UserID] FROM [dbo].[Users] WHERE ([UserID] = 1)
        -->>> %UserID1%
13
14
        DELETE FROM [dbo].[Addresses] WHERE EXISTS(SELECT * FROM [dbo].[Users] WHERE [dbo].[Users].[UserID] =
15
        [dbo].[Addresses].[UserID] AND ([dbo].[Users].[UserID] = '%UserID1%'))
16
        MERGE INTO [dbo].[Addresses] AS T USING ( VALUES ( '%UserID1%', 1, 1, (CAST('Home' AS nvarchar(20))),
17
        (CAST('Louisville' AS nvarchar(50))), (CAST('Elm Street' AS nvarchar(50))), (CAST('12' AS
        nvarchar(20))) ) AS S ( [UserID], [IsShipping], [IsBilling], [AddressType], [City], [Street],
        [Number] ) ON ( (S.[UserID] = T.[UserID]) ) WHEN MATCHED THEN UPDATE SET [IsShipping] = S.
[IsShipping], [IsBilling] = S.[IsBilling], [AddressType] = S.[AddressType], [City] = S.[City],
         [Street] = S.[Street], [Number] = S.[Number] WHEN NOT MATCHED THEN INSERT ( [UserID], [IsShipping],
         [IsBilling], [AddressType], [City], [Street], [Number] ) VALUES ( S.[UserID], S.[IsShipping], S.
         [IsBilling], S.[AddressType], S.[City], S.[Street], S.[Number] );
18
Mapping
            DB Query Output
```

データベース型がMERGE をサポートしない場合、生成されたSQL スクレプトは、UPDATE ステートメートのみを含みます。「以下の 条件で更新する...」条件が満たされている場合のみ、実行されるため、(マピングの実行前には未知である) INSERT ステートメート は、プレビューでは表示されません。

灹

- MapForce はサポートされるデータベースからを検知すると、MERGE ステートメートを自動的に作成します。MapForce が MERGE ステートメートを作成するか否かを手動で影響することはできません。
- 親子リレーションシップを持つ複数のテーブルを更新する場合、マージゴリーフ」テーブルのナックにのみ作成されます。「リーフ」テーブルはマップされている最も深い子テーブルです。例えば、下のマッピングでは、「以下の条件で更新する…」と「残りを挿入」アクションがUsers テーブルとAddresses テーブル両方のナックに定義されています。しかしなから、MERGE ステートメントはレーフテーブルAddresses のナックにのみ生成されます。親テーブルUsers は、MERGE の代わりにUPDATE まけはINSERT ステートメントを取得します。



• ステートメートに関しては、「バルク転送」オプション(次を参照してくたさい、<u>バルク転送(MapForce Server)</u>は、ODBC と JDBC データベース接続のためこのみによりサポートされています。

6.2.3.6 親テーブルを更新する際の子テーブルのためのオプション

マピングは「親」テーブルであるテーブルを更新する際、(つまし、テーブルに対して外部キー・ルーションシップ持っている場合)、ソースデー タとターゲルテーブル内で、どのようにな存するレコードか扱われるかを構成することができます。例えば、altova.mdb データベース内の 「Department」テーブルを更新するとします。すべての個人は、外部キーを使用して、departmentに、リンクされています。「Person」 テーブルに対しても、(挿入、更新、また」が開除などのアグションを取る必要があるかもしれません。これを行うことは、データベースの整合性を 保持し、マピングエラーを回避する助けとなります。

このトピックは親「Department」テーブルを更新した際に、「Person」テーブルに対して使用することのできるオプションについて説明しています。以下のサンプルファイルが使用されています。

- altova-cmpy-extra.xml データベース:挿入されるノースデーを含んでいます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲトデータベースです。

全てのファイルはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内にあります。下の説明では、 簡素化のためにこれらのファイルへの完全ない いは省略されています。

この例のマッピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同 じ結果を得ることができます。詳細に関しては、次を参照してくたさい、<u>データベースを変更するマンピングの実行</u>。

最初に、ソースXML スキーマとインスタンスとターゲナ・データベースをマッピングに追加します(次を参照: 例: "altova.mdb" データベースを マッピングに追加する)。「以下の条件で更新し、残いを挿入」アグションと同じステップを踏んでくたさい。次に、以下に表示されるようにマッ ピングの接続を描きます:



上記のように、マピングはターゲナデータベース内の「Department」テーブルを更新します。「Department」テーブルは「ルート」 テーブルとして選択されています。ルートテーブルは関する情報とその重要性に関する詳細は、次を参照してくたさい、データベースルーショ ンシップの処理。子「Person」テーブルは、プレントのアクションは、このトピックの主題です。

次のテーブルは、多種の構成オプションと対応するマッピングの結果を表しています。これらのオプションは、親「Department」テーブルと 子「Person」テーブルのデータベーステーブルアグションダイアログボックスから選択することができます。

構成 A

設定	マッピングの結果
Action on record 以下の条件で更新 「PrimaryKey 一致 「PoreignKey ・ IName ・ 子テーブルのデータ削除 ・ 入力子データを無視 ・ 「Person ・ Compartment J テーブル ・ Action on record 全て挿入 「PrimaryKey マップされた値 「ProreignKey 外部キー EMail マップされた値 「PhoneExt マップされた値 「Person」 ・ 「Person」 ・	 ソースXML内のPrimaryKeyがデータベーステーブル内の PrimaryKey 「対応する箇所で、Department レコードを 更新します。 入力 XML ファイル内に対応するレコードを持たない(このような キーはメース) 存在しません)既存のDepartment レコードを 更新しません。 データベース内で既存ではないすべてのPerson レコードを、入 力 XML インスタンスから挿入します。 「以下の条件で更新する」条件を満たす Department レコードの子デーダPerson レコード)を削除します。

構成 B

設定		マッピングの結果
Action on record 『PrimaryKey 『ForeignKey Name 子テーブルのデータ削除 入力子データを無視 『Person <i>「Department」 テー</i> :	以下の条件で更新 →致 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	SQL 実行エラーと共にマンピングは失敗します。理由は、マンピング が、新規のPerson レコードを既存のPerson レコードと同じ主キ ーを使用して挿入しようと試みオカらです。入力 XML からのレコ ードを、データベース内に既存のレコードに加えて追加する場合は、 次を参照してくたさい。
Action on record	全て挿入	
🕫 PrimaryKey	マップされた値 🗾	
💵 ForeignKey	外部キー	
💷 EMail	マップされた 値 📃 👤	
🔲 First	マップされた 値 📃 👤	
💷 Last	マップされた値 📃	
💷 Phone Ext	マップされた値 📃	
💷 Title	マップされた 値 📃 👤	
「Person」 テー ブル		

構成 C

設定		マパングの結果
Action on record 『PrimaryKey 『ForeignKey 『Name 子テーブルのデータ削除 入力子データを無視 『Person <i>TDepartment」テーラ</i> Action on record 『PrimaryKey	以下の条件で更新… →致 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	 ソースXML内のPrimaryKeyがデータベーステーブル内の PrimaryKey に対応する箇所で、Department レコードを 更新します。 入力 XML ファイル内に対応するレコードを持たない(このような キーはノースご存在しません)既存のDepartment レコードを 更新しません。 新規のPerson レコード(生成された主キーと共に)は、既存 のもの加え、Person テーブルに挿入されます。
Enterignkey	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
III First	マップされた 値 📃 👤	
💷 Last	マップされた 値 📃 💌	
💷 Phone Ext	マップされた 値 📃 💌	
🎹 Title	マップされた 値 📃 💌	
「Person」 テーブル		

構成 D

設定	マッピングの結果
Action on record 以下の条件で更新 『PrimaryKey 一致 『ForeignKey ▼ 『Name ▼ 子テ -ブルのデータ削除 ↓ 入力子データを無視 ▼ 『Person ▼ <i>ForeignKey</i> ▼ 『PrimaryKey マップされた値 『「PrimaryKey マップされた値 『「ForeignKey 外部キー 『「ForeignKey 外部キー 『「First マップされた値 『」 Last マップされた値 『」 Title マップされた値 『「 <i>Flerson」 テーブル</i>	 ソースXML内のPrimaryKeyがデータベーステーブル内の PrimaryKey に対応する箇所で、Department レコードを 更新します。 入力 XML ファイル内に対応するレコードを持たない(このような キーはよースご存在しません)既存のDepartment レコードを 更新しません。 オプション入力子データを無視するが親 Department テーブルのためご有効化されているため、Person テーブルコレコード は挿入されません。

構成 E

設定	マッピングの結果
Action on record 以下の条件で更新 『PrimaryKey 一致 『ForeignKey ・ 『Name ・ 子テーブルのデータ削除 ・ 八力子データを無視 ・ 『Person ・ <i>FDepartment J</i> テーブル Action on record 以下の条件で更新 『PrimaryKey 一致 『ForeignKey 外部キー 『EMail ・ 『First ・ 『Last ・ 『Title ・ <i>TPerson J F</i>	 ソースXML内のPrimaryKeyがデータベーステーブル内の PrimaryKey に対応する箇所で、Department レコードを 更新します。 入力 XML ファイル内に対応するレコードを持たない(このような キーはメースご存在しません)既存のDepartment レコードを 更新しません。 ソースXML内に対応する、PrimaryKeyをもつ Department にコレクされているレコードを全て削除します。理 由は、親 Department テーブルのよめに「子テーブル内のデ ータの削除」オプションカ特効化されているからです。 「以下の条件で更新する」条件を満たさなかす:部署にコレンだれているPerson レコードは、データベースご残されます。 Person テーブル内のレコードは更新されていません。

構成 F

設定	マパングの結果
Action on record 以下の条件で更新 『PrimaryKey 一致 『ForeignKey ・ 『Name ・ 子テ ーブルのデータ前除 ・ 入力子データを無視 ・ 『Person ・ <i>TDepartment J テーブル</i> ・ Action on record 以下の条件で無視 『PrimaryKey 一致 『ForeignKey 外部キー 『First ・ □Last ・ 『Title ・ <i>TPerson J テーブル</i>	 ソースXML内のPrimaryKeyがデータベーステーブル内の PrimaryKey に対応する箇所で、Departmentレコードを 更新します。 入力 XML ファイル内に対応するレコードを持たない(このような キー(おノース)ご存在しません)既存のDepartmentレコードを 更新しません。 ソースXML内に対応する、PrimaryKeyをもつ Department にした/されているレコードを全て削除します。この 理由は、「子テーブル内のデータの削除」オブションは、親 Department テーブルのオータの削除」オブションは、親 Department テーブルのオータの削除」オブションは、親 Department デーブルのオータの削除」オブションは、データベースに の条件で更新する」条件を満たざなかすこ Department にした/されているPersonレコードは、データベ ースに残されます。

構成 G

設定	マッピングの結果
Action on record 以下の条件で更新 PrimaryKey 一致 ForeignKey Image: Comparison of the second s	 ソースXML内のPrimaryKey がデータベーステーブル内の PrimaryKey 「文内応する箇所で、Department レコードを 更新します。 入力 XML ファイル内に文内応するレコードを持たない(このような キー(おノース)、存在しません)既存のDepartment レコードを 更新しません。 次の2つの条件を満たす、すべてのPerson レコードを削除し ます:
Action on record 以下の条件で無視 PrimaryKey 一致 ForeignKey 外部キー EMail First Last PhoneExt Title <i>The started for the started sta</i>	 a. 対応するソースXML内に対応するPrimaryKey が存在するDepartment ビングされているPersonレコード。 b. 対応するソースXML内に対応するPrimaryKey が存在するPersonレコード

6.2.3.7 「以下の条件で削除…」アクション

テーブルアクション「以下の条件で削除…」(Delete if..)は、データベーステーブルからデータを条件付きで削除する場合に使用されま す。データベーステーブルアクションダイアログボックスからこのアクションを定義することができます。(次を参照してください、データベーステーブ ルアクション設定)。例えば、ソースからのデータXMLをターゲホデータベースにマンピングする際、ソースXML内の特定の条件が、ター ゲホデータベースの記録(通常、主キーの値)内のフィールドに対して同じかをチェックするために、「以下の条件で削除…」条件を構成 することができます。「以下の条件で削除…」条件がtrueの場合、(つまり、2つのフィールドが等し、場合)データベースの記録は、マンピ ングが実行されると削除されます。

メモ 「以下の条件で削除…」テーブルアクションをデータベーステーブルアクションダイアログボックス内で使用することのできる「子テ ーブル内のデータの削除」オプションと混合しないように注意してくたさい。「以下の条件で削除…」テーブルアグションは、アク ションカ定義されているテーブルのみに影響を与え、他のテーブルコよ影響におりません。

この別は、データベーステーブルからデータを条件付きで削除する方法、お消除条件の結果をベースはして同じデータベーステーブルはコードを挿入する方法を説明します。

この例は以下のファイルを使用します

- altova-cmpy-extra.xml デーダベース:挿入されるノースデーダを含んで、ます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲホデータベースです。

全てのファイルはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内にあります。下の説明では、 簡素化のためにこれらのファイルへの完全ない いは省略されています。

この例のマッピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新

規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同じ結果を得ることができます。詳細に関しては、次を参照してくたさい、データベースを変更するマンピングの実行。

このマピングの目的は以下の通りです

- 同じ PrimaryKey を持つ、 person レコードが、ソースXML とターゲナ Person テーブル内両方に存在する場合、これら はPerson テーブルか削除されなければなりません。
- ソースXML からの全てのレコードは、Person テーブルは挿入される必要があります。

マンピングの目的を達成するために、以下のステップを行います。

ステップ 1: ソース XML コンポーネントを挿入する

• 「挿入」メニューから、「XML スキーマ/ファイル」をクトックして、Altova_Hierarchical.xsd を参照します。インスタンスファイ ルを与えるよう・問われると、altova-cmpy-extra.xml を参照してください。

ステップ2: ターゲナデータベースを挿入する

• 「挿入」メニューから「データベース」をクリックして、altova.mdb に接続するためにウィザードのステップを行ってくたさい。(次を参照してくたさい) 例: "altova.mdb" データベースをマッピングに追加する。

ステップ 3: 接続を描く

• 下に表示されるとおりマッピングの接続を描きます。



ステップ 4:「以下の条件で削除...」「残りを挿入」" アクションを構成する

- 1. ターゲオコポーネオで、「Person」テーブルの横のアクション. 挿入 (Alm)) ボタをクリックします。
- 2. 「レコードに対するアクション」の横の、「以下の条件で削除…」を選択します。これは、データベーステーブルアクションを条件 月の「アクションの削除」に変更します。つまり、条件が満たされる場合のみ、現在のレコード」は削除されます(次のステップを参照 してください)。
- 3. 「PrimaryKey」の横の、値「一致」を選択します。これにより、更新条件を定義します。つまり、下に表示されるとおり、データベースの記録は、「PrimaryKey」値がマンプングは与えられる「PrimaryKey」値に等し、場合、削除されます。

・各レコードに対して実行するアクション

以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て真であれば、指定されたアクションが実行されます。

入力データアクション	以下の条件で削除		アクションを追加(A)
🛛 🕅 PrimaryKey	一致		
🕅 ForeignKey		=	アクションを挿入(I)
🛄 EMail			
III First			
🔲 Last		Ŧ	
 ■ 		Þ	アクションを削除(D)

- 任意で、NULL equal チェックボックスを選択します。このサンプルでは、ソースXML内のPrimaryKey フィールド、おけまデ ータベース内のPrimaryKey フィールドに null 値が含まれていない ため、このチェックボックスは お関連性はおりません。 しかしな から、データ こ null 値が含まれており、このような値を等価として扱うはは、場合、このチェックボックスをチェックする必要があます。そ れ以外の場合は、希望しない結果が生成される場合があります。詳細に関しては、データベーステーブルアグション内の Null 値の 処理を参照してくたさい。
- 5. 「アケションの追加」をクリックします。これは、新規のアクションを既存の「以下の条件で削除…」アクションの右に追加します。 新規のアクションを「残りを挿入」として構成します:

- 各レコードに対して実行する 以下のオペレーションを使用 比較の結果が全て真であ	らアクション 用して全ての入力データは れば、指定されたアクション	DB デーブルのデータと比車 が実行されます。	<u> </u> 交されます。
入力データアクション	残りを挿入		アクションを追加(A)
💵 PrimaryKey	マップされた 値		
🔊 ForeignKey	マップされた 値	=	アクションを挿入(I)
💷 EMail	マップされた 値		
III First	マップされた 値		
💷 Last	マップされた 値	Ψ	
•		111 b	アクションを削除(D)

上のイメージでは、データベーステーブルアケションは、マパングの目的に従い構成されています。つまり、「以下の条件で削除 …」(Delete If...) 条件が満たされる場合のみ、レコード」消除されます。それ以外の場合、挿入されます。オプション「マップされた道」は、マパングからの値がレコードの全てのフィールドを作成するために使用されることを指定します。

6. ダイアログボックスを閉じるためは「OK」をクトックします。マンピングの背景で、アクション: 挿入 Ain ボタンが、アクション: 削除; 挿入(ADe.h) ボタン「変更されていること」注意してくたさい、これは、削除と挿入アクションが、このテーブルのためは構成されることを示します。 ステップ 5: マピングをプレビューし、データベースを更新する

マピングをプレビューするすよりゴ出力」タブをクリックします。データベースコネルで実行されるアクションを含む、 SQL スクリプトが生成されます。このスクリプトはプレビューのすようであり、データベースをおう変更していません。データベースコネルてスクリプトを実行するコお以下を行います。

- 「出力」メニューから「SQL-スクリプトの実行」をクリックします。
- メモ MapForce から直接 SQL スクレプトを実行することは、データベースを更新する方法のひとつです。次も参照してくたさい、データ ベースを変更するマンピングの実行。

MapForce 内のDB クエルダブ内の「Person」テーブルを開くと(次を参照してくたさい データベースの参照とクエル)、マッピングの結果は 以下の通りです:

- XML ファイル内に対応する主キーを持つすべてのデータベースの記録は、削除されました。例は、主キー1、2、3、4、および 5を持つレコードです。
- XML ファイル内に対応するキーをもたない全てのデータベースの記録は、影響を受けず、変更されていません。例は、主キー6、 7、8、9、10、11、12、および13を持つレコードです。
- 新規のレコードが(データベース内でキーか存在していない箇所である「Person」テーブルに挿入されまた。例は、主キー 30と31を持つレコードです。

	PrimaryKey •	ForeignKey •	EMail •	First 🔹	Last •	PhoneExt •	Title •
1	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
2	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
3	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
4	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
5	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
6	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
7	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
8	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
9	30	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
10	31	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

データベースを更新後の「Person」テーブル

6.2.3.8 「以下の条件で無視…」アクション

テーブルアクション「以下の条件で無視…」は、定義された条件に従い、データベース内の特定のレコードが更新されることを回避するため に使用されます。「以下の条件で無視…」アクションは、(「残りを挿入」アクションなどの)他のデータベーステーブルアクション組み合わ せて使用される場合のみ、意味があります。例えば、ソースXMLからのデータをターゲトデータベースコマッピングする場合、「以下の条件 で無視…」条件を構成し、ソースXML内の特定のフィールドがターゲトデータベースの記録内の通常、主キーの値である)フィールドと等 価がであるかをチェックするようご構成することができます。マンピングが実行される際、「以下の条件で無視…」条件がfrueの場合、(つ まり、2つのフィールドが等し、場合)、データベースの記録は無視され、次に定義されたアクション(例えば、「残りを挿入」)は、実行され ます。

この例は「以下の条件で無視…」条件の結果に基づ、サニデータベーステーブルコンコードを挿入する方法について説明します。この例は、次のファイルを使用します:

- altova-cmpy-extra.xml デーダベースご挿入されるノースデーダを含んでいます。
- Altova_Hierarchical.xsd 上記のインスタンスファイルを検証するために使用されるスキーマです。
- altova.mdb 更新されるターゲナデータベースです。

全てのファイルはくマイドキュメント>いtovaWapForce2021WapForceExamples¥フォルダー内にあます。下の説明では、 簡素化のためにこれらのファイルへの完全ない、なば略されています。

この例のマッピングは、サンプルデータベースファイルを変更します。元のデータベースをバックアップし、下記のステップを開始する前に、新 規のコピーを起動することが奨励されます。することが奨励されます。バックアップを作成することにより、元の例が上書きされず、下記と同じ結果を得ることができます。詳細に関しては、次を参照してくたさい、データベースを変更するマッピングの実行。

このマピングの目的は以下の通りです

- ソースXML とターゲナ Person テーブル内の両方で、同じ PrimaryKey を持つ個人のレコードが存在する場合、アクションは取られません。(つまり、無視されます)。
- 個人のレコードがPerson テーブル内に存在する上記の条件を満たされ場合、アクションは取られません。
- Person テーブル内に対応する(主キーが無し)ソースXML からのレコードは新規とて扱われ、新規の主キーと共に Person テーブルは挿入されます。

マノピングの目的を達成するために、以下のステップを行います。

ステップ 1: ソース XML コパーネントを挿入する

• 「挿入」メニューから、「XML スキーマ/ファイル」をクトックして、Altova_Hierarchical.xsd を参照します。 インスタンスファイ ルを与えるよう 出われると、 altova-cmpy-extra.xml を参照してください。

ステップ 2: ターゲナデータベースを挿入する

「挿入」メニューから、「データベース」をクリックして、altova.mdb (注接続するため)ことが、「のステップを行ってくたさい。(次を参照してくたさい、例 "altova.mdb"データベースをマソングに追加する。

ステップ 3: 接続を描く

• 下に表示されるとおりマピングの接続を描きます。



ステップ 4: 「以下の条件で無視..残しを挿入」"アクションを構成する

- 1. ターゲオコンポーネトで、「Person」テーブルの横のアクション: 挿入 (<u>Alu</u>) ボタンをクリックします。
- 2. 「レコードに対するアクション」の横の、「以下の条件で無視…」を選択します。これは、条件付きの無視アクションへのデータベ ーステーブルアクションを変更します。つまり、条件が満たされる場合のみ、現在のレコードは無視されます(次のステップを参照して ください)。
- 3. 「PrimaryKey」値の横から下に表示されるとおり、 equal を選択します。これは無視の条件を定義します。 つまり、 データベー スの記録は、 PrimaryKey 値がマンピングから与えられる PrimaryKey の値と同じ場合、無視されます。

各レコードに対して実行するア・ 以下のオペレーションを使用し 比較の結果が全て真であれば	ワション て 全ての 入力データは DB デー ず、指定されたアクションが実行	ブルのデータと比較されます. されます。	•	
Action on record	以下の条件で無視	•		アクションを追加(A)
🛛 💵 PrimaryKey	一致	_		
💵 ForeignKey		-		アクションを挿入(I)
💷 EMail		-		
First		-		
🔲 Last		-		
💷 Phone Ext		-		
III Title		•		
			~	アクションを削除(D)

- 任意で、NULL equal チェックボックスを選択します。このサンプルでは、ソースXML内のPrimaryKey フィールド、おけまデ ータベース内のPrimaryKey フィールドに null 値が含まれていないため、このチェックボックスは 想達性はおりません。 しかしな から、データ こ null 値が含まれており、このような値を等価として扱うはは、場合、このチェックボックスをチェックする必要があます。そ れ以外の場合は、希望しない結果が生成される場合があります。詳細に関しては、データベーステーブルアグション内の Null 値の 処理を参照してくたさい。
- 5. 「アクションの追加」をクリックします。これは、新規のアクションを既存の「以下の条件で無視…」アクションの右に追加します。 主キーがmax() + 1 に設定され、下に表示されるとおり、新規のアクションが「残りを挿入」として構成されます:

各レコードに対して実行するアク	各レコードに対して実行するアクション					
以下のオベレーションを使用して	こ全ての人力データは D	Bデーブルのデータと比較されます。				
比較の結果が全て真であれば	、指定されたアクションカ	「実行されます。				
Action on record	以下の条件で無視	▼残りを挿入		アクションを追加(A)		
🕫 PrimaryKey	一致	▼ max() + 1		/ // //		
🕫 ForeignKey		💌 マップされた 値		アクションを挿入(I)		
💷 EMail		💌 マップされた 値				
III First		💌 マップされた 値				
💷 Last		💌 マップされた 値				
Phone Ext		💌 マップされた 値				
🎹 Title		💌 マップされた 値				
			*			
•			P	アクションを削除(D)		

上のイメージでは、データベーステーブルアクションは、マピングの目的に従い構成されています。つまり、「以下の条件で無視…」条件が満たされると、レコードはスキップされます。それ以外の場合、挿入されます。オプション「マップされた値」は、マピングからの値がレコードの全てのフィーリドを作成するために使用されることを指定します。オプション max() + 1 は、レコードのために 一意の、新規の主キーの値を生成します。

6. ダイアログボックスを閉じるけっか」「OK」をクトックします。マッピングでは、アクション:挿入 Ain ボタイは アクション:無視;挿 入())ボタイで変更されます。これは、アクションの無視と挿入がこのテーブルのために構成されていることを示しています。

ステップ 5: マッピングをプレビューし、データベースを更新する

マピングをプレビューするオーダリゴ 出力」タブをクリックします。データベース マネル て実行されるアクションを含む、 SQL スクリプト か生成されます。 このスクリプト はプレビューのオーダであり、データベースをおう変更していません。 データベースマオル てスクリプト を実行する ココンド下を行います。

- 「出力」メニューから、「SQL-スクリプトの実行」をクリックします。
- メモ MapForce から直接 SQL スクレプトを実行することは、データベースを更新する方法のひとつです。次も参照してくたさい、データベースを変更するマレングの実行。

MapForce 内のDB ケエタブ内の Person」テーブルを開くと(次を参照してくたさい、データベースの参照とケエル)、マンピングの結果は以下の通りです:

- XML ファイル内に対応する主キーを持つ全てのデータベースの記録は「以下の条件で無視…」条件を満たし、影響を受けず、変更されていません。例は、主キー1、2、3、4、および5を持つレコードです。
- XML ファイル内で対応するキーを持たない全てのデータベースの記録は、「以下の条件で無視…」条件を満たしませんでしたが、影響を受けず変更されていません。例は、主キー6、7、8、9、10、11、12、および13を持つレコードです。
- 新規のレコードが(データベース内でキーが存在していない箇所である)「Person」テーブルは挿入されました。例は 主キー 30と31をソースXML ファイル内に持つレコードです。これらは それぞれ新規の主キー22と23と共にデータベースは挿入されます。

	PrimaryKey •	ForeignKey •	EMail •	First •	Last •	PhoneExt •	Title •
1	1	1	v.callaby@nanonull.com	Vernon	Callaby	582	Office Manager
2	2	1	f.further@nanonull.com	Frank	Further	471	Accounts Receivable
3	3	1	I.matise@nanonull.com	Loby	Matise	963	Accounting Manager
4	4	2	j.firstbread@nanonull.com	Joe	Firstbread	621	Marketing Manager Europe
5	5	2	s.sanna@nanonull.com	Susi	Sanna	753	Art Director
6	6	3	f.landis@nanonull.com	Fred	Landis	951	Program Manager
7	7	3	m.landis@nanonull.com	Michelle	Butler	654	Software Engineer
8	8	3	t.little@nanonull.com	Ted	Little	852	Software Engineer
9	9	3	a.way@nanonull.com	Ann	Way	951	Technical Writer
10	10	3	l.gardner@nanonull.com	Liz	Gardner	753	Software Engineer
11	11	3	p.smith@nanonull.com	Paul	Smith	334	Software Engineer
12	12	4	a.martin@nanonull.com	Alex	Martin	778	IT Manager
13	13	4	g.hammer@nanonull.com	George	Hammer	223	Web Developer
14	14	4	j.band@nanonull.com	Jessica	Bander	241	Support Engineer
15	15	4	l.king@nanonull.com	Lui	King	345	Support Engineer
16	16	5	s.meier@nanonull.com	Steve	Meier	114	Office Manager
17	17	5	t.bone@nanonull.com	Theo	Bone	331	Accounts Receivable
18	18	6	m.nafta@nanonull.com	Max	Nafta	122	PR & Marketing Manager US
19	19	7	v.bass@nanonull.com	Valentin	Bass	716	IT Manager
20	20	7	c.franken@nanonull.com	Carl	Franken	147	Support Engineer
21	21	7	m.redgreen@nanonull.com	Mark	Redgreen	152	Support Engineer
22	22	1	c.Cicada@microtech.com	Camilla	Cicada	765	HR
23	23	1	c.corrigan@microtech.com	Carol	Corrigan	629	Admin

データベースの更新後の「Person」テーブル

6.2.3.9 トランザクションロールバックの使用

データベースを変更するマピングを実行する場合、MapForce(おけよMapForce Server、おけよMapForceーにより生成されたプロ グラム)ではデータベースは関連したエラーが発生する場合があります。例えば、データベースアカウナトには、特定のデータベースアグションを実 行するけれの特権がない、おけは無効なデータを入力、おけは更新することを回避するデータベース制約を実行する特権がない場合がありま す。後者は、列か必須で、マピングが価を適用しない場合に発生する場合があります。 データベース例外が発生した場合にコールレジクを可能にするコよ、マピングがデータベーストランザクションを使用するように構成します。下て説明されるとおり、データベースコンポーネントレベルとテーブルアクションレベルてトランザクションを有効化することができます。

データベースコンポーネントレベルでのトランザクション

データベースコンポーネントレベルでトランザクションを有効化するコよ、以下を行います

1. データベースエンポーネントを右クトックし、コンテキストメニューカンジプロノティ」を選択します(代わりに、データベースエンポーネントのタイトルレチをダブルクトックします)。これによりデータベースエンポーネント設定ダイアログボックスか開かれます。



2. 「トランザクションの使用」チェックボックスを選択します。

これを行うことにより、データベースエンポーネントにより行われるすべての変更がエラーを発生した場合ロールレックされるトランザクション内に囲まれます。マッピングの実行中にエラーが発生すると、出力はドロップダウンノストから選択されたオプションにより異なります。

- トップトランザクションをロールバックし停止 データベースコンポーネントにより行われるすべての変更を囲むトランザクションはコールレッシンされ、マンピングは停止されます。
- トップトランザクションをロールバックし継続 上記同様ですが、マピングはロールバック後も継続されます。(例えば、存在する場合、2番目のターゲトコンポーネントを処理する場合など)。

マピングの一部は複数のデータベースコンポーネントを含んでいる可能性があります。これらは、同じデータベース接続を使用する、おけますっ ナメ異なるデータベースへの接続を使用している場合があります。このようなマピングの出力は、データベースエラーが発生する場合、実行エ ンジントことにとないます:

マンピングがMapForce で実行される場合マンピングの実行時、1つのターゲトコンポーネトか実行されます。「プレビュー」
 ボタンが有効化されている個所のコンポーネントです。そのコンポーネント内でデータベースエラーが発生し、「トランザクションの使用」チェックボックスが有効化されていると、コンポーネントによる全ての変更がローフレジンクされます。この場合、データベースエラーか発生すると、エラーか発生する箇所でローフレジンが発生し、マンピングは停止されるか、エラーが発生する箇所でドロップダウノリストで選択されたとおり、次のターゲトコンポーネントの実行を継続します。

データベースコンポーネント設定ダイアログボックスで使用することのできるすべての設定に関数の参照に関しては、データベースコンポーネント 設定を参照してくたさい。

テーブルアクションレベルでのトランザクション

トランザンタン at テーブルアンタン レベルでのトランザンタンは、個別の 挿入、更新などのテーブルアンタンを含みます。このセトアップ では、レコード上で挿入、おけま更新アクションが不足する場合、現在失敗したアクションのみをロール シグレ、他のレコードの処理を続行す ることができます。テーブルアンタン レベルでトランザンタンを実行する コよ、データベーステーブルアクション ダイアログボックス「トランザク ションの使用」 チェックボックスから選択します。(次を参照:データベーステーブルアグション設定).

😢 データベーステーブルアクション - products	×
最初のレコードの前に実行する SQL ステートメント ● なし(N) ○ 全てのレコードを削除(L) ○ 全ての子テーブルにある全てのレコードも削除 ○ カスタム SQL(C):	
各レコードに対して実行するアクション ここで定義された演算子を使用して、すべての入力データはDBテーブルデータと比較されます。 NULL 値を等価として扱うように「NULL equal」チェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。	
レコードに対するアクション NULL 等価 全て插入 ▼ 10 name マップされた値 ▼ 11 quantity マップされた値 ▼ 12 description マップされた値 ▼ 11 created_at マップされた値 ▼ 11 updated_at マップされた値 ▼	(A) (I)
子テーブルのデータ削除 ・ ・ アクションを削除 入力子データを無視 ・ ・ アクションを削除	(D)
 データベーストランザクションを使用(T) エラーが発生した場合: トップトランザクションをロールバックし停止します トレース トップトランザクションをロールバックし停止します リップトランザクションをロールバックし停止します 現在のトランザクションをロールバックし停止します 現在のトランザクションをロールバックし停止します リースレベル(E): コンパー・イントロメンをロールバックし継続します バルク転送を使用 (MapForce Server のみ)(B) バッチサイズ: 1000 レコード バルク転送はデータベース接続によりサポートされていません。 	
 OK 年ャン	セル

使用することのできるオプションは以下の通りです

・トップトランザウションをロールバックし停止 – 親子リレーションシップを持つマピングが複数のテーブルを更新する場合、親テーブルのオメガヨナではよく、子テーブルのオメリコトランザションを有効化することができます。トランザションが複数のヘベルで有効化することができます。トランザションが複数のヘベルで有効化することができるよう、「トップ」は階層で更に高いレベルを参照することができます。これは、そのヘベルでのトランザションの有効化されている場合、データベースエンポーネントレベルからのトランザションです。それ以外の場合、エンポーネントしば複数のホストされたテーブルアグションが存在することが仮定され、一番上のテーブルアグションで定義されたトランザションが取られます。データを複数にコンクされているテーブルは挿入するで説明されるとおり、このオプションを理解するこは、マネントホーネントの物であることを思いすう、てくたさい、例えば、エンポーネントには、個人のレコードを挿入する親「挿入」アグション、および、個別のテーブルに各個人のレコードを挿入する1つまがは複数のホストされた「挿入」アグションの存在します。両方の挿入アグションのすがコトランザブションを有効化することができ、「トップ」は階層レベルの最も高いレベルを指しています(このサンブルでは、個人がアグションを挿入しています)。このナッグ、アドレスを挿入する際に初外が発生すると、これはこのサンプルでは、個人の、アジョンのためになっていたのようなアドレスを挿入する際に初外が発生すると、これはこのサンプルでは、個人の、マーク、マーク、マンク、マンク、マンク、マンク、アドレスを挿入する際に初外がないた。

テーブルレベルであるトップのトランザクションをロールバッグし、実行は停止されます。すなわち、個人のアドレスを挿入できない場合、個人のレコードは挿入されません。

- トップトランザクションをロールバックし継続する 上記同様ですが、ロールバックの後もマンピングは継続されます(例えば、存在する場合、他のターゲットコンポーネントを処理するために継続されます)。
- 現在のトランザクションをロールレックし停止する 例外が発生すると、このロールレックは、現在のトランザクション内で発生した変更のみをロールシクします。現在のトランザクション外で発生した変更はコミットされます。
- 現在のトランザクションをロールバックし継続する 上記同様ですが、マピングはロールバック後も継続されます。

ストアドプロシージャレベルトランザクション

ストアドプロシージャのオメのデータターゲットと呼ばれるデータベーストランザグランを有効化することができます。例えば、<u>ターゲットコンポーネント内のストアドプロシージャ</u>を参照してくたさい。(入カパラメーターを持たない)データノースとして呼び出される、ませょば関数のようコ呼び出される(入力と出力を持つ)ストアドプロシージャのオメリコよトランザグションイオサポートされていません。

ロールック

トランザションがマピングランタイムご発生するかを確認するコよ「出力」ダブをクリックして、MapForce にお生成された疑似-SQL codeコードを確認します。下のイメージは、コンポーネトレンバレテーブルアションベルクトランザションが有効化されているマピングのプ レビューを示しています。下に示されているとおり、メインのトランザションは各オペレーションのためご発生します(コメント --begin transaction と--commit transaction にお意別されます)。また、複数の小さ、トランザションは各入力オペレーションのた めご発生します(SAVEPOINT コマンドにお意別されます)。

7	
8	begin transaction
9	
10	PRAGMA foreign_keys = ON;
11	
12	SAVEPOINT PRODUCTS
13	
14	INSERT INTO "products" ("name") VALUES ('Product A')
15	
16	SAVEPOINT PRODUCTS
17	
18	INSERT INTO "products" ("name") VALUES ('Product B')
19	
20	SAVEPOINT PRODUCTS
21	
22	INSERT INTO "products" ("name") VALUES ('Product C')
23	
24	commit transaction
25	
マッピング	ブ DB クエリ H力

マピングがMapForce Server おけよMapForce により生成されたプログラムにより実行されると、上記のとおり、マピングのデザイン中 に構成されたオプションに基づきロールバックが実行されます。例えば、オプション「トップトランザクションをロールバックし停止」を選択する と、他の挿入が対功した場合でも、失敗した入力オペレーションにより全ての変更がロールバックされます。しかしながら、「現在のトランザク ションをロールバックし継続する」を選択すると、失敗した挿入か最も近い保存ポイントにコールバックされ、マピングは他のレコードを挿入 しようと試みます。

メニューコマボ「出力 | Run SQL-スクリプトの実行」を使用して MapForce 内でトランザブションが有効化されたマッピングを直接実行すると、ダイアログボックスはデータベースの別外が発生したかを通知します。例:

😢 MapForce: データベース トラ	ンザクション例外	×
トランザクション内のエラー:	PRODUCTS	
エラーの原因:		
NOT NULL constraint failed: pr	oducts.quantity	<u>.</u>
SOL275-63276:		
INCEDT INTO l'acaduata!! (l'acad	-1) //// //// /// //////////////////////	
どのように続けるか選択		
○このトランザクションをロール	バックして停止	
○このトランザクションをロール	バックして続ける	
○ 今後のエラー全てに対	してロールバックして続ける	
● トップをロールバックして停」	Ł	
○トップをロールバックして継続	売	
	ОК	

以降の振る舞いを変更することができます

- このトランザクションをロールバックし停止する これは、現在のトランザクションに含まれる変更のみをロールバックし、マピングの 実行を停止します。現在のトランザクションの外部で前に行われた変更はエミナトされます。
- このトランザクションをロールバックし継続する 上記同様ですが、マパングロールバック後も実行されます。チェックボックスロールバックして全てのエラーのために継続するを選択して、更にエラーが発生した場合、ダイアログボックスを抑制します。
- すべてをロール、シックし停止する-このデータベース呼び出し内のすべてのトランザクションは、ロール、シックンクの実行 は停止されます。

6.2.3.10 バルク挿入 (MapForce Server)

「バルク転送」コマンドによりMapForceコンポーネント(TXT、CSV、DATなどからデータベーステーブルに高速にデータを挿入することができます。このオプションを使用することにより、1つのステートメントを実行することにより、挿入プロセスを劇的にスピードアップすることができます。

「バリク転送を使用」オプランはマンピングのデザイン時に下に示されるとおり、MapForce内で有効化することができます。このオプションが有効化することのできるマンピングは、MapForce内で実行することができまが、バリン挿入はこの時点では適用することはできません。 データの実際の、シレク転送はMapForce Serverによりマンピングが実行された時に起こります。

パルク転送は以下の条件が満たされた場合にサポートされます

• マピング変換言語がBUILT-IN に設定されている場合。詳細に関しては、次を参照してください変換言語の選択。

- マピングがスタイアロン、おはFlowForce Server の管理下で MapForce Server にお作動している場合。これ は、マピングが.mfx フォーマナにエン ペルされるか、おけまFlowForce Server にデプロイされる必要かあることを意味しま す。詳細に関しては、次を参照してくたさ、<u>MapForce Server 実行可能ファイルにマピングをエン ペルする</u>と<u>FlowForce</u> <u>Server にマピングをデプロイする</u>。
- MapForce Server ライセンスは、マルチコアマシン上の「単一スレンドの実行」だけに制限されません。つまり、チェックボックス <u>Altova LicenseServer</u>の「サーバー管理」ダブ内の「単一スレンドの実行に制限する」が無効化されている必要があります。
- データベースアケンコンは「全て挿入する」です。次も参照してくたさい、データベーステーブルアケンコン設定。
- データが シレクトコードされるテーブルは、 ベリーフテーブル である必要があります。 すなわち、 データベースの一番下の階層である必要があります。 関連したテーブル ビュー、 おけはマンピング内のテーブルを参照するストアドプロシージャ は存在してけないません。
- データベースドライバーはWHERE 条件でのバルク挿入をサポートします。

ADO ODBC **JDBC** ADO.NET ネイティブ Access いえ いえ n/a n/a n/a はい DB2 いえ はい はい n/a Firebird はい はい いえ n/a n/a いえ はい はい はい Informix n/a **iSeries** いえ はい はい はい n/a はい MariaDB いえ はい はい n/a はょ はい はい **MvSQL** n/a n/a Oracle いえ はい はい はい n/a はい **PostgreSQL** n/a はい n/a はい Progress はい はい n/a n/a n/a **SQL Server** はい はい はい はい n/a いえ SQLite n/a n/a n/a n/a Sybase いえ はい はい n/a n/a Teradata n/a はい はい n/a n/a

次のテーブルは使用されるデータベースの種類とデライバーにとい異なるいが分挿入へのサポートの概要を示しています。

*MySQL バージョン5 または以降が必要とされます。

この例は、サンプルsource.txt ファイルからデーダをターゲトデータベースロッシクロードするマピングを作成する方法について説明しています。例は、SQL Server 2014 とAdventureW orks 2014 データベースを使用しています。後者は、CodePlex W eb サイトからダウンロードすることができます(https://msftdbprodsamples.codeplex.com/)。

```
Location A,15.3,39
Location B,46,34
Location C,56.33,0
Location D,0,399
Location E,0,97.43
```

source.txt

バルク転送の定義

- 1. 内蔵のターゲナ出力アイコン 間間 か有効化されていることを確認してくたさい。
- 2. AdventureW orks 2014 データベース 接続し、 "Production.Location" テーブルをマピングに追加します。詳細に関しては、次を参照してくたさい、データベースをマピングに追加する。

O AdventureWorks2014	
Production.Location Location	I
> In LocationID smallint NOT NULL	I
> III Name nvarchar(50) NOT NULL	I
> DCostRate smallmoney NOT NULL	I
> Motinal(8,2) NOT NULL	I
Image: Image:	I

1. 「挿入」メニューカら、「テキスト」をクトックして、(上に表示されている source.txt サンプル等のソーステキストファイル をマッ ピングに追加します。詳細に関しては、次を参照してください。CSV とテキストファイル。ソースとターゲトコンポーネントのデータ型

と互換性があることを確認してくたさい。「データ型の表示」(「ジールボタンが有効化されている場合、データ型はコンポーネト上で表示されます。

I source		🖯 AdventureWorks2014	
日ロファイル: source.txt ファイル/文字列 string[}	P IProduction.Location Location	Ŗ
¦> ⁱ . <mark>⊖ () Rows</mark> [0∞]	ז <	👌 🔤 🖫 📭 🖓 🗠 🖓 🔤 🖓 🐂 🖓	¢.
🗧 💳 = Field1 string [01] 🛛 🖓	ז <	> • Name nvarchar(50) NOT NULL	¢.
🗧 💳 = Field2 decimal [01] 🛛 🛛 🛛	<u>ነ</u> ל	CostRate smallmoney NOT NULL	Ę.
🗧 🛄 🗕 Field3 decimal [01]	<u>ነ</u> ל	🗧 🔤 🖾 🗛 🖓 🗛 🗛 🗛 🗛 🗛 🗛 🗛	Ę.
	C	Image: ModifiedDate datetime NOT NULL	- k

2. 下に表示されるとおりマッピングの接続を描きます。データベースアクションボタン Alm は テーブル名の右に表示されていることに 注意してくたさい。

I source		🖯 AdventureWorks2014	
日ロファイル: source.txt ファイル/文字列 string[} _⊣	👎 🛄 Production. Location 🔼 In Location	Ę
[¹ .□ () Rows [0∞] I	ו '-	IT LocationID smallint NOT NULL	þ
Field1 string [01]	▶ - I	Mame nvarchar(50) NOT NULL	Ļ
🖓 🔤 🗕 🖛 🗕 🖓 🖓 🖓 🖓	▶ - I	CostRate smallmoney NOT NULL	þ
🖞 🛄 🗕 Field3 decimal [01]	▶ - I	MOT NULL	þ
	1	Imm ModifiedDate datetime NOT NULL	k

3. データベースアクション ボタン(All)をクリックして、ベバリレ東送の使用 チェックボックスを有効化します。「OK」をクリックして 確認します。「バリノケ転送の使用」チェックボックスを選択して、「OK」をクリックして確認します。

🕑 データベーステーブルのアクション - Production.Location	×
最初のレコードの前に実行する SQL ステートメント ◎ なし(N) ◎ 全てのレコードを削除(L) □ 全ての子テーブルにある全てのレコードも削除 ◎ カスタム SQL(C) :	
各レコードに対して実行するアクション 以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て真であれば、指定されたアクションが実行されます。	
Action on record 全て挿入 ▼ □□ Lo cationID DB 生成 ▼ □□ Name マップされた値 ▼ □□ CostRate マップされた値 ▼ □□ Availability マップされた値 ▼ □□ ModifiedDate マップされた値 ▼	アクションを追加(A) アクションを挿入(I)
□トランザクションを使用(T) ■ バルク転送を使用 (MapForce Server のみ)(B) バッチ サイズ: 1000 レコード ([トランザクションの使用]は[バルク転送の使用]を 非アクティブ化した後選択できます。)	, , , , , , , , , , , , , , , , , , ,
ОК	キャンセル

上のダイアログボックス内で、「レコードに対するアクション」が「全て挿入」に設定されていることを確認してください。「バッチサイズ」フィールドは、各アクションに挿入されるレコードの数量を定義します。

メモ 「バルク転送の使用」オプションが有効化されていると、「トランザウションの使用」オプションは無効化されます。また、「トラン ザウションの使用」オプションが有効化されていると、「バルク転送の使用」オプションは無効化されます。トランザウション処理を 有効化するコよ、「バルク転送の使用」チェックボックスのチェックを解除します。

バルク挿入か有効化された次のステップは、MapForce Server内でのマッピングの実行です、スタイアロン、おけまFlowForce Server の管理下で)。詳細に関しては、次を参照してくたさい、MapForce Server 実行可能ファイルにマッピングをコンパイルすると FlowForce Server にマッピングをデプロイする。

6.2.3.11 データベーステーブルアクション内の Null 値の処理

「Ignore If」、「Update If」、「Delete If」などのテーブルアケションを使用してマンピングがターゲトデータベースを更新する場合、 MapForce はソースデータをターゲットデーター、対して比較し、結果として内部データベース更新クロを生成します。(これらの内部クロノ は、MapForceの出力ペイン内のプレビューのために使用することができます次を参照してくたさい、データベースを変更するマンピングの実 行)。生成されたクロルは、「データベーステーブルアケション」ダイアログボックスから定義された比較条件を反映しています。

最初のレコードの前に実行する SQL ステートメント ● なし(N) 全てのレコードを削除(L) 全ての子テーブルにある全てのレコードも削除 ● カスタム SQL(C): 客レコードに対して実行するアクション ここで定義された演算子を使用して、すべての入力データは DB テーブルデータと比較されます。 NULL 値を等価をして扱うように TNULL equal チェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。 「図 id ・ 「図 id ・ 「回 istname - 一致 ・ マップされた値 マップされた値	😢 データベーステーブルのアクシ	ヨン - dbo.TAR	GET				>
各レコードに対して実行するアクション ここで定義された演算子を使用して、すべての入力データは DB テーブルデータと比較されます。 NULL 値を等価として扱うように 「NULL equal Jチェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。 レコードに対するアクション NULL 6を等価として扱うように 「NULL equal Jチェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。 アクションが実行されます。 アクションを追加(A) 10 firstname 一数 マップされた値 回 email 一数	 最初のレコードの前に実行す ● なし(N) ○ 全てのレコードを削除(L) ○ カスタム SQL(C): 	る sqL ステート):	メント 全ての子テーブルにある全て	のレコードも削	除		
レコードに対するアクション NULL 等価 以下の条件で無視 ▼ 残りを挿入 アクションを追加(A) □□ id □ DB 生成 □ Tristname □ マップされた値 □□ lastname □ 一致 マップされた値 □□ email □ 一致 マップされた値	各レコードに対して実行する) ここで定義された演算子をf NULL 値を等価として扱うよ 全ての比較が true の場合。	Pクション 使用して、すべて うに「NULL equa 、特定のアクショ	の入力データは DB テーブ/ JUチェックボックスをチェックし ンが実行されます。	レデータと比較 てください。	されます。		
□ □	レコードに対するアクション	NULL 等価」	以下の条件で無視	•	残りを挿入	<u> </u>	アクションを追加(A)
□ lastname □ 一致 マップされた値 □ email □ 一致 マップされた値	10 firstname			<u> </u>	DB 生成 マップされた値	_	アクションを挿入(1)
□□ email □ 一致				.	マップされた値		/// L/ L// (L/
	🔲 email	· · ·	一致	•	マップされた値		
マカションを買用金の						v	アカミュン,た省(昭全/m)

データベーステーブルアクション ダイアログボックス

ソースおけおテーゲオデータにnullable フィーリドが含まれている場合、ソース内のnull 値をターゲオデータ内の値と比較する方法がつ 存在します:

- 1. null 値を等価とて扱う
- 2. null 値を等価とて扱わない

デフォルトでは、NULL-対応は、無効化されています。条件が満たされており、NULL-対応が無効化されている場合、ターゲルデータベーステーブルが期待通り更新されないインスタンスが存在する場合があります(例えば、必要以上の行か挿入、おけは更新されます)。NULL 値 かデータの比較を影響し、希望してい結果を生成するナタリニエのような状況が発生します。このような状況を回避するナタリニ、「データベー ステーブルアグション」ダイアログボックスから各 Null 許容 フィーリド(下のイメージ内 email)の横のチェックボックスを選択します。

マピング内のNULL対応を良く理解するために、以下の条件がtrueの場合、NULLデータの比較が起こるサンプルを分析してみましょう

- 1. 「データベーステーブルアケンコン」ダイアログボックスは、「Ignore if」、「Update if」、「Delete if」アクションを含んています。
- 2. これらのアクションは NULL 値を含んている可能性のあるレコードに対して行われます。
- 3. ソーステーブル内のNULL 値は ターゲオテーブル内のNULL 値の等価として扱われる必要がおます。

デフォルトでは、NULL-対応は、無効化されています。条件が満たされており、NULL-対応が無効化されている場合、ターゲルデータベー ステーブルが期待通り更新されないインスタンスが存在する場合があります(例えば、必要以上の行か挿入、まけは更新されます)。NULL 値がデータの比較を影響し、希望しない結果を生成するためにこのような状況が発生します。このような状況を回避するために、「データベー ステーブルアゲンタン」ダイアログボックスから各 Null 許容フィールド(下のイメージ内email)の横のチェックボックスを選択します。Null 許 容であるフィールドのためにのみ、また、少なくとも1つのテーブルアゲンコンが「等価」まけけば、等価(大文字と小文字を区別しない)」場合のみ、 チェックボックスを選択することができます。

サンプル

マピング内のNULL対応を良く理解するために、NULLデータの比較が起こるサンプルを分析してみましょう。このサンプルは、 Microsoft SQL Server データベースを使用します。しかしながら、サポートされる他のデータベースの型にも適用することができます。任意 で、Microsoft SQL Server を使用する場合、次のデータベーススクリプトを実行してこのサンプル内で使用されているテーブルとデータを作 成することもできます: くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\CreateNullableFields.sql.

簡素化のために、データベーステーブルは以下のように説明されています。両方のテーブルは、個人のデーダを保管しており、同じ列が存在しています。 更に、列emailは、両方のテーブル内のnullデーダを含むことができます。

SOURCE テーブル

++	irstname	lastname	email
1 M:	lia	Dahill	NULL
2 F1	'red	Weinstein	f.weinstein@nanonull.com

TARGET テーブル

SOURCE テーブルかのデータをTARGET テーブルニマージすることが目的と仮定します。新規のレコードのみがTARGET テーブルご挿入さ れます(このサンプルでは、「Tobie Hughey」)。両方のテーブルご存在するレコード(「Mia Dahill」と「Fred W einstein」)は、無視 されます。

タスクは以下のように達成することができます。

- 1. 「挿入」メニューから「データベース」を選択します。ウィザードステップに従い、データベースは接続します。(次を参照: <u>データベ</u> <u>ースは接続</u>)。データベースオブジェクトを追加するようにプロンプトされると、テーブル SOURCE を選択します。
- 2. 「挿入」メニューから「データベース」を選択します。データベースニ再接続し、テーブルTARGET をマンピングに追加します。
- 3. ソースとターゲトコンポーネント間のマンピング接続を描きます。

D NANONULL [NANONULL]		DI NANONULL [NANONULL]	
이 대 dbo.SOURCE	I	Dim dbo.TARGET A:In	l 1
(> []¶id		• 🕅 id	Į 1
👌 📖 firstname 🛛 🛛		🕨 🔤 firstname	l 1
👌 📖 lastname 🛛 🛛	I	🕨 🔲 lastname	Į 1
👌 📖 email 👘 🛛		🕨 🔲 email	Į 1

1. Action:Insert All ボタンをクリックして、以下のようにデータベーステーブルアグランを構成します:

各レコードに対して実行するアクション

ここで定義された演算子を使用して、すべての入力データは DB テーブルデータと比較されます。 NULL 値を等価として扱うように「NULL equal」チェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。

レコードに対するアクション	NULL 等価	以下の条件で無視	•	残りを插入	\wedge
🛛 🕄 id			•	DB 生成	
firstname		一致	T	マップされた値	
I lastname		一致	•	マップされた値	
🔲 email		一致	T	マップされた値	

上記のとおり、「Ignore if.. Insert Rest」アクションの組み合わせか定義されます。この構成は、各レコードに対してマソピングが以下を チェックします:

- ソース内の firstname は ターゲナ内の firstname と等し です。 そして、
- ソース内のlastname はターゲオ内のlastname と等しいです。 そして
- ソース内のemail は ターゲナ内のemail と等しいです。

上のすべての条件が満たされる場合、(条件に応じて)レコードは無視されます。それ以外の場合、新規のレコードは、ターゲトテーブルに 挿入されます。新規のレコードのidは、データベースにより生成されますが、他のフィーリド(firstname、lastname、email)は、ソー スからマップされた値を使用して表示されます。

emailの横のチェックボックスは、このフィーリドのナックのNULL-対応の比較を有効化、おけは無効化することができます。emailはNullの値を含むことができるすっな、このチェックボックスを選択する必要があります。(すなオち「Mia Dahill」 はNULL 電子メーリフドレスが存在するからです)。チェックボックスの没割を確認するはは、データベースを2度更新してくたさい、最初はチェックボックスを選択し更新し、2度目はチェックボックスのチェックが解除された状態で更新してくたさい。

データベースを更新するコよ出カダをクリックして、メニューコマンド「出カ|SQL-スクリプトの実行」を実行します。

チェックボックスが選択されている場合、MapForceはNULLフィールドを等価として扱うように明示的に指定されています。ですから、レコード「Mia Dahill」はターゲットテーブルに挿入されせん。これは期待されている結果です。

チェックボックスが選択されていたい場合、希望する結果ではないレコード「Mia Dahill」がこのレコードは既存にもかかわらず)ターゲルテ ーブルは挿入されます。MapForce にNULL 値を等価として扱うように明示的な指定が与えられていたいからです。データベースに次のク コを実行すると同様のシチュエーションが発生する可能性があります(NULL 値 が「=」演算子と比較され、NULL 対応ではないすっかこ のクエリは、レコードを抽出しません)。

SELECT firstname, lastname, email FROM TARGET WHERE firstname = 'Mia' AND lastname =
'Dahill' AND email = NULL;

NULL に対応するには、上のケエル以下のように書かれる必要があります。

```
SELECT firstname, lastname, email FROM TARGET WHERE firstname = 'Mia' AND lastname =
'Dahill' AND email IS NULL;
```

メモ 上のクロルは、説明のナックで、MapForceにより生成される内部クロルの実際の構文を反映していません。(異なるデータベース ベンダーはNULL 比較を異なるアプローチで処理するナッグ) NULL 対応が有効化されると、MapForce は、データベース型に 応じて生成されナクロルの構文を適応します。

6.2.3.12 データベーステーブルアクション設定

データベーステーブルにマッピング接続を作成する都度、データベースアクションポタンカ影響を受けるテーブルの横に表示されます。このボタン をクリックすると、データベースの挿入、更新、アクションの削除、とその他のオプションを構成することのできる、データベーステーブルアクションダ イアログボックスか開かれます。

 最初のレコードの前に実行する SQL ステートメント ③ なし(N) ④ 全てのレコードを削除(L) ① 全ての子テーブルにある全てのレコードも削除 ⑦ カスタム SQL(C): 名レコードに対して実行するアウション 以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て真であれば、指定されたアクションが実行されます。 Action on record 以下の条件で更新… ● 残0を挿入 ● マックされた値 ● ForeignKey ● マックされた値 ● ForeignKey ● マックされた値 ● PhoneExt ● マックされた値 ● PhoneExt ● マックされた値 ● Title ● マックされた値 ● アウションを使用(T) ● パックサポズ: ● パック転送はデータペース接続によりサポートされていません。 	🕑 データベーステーブルのフ	アクション - Person			×
各レコードに対して実行するアクション 以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 ケロケッロンを使用して全ての入力データは DB デーブルのデータと比較されます。 ケロケッロンを使用して全ての入力データは DB デーブルのデータと比較されます。 ケロケッロンを使用して全ての入力データは DB デーブルのデータン比較されます。 ケロケッロンを使用(T) トランザクションを使用(T) ハルク転送を使用 (MapForce Server のみ)(B) パッチ サイズ: 1000 レコード パルク転送をすータペース接続によりサポートされていません。	最初のレコードの前に実行する ● なし(N) ● 全てのレコードを削除(L) ● カスタム SQL(C) :	SQL ステートメント	ある全てのレコードも削除		
以下のオペレーションを使用して全ての入力データは DB デーブルのデータと比較されます。 比較の結果が全て 頁であれば、指定された アクションが実行されます。 Action on record 以下の条件で更新 「「PrimaryKey 1 「「PrimaryKey 2 「PoreignKey 2 「PoreignKey <td>各レコードに対して実行するアク</td> <td>ワション</td> <td></td> <td></td> <td></td>	各レコードに対して実行するアク	ワション			
Action on record 以下の条件で更新 ・ 残りを挿入 アクションを追加(A) 「PrimaryKey 一致 マップされた値 アクションを追加(A) 「PrimaryKey 一致 マップされた値 アクションを挿入(I) □ EMail ・ マップされた値 アクションを追加(A) アクションを挿入(I) □ Last ・ マップされた値 ・ アクションを挿入(I) □ Last ・ マップされた値 ・ ・ □ Title ・ マップされた値 ・ ・ ・ 「Title ・ マップされた値 ・ ・ ・ 「 ・ マップされた値 ・ ・ ・ 「 ・ マップされた値 ・ ・ ・ 「マップされた値 ・ ・ ・ ・ 「 ・ マップされた値 ・ ・ ・ 「 ・ ・ ・ ・ 「 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ <td>以下のオペレーションを使用し 比較の結果が全て真であれな</td> <td>て全ての入力データは DE £、指定されたアクションが</td> <td>B デーブルのデータと比較されます。 「実行されます。</td> <td></td> <td></td>	以下のオペレーションを使用し 比較の結果が全て真であれな	て全ての入力データは DE £、指定されたアクションが	B デーブルのデータと比較されます。 「実行されます。		
PrimaryKey 一致 マップされた値 ProreignKey マップされた値 EMail マップされた値 First マップされた値 Last マップされた値 PhoneExt マップされた値 Title マップされた値 ・ アウションを削除(0) ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	Action on record	以下の条件で更新	残りを挿入	*	アクションを追加(A)
III ForeignKey ・ マップされた値 EMail ・ マップされた値 First ・ マップされた値 Debase マップされた値 PhoneExt ・ マップされた値 Title ・ マップされた値 トランザクションを使用(T) バルク転送を使用 (MapForce Server のみ)(B) パッチ サイズ: 1000 バルク転送はデータベース接続によりサポートされていません。 OK キャンセル	🛄 PrimaryKey	一致	マップされた 値		
Imail マップされた値 Imail マップされた値 Image: Imag	🛛 🕅 ForeignKey		マップされた 値		アクションを挿入(I)
Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint マップされた10 Imprint アクションを削除(D) Imprint アクションを削除(D) Imprint アクションを削除(D) Imprint Imprint	EMail		▼ マップされた 値		
Image: Control of the second sec			▼ マツノされに1世		
■ Title ■ Title ▼ワションを削除(D) ■ トランザクションを使用(T) ■ パルク転送を使用 (MapForce Server のみ)(B) パッチサイズ: 1000 レコード パルク転送はデータベース接続によりサポートされていません。 OK キャンセル	Last DecreExt		▼ マップされた 値		
● 「「」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」	III Title		▼ マップされた 値		
▼クションを削除(D) ■トランザクションを使用(T) ■ パルク転送を使用 (MapForce Server のみ)(B) パッチ サイズ: 1000 レコード パルク転送はデータベース接続によりサポートされていません。 OK キャンセル			(7) Child 12	-	
 ■トランザクションを使用(T) ■ パルク転送を使用 (MapForce Server のみ)(B) パッチ サイズ: 1000 レコード パルク転送はデータベース接続によりサポートされていません。 OK キャンセル 	•	1		Þ	アクションを削除(D)
□ パルク転送を使用 (MapForce Server のみ)(B) パッチ サイズ: 1000 レコード パルク転送はデータベース接続によりサポートされていません。 OK キャンセル	■トランザクションを使用(T)				
バルク転送はデータベース接続によりサポートされていません。 OK キャンセル		orce Server のみ)(B)	バッチ サイズ: 1000 レコード		
	パルカ転送けデータペーフ接続	続け上れ世ポートされていま	Ett 6.		
ок <i>キャンセル</i>			~ E100		
OK キャンセル					
					ок キャンセル

データベース テーブル アクション ダイアログボックス

以下は、データベーステーブルアクションダイアログボックスで使用することのできる設定の説明です。

最初のレコードの前に実行する SQL ステートメント

このグループボックスでは、最初のレコードがデータベースから取得される前に実行する SQL ステートメントを指定 することができます。以下にあるラジオボタンから選択を行ってください:

- **無し**(デフォルトの設定): アクションは取られることはありません。
- 全てのレコードを削除:「各レコードに対して実行するアクション」グループボックスにて定義されたテーブルアクションを実行する前に、選択されたテーブルにある全てのレコードを削除します。全ての子テーブルにある全てのレコードも削除チェックボックスを有効にすることで、選択されたテーブルにある子テーブル内にあるデータを取り除くことができます。例えば、次を参照してくたさいデータを複数ロレクされているテーブルに挿入する。
- カスタム SQL:テーブル全体を対象とたカスタムSQL ステートメートを記述することができます。

各レコードのために実行するアクション

設定は、マピングが実行されるときにこのテーブルマオして取られるデータベースアクションを指定します。テーブルアクションを管理するこよ 「アクションの追加」をクリックし、「アクションの挿入」おけは「アクションの削除」 ボタン、複数のアクションを必要であれば定義することが できます(例えば、「残いを挿入」 アクションの後に続く「以下の条件で更新...」アクション。

定義されたテーブルアケンコンは、左から右に処理されます。上の例では、「以下の条件で更新…」アケンコンカ最初に処理されます。更新の条件が満たされない場合、次のアケンコンカ処理されます(この例では、「残しを挿入」アケンコン)。以下の点に留意してくたさい

- 1つのアケションのすべての定義された条件が実行されるテーブルアケションのために満たされなければなりません。条件が全て満たされる場合、これらすべてのフィールドは、マナピング上でノースとターゲットアイテムの間の接続が存在する場所で更新されます。(条件が一致するアケションの右にある)その後のテーブルアケションは、そのレコードのために無視されます。
- 定義された条件が満たされない場合、テーブルアクションはスキップされ、次のアクションが右につかって)処理されます。
- 条件が満たされない場合、テーブルアクションは発生しません。

列の条件が入力アウションのために存在しないため、「全て挿入」おけは「残いを挿入」アウションの後に定義されているアウションは、 実行されません。この場合、その後のテーブルアウション列が削除されることを通知するダイアログボックスか表示されます。

「NULL Equal」列内のレコードの横にあるチェックボックスにより、適用できる場合は、MapForce に明示的に、全ての列がNULL 値を持ちそのように扱われることを命令することができます(次を参照してくたさい、データベーステーブルアクション内のNull の扱い)。

他のテーブルへの外部キールーションシップを持つマンピングを更新する場合、次のオプションを使用することができます

子テーブル内のデータの削除	親テーブルのためゴ以下の条件で更新…」アクションを選択すると、このオプションは、役に 立ちます。を選択すると、ソースファイル内のレコードの数かターゲルデータベース内のレコー ドの数と異なる場合、必要になる可能性かあり、データベースを同期化(子テーブル内で孤 立下データがない状態につしておくとは、かもしれません。以下も参照してくたさい、親テーブル を更新する際の子テーブルのためのオプション。
入力子データを無視する	テーブルの子テーブルレコードに影響を及ばすこと無く、ターゲト親テーブルを更新するため にこのオプションを使用することができます。以下も参照してくたさい、 <u>親テーブルを更新する際</u> の子テーブルのためのオプション

アクションの多種の組み合わせを説明している例に関しては以下を参照してくたさい

- データをテーブル:挿入する
- データを複数にしとりされているテーブルは挿入する
- <u>テーブルの更新</u>
- 親テーブルを更新する際の子テーブルのためのオプション
- 「以下の条件で更新し、残りを挿入」アクション
- 「以下の条件で削除…」アクション
- 「以下の条件で無視…」アクション

トランザクションの使用

この特定のテーブルアケションのオメクのデータベーストランザケションを有効化します。詳細に関しては、次を参照してくたさいトランザケションロ ールレックの使用。

バルク転送の使用

(複数のINSERT ステートメイを1つのケエリとしての)、シレク転送を有効化します。マメピングがMapForce Server により実行されてい る場合およびデータベースアクションが「全て挿入」の場合、、シレク転送はサポートされています。詳細に関しては、<u>、シレク挿入</u> (<u>MapForce Server</u>)を参照してくたさい。

6.2.3.13 例: XML から SQLite へのデータマッピング

このサンプルでは、XML ファイルはらデータを読み取り、SQLite データベースにデータを書き込む MapForce マピングの作成に必要なス テップについて説明されています。このサンプルゴはサンプルマピングデザイン (.mfd) ファイルが付属しています。このサンプルを開始する前 に、サンプルファイルを確認するゴは、次の、父を使用してファイルを開いてくたさい、 <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\XMLtoSQLite.mfd。

XML ファイルからデータを SQLite データベース 挿入することがこのサンプルの目的です。サンプルのゴールを達成するけっとう、データを書き 込むことの出来る SQLite データベースが必要しています。このサンプルの目的を達成するけっとり、データを書き込むことの出来る空のデータベ ースが必要しています。上で説明されているとおり、 Altova DatabaseSpy おけよ 公式な SQLite Web サイトからダウンロード数 ことの出来るコマイドラインシェルを使用して SQLite データベースを作成探索します。

SQLite データベースの作成:

(Altova MissionKit の一部、おけなタイアロンとして) DatabaseSpyis が使用中のコンピューターにインストールされている場合、新規 SQLite データベースを次のように作成することができます:

- 1. DatabaseSpy を起動します。
- 2. 「ファイル」メニューから、「データベースの接続の作成」をクリックします。
- 3. 「接続ウィザード」をクルクし、「SQLite」をクルクします。
- 4. 「新規 SQLite データベースの作成」をクリックし、c:\sqlite\articles.sqlite を ひとして入力し、「接続」をクリックします。
- 5. データノース名を設定するようにプロンプトされると、デフォルトの名前のままします。
- 6. 新規 SQL エディターを開き(Ctrl+N) データベース マ対して次のカエ を実行します:

create table articles (number smallint, name varchar(10), singleprice real);

それ以外の場合、下のステップに従いデータベースを作成してくたさい

- SQLite ダウロードページからWindows のかの、SQLite コマドラインシェルをダウロードしてくたさい。 (<u>http://www.sqlite.org/download.html</u>)、そして、使用中のローカルマシンのディンケトリで.zip アーカイブを解凍してくたさい (このサンプルのすぶはこま、c:\sqlite を使用してくたさい)。
- 2. c:\sqlite\sqlite3.exe を実行して、次のステートメートを入力します:

create table articles (number smallint, name varchar(10), singleprice real);

これによけーブルarticles がインメモリデータベース内に作成されます。テーブルarticles はっつの別によ構成されています。number、name、および singleprice。この別の目的はXML ソーススキーマ内で定義された同じ名前を持つ要素からのデータを補完するためです。各列は、その別内に保管されるデータにふされていデータ方と共に宣言されています。

3. コマドを実行します:

.save articles.sqlite

これによりインメモリのデータベースか現在の作業パマに保存されます: c:\sqlite\articles.sqlite。この、マを後のステップで参照する必要があることに注意してください。
このサンプルで必要とされるSQLiteデータベースが作成されまた。

SQLite マッピングデザインに XML を作成する方法:

- 1. MapForce を起動し、(メニューコマボ「出力 | ビルトイン実行エンジン」を使用しての変換言語がBUILTHN てあることを 確認してください。
- 2. <Documents>\Altova\MapForce2021\MapForceExamples\Tutorial フォルダー内にあるファイル Articles.xml を(「挿入 | XML スキーマ/ファイルの挿入」を使用してマッピングエルアに追加します。
- 3. 前のステップ(メニューコマボ「挿入 | データベースの挿入」)で作成されたデータベース articles.sqlite をマピングエルに 追加し、SQLite を選択します。
- 4. 「接続」をクリックします。データベースオブジェクトを選択するようこプロンプトされると、articles テーブルを選択します。

🕞 データベース オブジェクトを挿入
テーブル、ビュー、ストアドプロシージャを選択す トはローカルの mfd ファイルに保存されます。
Articles Articles Articles D
プレビューを表示(P) >>

5. 下に示されるようは接続を描きます:



6. データベースエンポーネント上のA: In ボタンをクリックし、「全てのレコードを削除する」オプションを選択します。これにより、マンピング実行される都度、重複を回避するナッシン全ての既存の行か最初に削除されます。

🕐 データベーステーブルのアクション - articles		×
最初のレコードの前に実行する SQL ステートメント ◎ なし(N) ◎ 全てのレコードを削除(L)		
各レコードに対して実行するアクション ここで定義された演算子を使用して、すべての入力データは DB テーブルデータと比較されます。 NULL 値を等価として扱うように「NULL equal」チェックボックスをチェックしてください。 全ての比較が true の場合、特定のアクションが実行されます。		
レコードに対するアクション NULL 等 全て挿入 <	*	アクションを追加(A)
□ number マップされた値 マ □ name マップされた値 マ □ singleprice マップされた値 マ		アクションを挿入(I)
	-	アクションを削除(D)

7. メインマッピングウィボウの「出力」ダブをクリックします。MapForceは、内蔵の実行エンジンを使用してマピングを実行し、出力ウィボウ内にSQLクロを表示します。

The following SQL statements are only for preview and may not be executed in another SQL query tool! 2 To execute these statements use function "Run SQL-script" from menu "Output". 3 4 Connect to database using the following connection-string: 5 6 C:/sglite/articles.sglite */ 7 8 PRAGMA foreign_keys = ON; 9 10 DELETE FROM "articles" 11 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (1, 'T-Shirt', 25) 12 13 14 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (2, 'Socks', 2.3) 15 16 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (3, 'Pants', 34) 17 18 INSERT INTO "articles" ("number", "name", "singleprice") VALUES (4, 'Jacket', 57.5) 19

8. (メニューコマナ「出力 | SQL-スクリプトの実行」)SQL スクリプトを実行し、データベースを作成します。MapForce かランタ イムエラーニ遭遇したは、場合、レコードはSQLite データベース 挿入されます。

2	The following SQL statements were executed during "Generate output" function. Every single result is written right to the ">>>" string.
4	These statements are only for preview and may not be executed in another SQL query tool!
5	The database was connected using the following connection-string:
6	C:/sqlite/articles.sqlite
2	^/
ğ	PRAGMA foreign keys = ON
10	>>> OK. 0 row(s).
11	
12	DELETE FROM "articles"
13	>>> OK. 0 row(s).
14	INCERT INTO "actividae" ("acumbae", "come", "cinglangiae") (ALLIES (1, "T.Shiet", 25)
15	->>> OK 1 row(s)
17	Service Content of the Content of
18	INSERT INTO "articles" ("number", "name", "singleprice") VALUES (2, 'Socks', 2.3)
19	>>> OK. 1 row(s).
20	
21	INSERT INTO "articles" ("number", "name", "singleprice") VALUES (3, 'Pants', 34)
22	>>> OK. 1 row(s).
25 24	INSERT INTO "articles" ("number", "name", "singlenrice") VALUES (4, "lacket", 57,5)
25	>>> OK, 1 rowls).
26	

SQLite データベースにデータが正確に挿入されているかチェックする方法:

- 1. MapForce の「DB クエリ」 タブをクトックします。
- 2. 「DB クエリ」 ペインの上にあるドロップダウノノストから articles エトリを選択します。
- 3. SQL エディター内に次のカエを入力します:

select * from articles;

おは、次のステップをおこなってください

1. ファイルc:\sqlite\sqlite3.exe を実行し、コマイを使用してデータベースを開きます:

.open articles.sqlite

2. 次のSelect ステートメーを実行します:

select * from articles;

テーブルデータの選択のアプローチー、関わらず、(MapForce まけはSQLite シェル)、クエリはつの行を返します。これは、このサンプルの目的であるノースXML ファイル内のレコードの数に対応しています。

6.2.4 データベースのデータのジョイン

データベースからデータを読み取るマンング内では、データベースオブジェクトテーブルオオオビューなどを、マンングインショインコンポーネントを 追加することにどうインすることができます。例えば、データがルーショナルデータベース内に保管される外部キー・ルーションシップにより、やう ンドされた2つ、オオオ複数のテーブルからのデーダを組み合わせることができます。INNER JOIN(オオオ適用可能な場合 LEFT JOIN) オペンーションを使用して2つませオ複数のテーブルがジョインされた箇所にデータベース SQL クロを実行する結果と同じ結果になります。

ジョインエンポーネントは接続されているデータの型により、ジョインの操作は、標準(非-SQL)モード、まけはSQL モード内で行うことができます。非-SQL モード内のジョインは、マノピングエンジンメにより実行され、SQL モード内のジョインは、マノピングがデータを読み取るデータベー スにより実行されます。

非-SQL モード内のジョインは、コンポーネント型を入力としてサポートするオダ柔軟性に富んでします(例えば、ジョインは異なるデータベース からのテーブル間、おさはXML 構造とデータベーステーブル間であることができます)。非-SQL ジョインのサンプルに関しては<u>例 XML 構造の</u> ジョイン内で詳しく説明されてします。。その一方、非-SQL ジョインの実行は、マンピングエンジムこと更にメモレを必要とする実行となりま す。(これは、比較の総数がジョインされた構造のグロンジョイン、おさまデカルト積を表すためです)。通常、データを中心としたマンピング内で は、この過程はとても速く、影響が小さいです。しかしながら、ジョインしたデータノースが大量の記録によい構成される場合、マンピング)は実行 するために時間を要します。マンピングが多数のレコードを処理する場合、マンピング実行をスピードアップするジョイン最適化に特化した機能 を含むMapForce Server Advanced Edition にライセンス供与することを考慮してくたさい。

SQL モード内のショインは、使用することのできるテーブルませまビューなどのデータベースオブジェクトを入力として受け入れます。しかしなが ら、データベースによりネイティブにより良いマンピングを行うため、柔軟性のある非-SQL ジョインとしては受け入れません。詳細に関しては <u>SQL モード内のショイン</u>を参照してください。

メモ ジョインコンポーネトの使用のみがデータベーステーブル、おけビューをジョインする唯一の方法ではありません。データベースへの ジョインの適用はSQL SELECT ステートメトを使用して行われます。データベースデータの選択(SQL SELECT)を参照し てください。SQL SELECT ステートメトのジョインコンポーネト主要な違いは、前者は手動で作成され、そのために柔軟性が あります。しかしながら、SQL ステートメトを手動で作成を希望したい場合、ジョインコンポーネトは簡単な代替方法です。

ジョインコンポーネントを追加する方法:

- 1. マノピング変換言語をBUILT-IN に設定します、これを行うココージャール・ボタククルクします、おけば出力 | ビルトイン 実行エンジン] メニューコマンドを使用します)。
- 2. [挿入] メニューから[ジョイン] をクリックします。 おけよ[ジョイン] № ツール デーボタンをクリックします。 ジョインエルポーネント が マノピングに表示されます。 デフォルト では つの構造からデータを受け入れるけな、 2つの nodes/rows 入力が存在します。 必要 であれば入力の追加](•) パタンをクリックして、 ジョイノニ新し、入力を追加することができます。 <u>3つ以上の構造をジョインする</u>を 参照してください。



- 3. ジョインコンポーネトのnodes/rowsアイテムどうインされる構造を選択します。
- 4. ジョイン(おけま複数の条件)のための条件を追加してください。これを行うけま、ジョインロンポーネントを右クトックしプロンティを 選択します。ジョインの条件は、関数のブール値の結果をジョインエンポーネントのcondition アイテムは接続することには、マッ ビングから直接追加することができます。データベースがジョインされる特定の場合、ジョインの条件は自動的にMapForce により 作成されます。詳細に関しては、ジョインの条件の追加を参照してください。

灹

- ジョインコンポーネントは、BUILT-IN 実行エンジン内でのみサポートされています。C#、C++、お」はJava 内でのコードの生成はサポートされていません。
- 構造がサポーされていない、おけおショインのための入力ソースによりサポートされて以内場合、MapForceマピング上直接と ントを表示するか、おけはマピングを検証する際に、メッセージウィンドウ内にといたを表示します。(マピングの検証参を照してくだ さい)。
- ジョインエンポーネントをインラインユーザー定義関数の結果まけは入力/ ラメーター は接続しないでください。このような接続が存在すると、マメピングの検証中に検証エラーが発生します。
- 使用することのできるデータベースコンポーネント (テーブルませ)まだコーなどの)を直接ジョインコンポーネント (注接続すると、SQL モード) SQL ポタンカ自動的 にジョインコンポーネントの右上に表示されます。 有効化されると、このパタンは、ジョイン操作に適用することのできる特別な SQL 機能を提供します (SQL モード内のジョインを参照してください)。
- ジョインされたアイテムの出力を他のジョインコンポーネントに接続することはできません。しかしなから、ジョインの結果の一部を他のジョインは接続することは可能です。

6.2.4.1 SQL モード内のジョイン

使用することのできるデータベースエポーネト(テーブルまけまビューなどの)を直接ジョインエンポーネトに接続すると、SQL モード「SQL ボタン がジョインエンポーネトの右上に表示されます。SQL モードが有効化されていると、ジョインオペレーションは、マンピングがデータを読み 込む箇所でデータベースにより行われます。すなオち、MapForce は、ジョインイン使用される全てのテーブルからのデータを選択し結合するた め、適切な SQL 構文を使用して、内部でデータベースにクエレを送信します。重要な点は、SQL を書く必要がない点です。下のサンプル で示されている通りジョインエンポーネトがマンピング上でデザインされた方法をベースして必要なくクエルが生成されます。

SQL モードを使用可能にするためコよ、以下の条件が満たされている必要があります。

- 1. ジョインされる両方のオブジェクト(テーブルませ」ゼュー)は、同じデータベースから与えられなければなりません。
- 2. ジョインされる両方のオブジェケトは、同じ MapForce ユポーネトから与えなければなりません。(ユポーネント内でデータベー スオブジェクトを追加および削除する方法は以下のとおりです:データベース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 追加/削除/編集データベースオブジェクトを選択します)。
- 3. ジョインの条件(おけま条件)は、マナング上ではなく、コンポーネントプロレティからのみ定義されます(ジョインコンポーネントのヘッダーを右クトックして、プロノティを選択します)(ジョインの条件の追加を参照してくたさい)。
- メモ SQL モード内でデータベーステーブルがショインされると、MapForce は、テーブル間で検出されオット部キーリルーションシップをベー スコして、ジョインの条件を自動的に作成します。自動的なジョインの条件が起こるコは、データベーステーブルがMapForce コン ポーネント上で親子関係内に存在する必要があります(すなオち、1つのテーブルが「親」まけまずよ」で、他のテーブルがセンポーネ ント上に存在する必要があります)。例 SQL モード内のテーブルのジョインを参照してくたさい。

4. すべてのデータベーステーブルは、現在のターゲトコンテキスト内にまた存在することはできません。ジョインの結果がターゲトコンポ ーネント内で使用される場合、ジョインされたテーブルは直接または間接的にターゲトの親ノードに接続されている場合がありま す。マンピングの実行方法に関してはマッピングルールと戦略を参照してくたさい。

ジョインコンポーネトの右上にあるSQL(30)がタンを使用して、SQLモードを閲覧おけ着理することができます:

- SQL モードが無効化されています。(ジョインはMapForce (おけば適用可能な場合 MapForce Server)にお実行されます。
- SQL モードが有効化されています(ジョインムデータベースには実行されます)。

Sell ボタンの不足している場合、SQLモードは意味を成されてとを示します、おけよジョインされたデータのためにサポートされていたいとを意味します。

特定のケースの場合、SQLモードは明示的に無効化されている必要があります 501)、例

- マピングがショインコンポーネントプロ、ティの外側のショイン条件を必要な場合(すなオカ、マピング上で定義されている条件が ジョインコンポーネントの condition アイテム 接続されている場合)。
- 異なるデータベースからのテーブルをジョインする場合。異なるデータベースからテーブルをジョインする場合、標準(非-SQL)ジョインを使用します。

ジョインモードの変換

SQL モード 500 内でエンポーネントをショインする場合、データベーステーブルをジョインおしお以下の方法のI つでビューすることができます:

- INNER JOIN 両方の入力セナ内の条件を満たすレコードのみがショインコンポーネントによし返されます。
- LEFT OUTER JOIN 「左端のテーブル空のすべてのレコードを含むジョインエンポーネント (MapForce 内では こればショインエンポーネントの一番上のテーブルです) どうイン条件を満たす後にジョインされたテーブルからのレコード。

テーブルのジョインモードをビュー、おけおショインコンポーネント上でビューするはおショインされたテーブルませましての前に表示されるアイコン
に着目してください。以下のアイコンの一つか最初のたのを除いてジョインされたテーブルませませっ一のために表示されます。

- 内部結合 ᢂ
- 左結合 ≥

ジョイノニ関する詳細のとこトを表示するコよアイコノニカーノルをポイントします。



ジョインモードを変更するコよ次の、ずれかを行ってくたさい

- 各ジョインされナテーブルおけまビューの前の内部結合 № おけまた結合 № アイエンをクトックし 内部結合 おけま 左外側結 合をエンテキストメニューから選択します。
- ジョインコンポーネントの2番目(おけお番目、4番目)のジョインされたテーブルおけまごユーを右クリックし型のジョイン|内部結合 おけま型の結合 | 左外側結合 をコンテキストメニューから選択します。



以下の点に注意してくたさい

- LEFT OUTER JOIN にジョインモードを変更すると、上のテーブルオゴオビューインタインの左側を表します。
- ジョインモードの変更はINNER JOIN おけましEFT JOIN はデータベース内のSQL クロル影響する同様にジョインコンポー おようにといたされたデーター影響を与えます。

エイアス名

ジョインされナデータベーステーブル、封まギュートコ同一のフィールド名が含まれていることが頻繁にあります。例えば、SQL モードが有効化されると、このようなアイテムは、キーワード「AS」によりプレフィックスされナニャポーネント上に表示されます。例えば、2つのジョインされナテーブ ルが「id」フィールドを含む場合、このフィールドは、最初にジョインされナテーブル上で「id」として表示されます。そして、2番目にジョインされ れナテーブルですが AS id2」として表示されます。ジョインされナテーブルは、例えば、同じテーブルカゲーブル内でジョインされた場合、エイリアス名を生成することができます。

マピング上で後に参照する場合エイリアスフィールドおゴムテーブル名は重要です。例えば、ジョインコンポーネントの出力がSQL WHERE /ORDER コンポーネントに接続されている場合などです。目的を達成するゴム、ジョインコンポーネントの出力を、SQL WHERE とORDER BY 句を入力する SQL WHERE /ORDER コンポーネントに接続することができます。

WHERE 句からフィールドを参照するコよ、フィールド名が後に続き、ピルド(.)文字が後に続く、テーブル名を書き込みます。テーブルエ イルアスを参照するコよ、ジョインコンポーネントに表示される順のエイルアス名を使用します。ORDER BY 句内では、同じテクニック (table.field)、おけよ(「AS」の後に表示される名前であるコエイルアスンィールド名を書き込みます。

SQLWHERE/ORDER 句を使用するマピングのサンプルは関してはサンプルSQLモード内のショインテーブルを参照してくたさい。

メモ データベーステーブルと泊インエンポーネント間でSQLWHERE/ORDER エンポーネントは許可されていません。ジョインエンポー ネトの前ではない後に追加することができます。SQLWHERE/ORDER エンポーネントの詳細に関しては、<u>データベースデ</u> <u>ータのフィルターと並べ替え(SQLWHERE/ORDER)</u>を参照してくたさい。

6.2.4.2 例: SQL モード内でテーブルをジョインする

このサンプルは、MapForce ジョインコンポーネントを使用した、データベーステーブルからのデータをジョインする方法について説明しています。ジョインの操作は、SQLモード内のジョインで説明されているようこ、SQLモード内で実行されます。3つ、おけお以上のテーブルワークを同様にごっインする場合も同様についます。例、CSVレポートを複数のテーブルから作成するを参照してください。

このサンプルは、マピングサンプルが伴い、次の、次で検索することができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\JoinDatabaseTables.mfd.



JoinDatabaseTables.mfd

上のマンピングの目的は、2つのソースデータベーステーブルからのデータを単一のターゲオ CSV ファイルに結合することです。下のデータベ ースダイアグラムに示されているとおり、最初のテーブル(users)は個人の住所を保管し、2番目のテーブル(addresses)は個人の 名前と電子メールアドレスを保管します。2つのテーブルは、共有のフィールドによりレンされています(users内のid がaddresses内 のuser id に対応しています)。データベース用語では、この種類おルーションは外部キーリルーションシップ」と呼ばれています。



簡素化のために、下のイメージは両方のテーブルの実際のデーダを表示しています。

first_name ·	last_name ·	email	email •			ated_at	•	upd	lated_at ·				
Marquita	Bailey	m.bai	ley@nanonul	l.com	201	6-12-29 14:37	7:14	[NU	LL]				
Sharda	Junker	s.juni	ker@nanonul	.com	201	6-12-29 14:37	7:14	[NU	LL]	ers			
Totie	Rea	t.rea(@nanonull.co	m	201	6-12-29 14:37	7:14	[NU	LL]	sn			
Tobie	Hughey	t.hug	hey@nanonu	I.com	201	6-12-29 14:37	7:14	[NU	LL]				
Eadith	Lafreniere	e.lafr	eniere@nand	null.com	201	6-12-29 14:37	7:14	[NU	LL]				
Yehudi	Sponga	y.spo	nga@nanoni	ull.com	201	6-12-29 14:37	7:14	[NU	LL]				
Laurianne	Huisman	I.huis	man@nanon	ull.com	201	6-12-29 14:37	7:14	[NU	LL]				
Fred	Weinstein	f.wei	nstein@nano	null.com	201	6-12-29 14:37	7:14	[NU	LL]				
Mia	Dahill	id 🔹	user_id 🔹	is_shippin	g •	is_billing 🔹	type	ð •	city	٠	street •	number 🔹	
June	Leiker	1	1	1		0	wor	rk	Bridgedell		Maple Lane	1	es
Benedick	Kocyk	2	1	0		1	hor	ne	Bridgedell		Olive Street	6	ess
Andrej	Hildebrand	3	3	1		1	hor	ne	Roseford		Evergreen Lane	34	ddr
Ariel	Phelan	4	4	1		1	wor	rk	Beardale		Route 44	9	o l
Matthaeus	Hulick	5	6	1		1	hom	ne	Johnson Ci	ty	Franklin Avenue	11	
Lotta	Mendes	6	7	1		1	hom	ne	North Kings	stown	Beach Alley	5	
Jessey	Decelles	7	8	1		1	hom	ne	Merrowme	adow	Freybeach Street	85	
Hilda	Lees	8	10	1		1	wor	rk	Barrowedg	je	Penn Street	8	
Mark	Marzolla	9	12	1		1	hom	ne	Elfville		Creek Road	3	
Dannie	Vignola	10	13	1		1	hom	ne	Roseford		Bowman Ave.	853	
Lanita	Krysiak	11	14	1		1	wor	rk	Beardale		Iroquois Street	98	
		12	17	1		1	hom	ne	Bridgedell		Smith Road	7	
		13	18	1		0	hom	ne	Roseford		Wood Street	7	
		14	18	0		1	wor	rk	Johnson Ci	ty	Thorne Lane	9677	
		15	20	1		1	hom	ne	Mechanics	ville	Vine Street	9065	
	first_name Marquita Marquita Sharda Sharda Totie Tobie Eadith Yehudi Iaurianne Fred Mia June Andrej Andrei Jessey Hilda Dannie Lanita	first_namelast_nameMarquitaBaileyMarquitaBaileyShardaJunkerTotieReaTobieHugheyEadithLafreniereYehudiSpongaYehudiMarsanFredWeinsteinMiaDahilJuneLeikerAndrejHulckAtrielMendesJesseyDecellesHidaLeesMarkMarzollaDannieKrysiak	first_name last_name email Marquita Bailey m.bail Sharda Junker s.junk Totie Rea t.read Tobie Hughey s.junk Tobie Hughey e.lafr Tobie Lafreniere e.lafr Yehudi Sponga y.spo Laurianne Huisman I.huis Fred Weinstein id June Leiker id June Leiker id June Hildebrand 3 Andrej Huick 5 Josesy Decelles 7 Hida Lees 3 Mark Marzolla 9 Dannie Vignola 10 Lanita Krysiak 11	first_name last_name email Marquita Bailey m.bailey@nanonul Sharda Junker s,junker@nanonul.co Totie Rea t.rea@nanonul.co Totie Hughey t.rea@nanonul.co Totie Lafreniere e.lafreniere@nanonul.co Eadith Lafreniere e.lafreniere@nanonul.co Yehudi Sponga y.sporg@nanonul.co Yehudi Sponga t.nusmanonul.co Yehudi Sponga y.sporg@nanonul.co Yehudi Dahil Inusmanonul.co June Huisman t.nusmanonul.co Maa Dahil Inusmanonul.co June Kocyk Inusmanonul.co June Leiker Inus June Kocyk Inus Andrej Hulck Inus Andrej Hulck Inus Jussey Decelles Inus Junak Lees Inus Mark Marzolla Inus Mark Krysiak Inus Junita Krysiak Inus June Inus Inus June Krysiak Inus Junita Krysiak Inu	first_namelast_nameemail	first_namelast_nameemailcreationMarquitaBaileym.bailey@nanonul.com201ShardaJunkers.junker@nanonul.com201TotieReat.rea@nanonul.com201TobieHugheyt.hugheyanunul.com201EadithLafrenieree.lafreniere@nanonul.com201YehudiSpongay.spoga@nanonul.com201IaurianneHuismanLhuismanoul.com201IaurianneHuismant.hugisenoul.com201JuneQalaniliduser_id201JuneLeiker11201JuneLeiker111JuneLeiker111AndrejHildebrand331JesseyDecelles781JanikLees8101HidaLees11MarkMarzolla9121JanikKrysiak10131LanitaKrysiak11141JanikSignal111JanikKarsola1011JanikKarsola1011JanikKarsola1011JanikKarsola1011JanikKarsola1011JanikKarsola1011JanikKarsola1011J	first_name last_name email	first_name last_name emai creat_at Marquita Bailey m.bailey@nanonull.com 2016-12-2914:37:14 Sharda Junker s.junker@nanonull.com 2016-12-2914:37:14 Totie Rea trea@nanonull.com 2016-12-2914:37:14 Tobie Hughey thughey@nanonull.com 2016-12-2914:37:14 Tobie Lafreniere e.lafreniere@nanonull.com 2016-12-2914:37:14 Yehudi Sponga y.spoga@nanonull.com 2016-12-2914:37:14 Yehudi Sponga y.spoga@nanonull.com 2016-12-2914:37:14 Laurianne Huisman f.neire@nanonull.com 2016-12-2914:37:14 Yehudi Sponga y.spoga@nanonull.com 2016-12-2914:37:14 Iaurianne Huisman f.neire@nanonull.com 2016-12-2914:37:14 Iaurianne Sponga y.spoga@nanonull.com 2016-12-2914:37:14 Madmanne Sponga y.spoga@nanonull.com 2016-12-2914:37:14 June Dahil f.de ss.shippin s.bilinga hypei June Dahil f.de ss.shippin s.bilinga hypei	first_name last_name email create_at upd Marquita Bailey mbailey@nanonull.com 2016-12-2914:37:14 [NU Sharda Junker s,junker@nanonull.com 2016-12-2914:37:14 [NU Tobie Rea thughey thughey 2016-12-2914:37:14 [NU Tobie Hughey thughey cold-12-2914:37:14 [NU Eadith Lafreniere elafreniere@nanonull.com 2016-12-2914:37:14 [NU Yehudi Sponga yspungamanoull.com 2016-12-2914:37:14 [NU Yehudi Sponga yspungamanoull.com 2016-12-2914:37:14 [NU Yehudi Sponga yspungamanoull.com 2016-12-2914:37:14 [NU Sponga yspungamanoull.com 2016-12-2914:37:14 [NU Marduniane Pahil fulcerane is_shipping is_shipping is_shipping June Leiker 1 1 1 is_shipping is_shipping June Leiker 1 1 1 is_shipping is_shipping Andrej Hulckrame <th>first_namelast_nameemailcreate_atupdate_atMarquitaBaileym.bailey@nanoull.com201-12-2914:37:4NULLShardaJunkertrea@nanoull.com201-12-2914:37:4NULLTotieReatrea@nanoull.com201-12-2914:37:4NULLTotieHugheythugh=y@nanoull.com201-12-2914:37:4NULLEadithLafreniereelafreniere@nanoull.com201-12-2914:37:4NULLYehudiSpongaysp-s@nanoull.com201-12-2914:37:4NULLLaurianneHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLFredWeinsteinfweinstein@nanoull.com201-12-2914:37:4NULLJuneHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLIdamianeHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLJuneGabillid<user_id< td="">is_shippiris_billingNULLIdamianeHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLJuneGabillid<user_id< td="">is_shippiris_billingNULLMaaDahillid<user_id< td="">is_shippiris_billingNULLisJuneLeiker111is_shippiris_billingNULLisJuneLeiker1111isisisisisJuneHuibband1111i<isisisisisisisAdrid<!--</th--><th>first_namelead_aneemailccemailmolloopMarquitaBaieymm</th><th>frst_name last_name genut c c u <thu< th=""> u <thu> u u<th>frst_namelast_namelemai<th< th=""></th<></th></thu></thu<></th></user_id<></user_id<></user_id<></th>	first_namelast_nameemailcreate_atupdate_atMarquitaBaileym.bailey@nanoull.com201-12-2914:37:4NULLShardaJunkertrea@nanoull.com201-12-2914:37:4NULLTotieReatrea@nanoull.com201-12-2914:37:4NULLTotieHugheythugh=y@nanoull.com201-12-2914:37:4NULLEadithLafreniereelafreniere@nanoull.com201-12-2914:37:4NULLYehudiSpongaysp-s@nanoull.com201-12-2914:37:4NULLLaurianneHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLFredWeinsteinfweinstein@nanoull.com201-12-2914:37:4NULLJuneHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLIdamianeHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLJuneGabillid <user_id< td="">is_shippiris_billingNULLIdamianeHuismanLhuis=m@nanoull.com201-12-2914:37:4NULLJuneGabillid<user_id< td="">is_shippiris_billingNULLMaaDahillid<user_id< td="">is_shippiris_billingNULLisJuneLeiker111is_shippiris_billingNULLisJuneLeiker1111isisisisisJuneHuibband1111i<isisisisisisisAdrid<!--</th--><th>first_namelead_aneemailccemailmolloopMarquitaBaieymm</th><th>frst_name last_name genut c c u <thu< th=""> u <thu> u u<th>frst_namelast_namelemai<th< th=""></th<></th></thu></thu<></th></user_id<></user_id<></user_id<>	first_namelead_aneemailccemailmolloopMarquitaBaieymm	frst_name last_name genut c c u <thu< th=""> u <thu> u u<th>frst_namelast_namelemai<th< th=""></th<></th></thu></thu<>	frst_namelast_namelemai <th< th=""></th<>

users テーブル内の各ユーザーはゼロの値、おけまaddresses テーブル内に複数の注所を有することができます。例えば、ユーザーは、 「home」の種類に属する1つの注所、おけま2つの注所(「home」と「work」の2種類の注所)、おけは注所なしを選ぶことができます。

マピングの目的は住所テーブル内に住所か少なくといっ記入されている全てのユーザーのデータ(名、姓、電子メール、住所など)を抽出 することです。また、特定の種類の住所を簡単に抽出することもできます(例えば、自宅の住所、ませば勝先の住所のみを抽出することが できます)。(「home」、ませは「work」)などの抽出する住所は、マピングにンテメーターとして与えられます。抽出された個人のレコードは 姓によりアルファベット順に並べ替えられます。

下で説明されているとおり、ジョインコンポーネントを使用して、マンピングの必要条件が満たされます。

メモジョインコンポーネトの使用のみがデータベーステーブル、おけビューをジョインする唯一の方法ではありません。データベースへのジョインの適用はSQL SELECT ステートメトを使用して行われます。データベースデータの選択(SQL SELECT)を参照してください。SQL SELECT ステートメトのジョインコンポーネント主要な違いは、前者は手動で作成され、そのために柔軟性があります。しかしながら、SQL ステートメートを手動で作成を希望しない場合、ジョインコンポーネントは簡単な代替方法です。

ステップ 1: ソースデータベースを追加する

- 1. [挿入] メニューから [データベース]をクリックしくたさい。(おけよ 「データベースの挿入」 🖯 ツール いーボタンをクリックします。)
- 2. [SQLite]をデータベースの型とて選択し、[次へ]をクリックします。
- 3. フォルダー内で使用することのできる Nanonull.sqlite ファイルを参照します。 <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\, を選択し[クリック]をクリックします。
- 4. プロレプトされると、 addresses と users テーブルを選択します。

「戸データベース オブジェクトを挿入
テーブル、ビュー、ストアドプロシージャを選択するか、 独自の SQL トはローカルの mfd ファイルに保存されます。
- - - - - - - - - -
▼ ▼ ◆ SELECT ステートメントの追加/編集 ■ ■ Nanonull2 ■ ■ ● main ■ ■ ● ● テーブル ■ ■ ● ● ユーザーテーブル ■ ■ orderedproducts ■ ■ orders ■ ■ products ■ ■ ■ □ ■ users
プレビューを表示(P) >>

ステップ 2: ジョインコンポーネントを追加する

- 1. [挿入] メニューから[ジョイン] をクリックします。(おけは[ジョイン] 🕺 ツール いーボタンをクリックします)。 2. users テーブ いうごうインエンポーネトの最初の入力に接続線を描きます。
- 3. users テーブルを展開し、addresses テーブル(users の子)かジョインコンポーネントの2番目の入力に接続を描きます。 ● ボタインド必要に応じて更にテーブルを追加することができます。しかしながら、このサンプルでは、2つのテーブルのみがらったさ れます。



- メモ addresses テーブル(users の子ではない)から直接接続を追加することができます。しかしなから、この場合、ジョイン条件 の追加で説明されているとおり、ジョインの条件は手動で定義されなければないません。このサンプルの目的のために、上に示される ように接続を作成してくたさい。これにより、必要なジョインの条件が自動的に作成されます。
 - 4. ジョインエポーネト上で使用することのできる[ジョインの条件の定義] Sel ボタンをクリックします。ジョインの条件が自動的に 作成される事に注意してください(users.id = addresses.user id)。

🕑 ジョインの条件の定義		×
以下の全ての条件が true の場合のジョインの構造:		条件の追加(A)
I addresses.user_id	= 🛯 🖗 users.id AS id2	X
・ 条件を定義するには、各構造からアイテムを選択して比較し 構造 1:	ます。 構造 2:	• ()
Image: margin black of the second		

ステップ 3: ターゲナ CSV コンポーネントの追加

- 1. 「挿入」メニューカジテキストファイル」をクルクします。(おけよ [テキストファイルの挿入] 国 ツールドーボンをクルクします。)
- 2. テキストの処理モードを選択されるように問われると、標準 CSVのために簡単処理を使用する...を選択します。。
- 3. [フィールドの追加]を数回クルクして、7つのCSV フィールドを作成します。他の設定をそのままします。

71	ールド1	_	フィー	ルド2		フィールド3	_	フィールド4		フィールド5		フィールド6	_	フィールドフ	
string		-	string		-	string	•	string	•	string	•	string	•	string	-
-															
															- Þ.
		<u> </u>	<u></u>		1		_				1				
71	-ルドを通	呈力の	(<u>A</u>)	フィール	バを	を挿入(<u>N</u>)	77	ールドを削除(<u>R</u>)	$\langle \langle \rangle \rangle$	J				

4. 各フィールドのタイトルのセルをダブルクリックして、詳細を説明する名前を与えます(これによりマッピングを簡単に読み込むことができます)。

4		名		電子メール	/	国名		都市名		通り名		番地名	
tring	•	string	•	string	-	string	•	string	•	string	•	string	•
4													

5. ジョインエポーネトとCSV エポーネト間の接続を下に示されるようは書きます。ジョインエポーネトのjoined アイテム とターゲトコポーネトのRows 間の接続はジョイン条件を満たす数量のレコードを(行)をターゲル内に作成することを意味し ます。



ステップ 4: SQL WHERE /ORDER 条件と入力パラメーターを入力する

- 1. ジョインエンポーネントのjoined アイテムとターゲトコンポーネントのRows アイテム間の接続を右クトックし、[SQL-WHERE /ORDER を挿入する」を選択します。
- 2. WHERE とORDER BY 句を下に示されるように入力します。

🕑 SQL-WHERE/ORDER プロパティ	×
パラメーターを作成するには、WHERE 条件で使われる名前の前に ':'を入力。 例: Name = :Name AND Age >= :ValidAge SELECT (フィールド) FROM addresses, users WHERE	
address.type = :address_type	*
パラメーター: <mark>パラメーター 型</mark> address_type string	-
ORDER BY last_name	*
ОК ++>>	11.

3. マピング上で、([挿入 | 入力の挿入] メニューコマドを使用して) 入力コンポーネントを追加し、前のステップで作成された address_type / デメーター に出力を接続します。



4. 入力コンポーネントをダブルクトックして、下に示されるように構成します。MapForce内でマンピング出力をプレビューするオータリニデ ザイン時刻値(この場合、「home」)が必要とされます。職場の注所を抽出するオータリニプレビューを行うには、この値を[work] と 置き換えてくたさい。

 〇〇入力の編集
名前(N): input
デー久型(D): string ▼
▼入力は必須(I)
デザイン時実行
☑ 値を指定(S)
值(V): home
ОК キ ャンセル
h

マピングの説明

ステップ2 は定義されているジョインの条件 users.id = addresses.user_id を満たすレコードのみがターゲルにコピーされることを 確認します。2つのテーブルは外部キーレーションシップにより、イイドされており、マノピング接続は状況に応じて描かれているけっかジョインの 条件は自動的に追加されています。テーブルレーションシップの詳細に関しては、データベースレーションシップの処理を参照してくたさい。 このサンプルは既存のテーブルレーションシップを活用しているけっか、ジョイン条件を手動で定義する必要がありません。ジョイン条件の定義を 手動で行う方法に関するサンプルは例CSVレポートを複数のテーブルから作成するを参照してくたさい。

2つのノーステーブルは、同じデータベースおよび同じエンポーネントからであり、このショインはSQL SQL モード を活用することができます。 SQL モードが有効化されているため、ジョインの操作はMapForce*ではなく、データベースにより行われます。SELECT ステートメントが 代わりに作成され、データベースに送信されます。ジョイン(INNER JOIN)の型はジョインエンポーネント上のaddresses テーブルの前に ある 内部結合 M アイコイことと表示されています。ジョインモードの変更で説明される通りジョインの型を LEFT OUTER JOIN M, に変 更することもできます。ジョインモードの変更はこのサンプルの出力に影響を与えないてとご注意してくたさい。

ステップ4 で追加された SQL W HERE /ORDER コンポーネトにより、レコード セトを (自宅、 おけば勤務先の注所を抽出するなど) フィルターし並べ替えることができます。W HERE 句が型 string の デメーター : address_type を作成したこと 注意してくたさい。 を作成すること ご注意してくたさい い デメーターを使用することにより、マンピングがら注所の種類 (自宅、 おけば勤務先)を与えることができま す。 SQL W HERE /ORDER の詳細に関しては、 データベースデータのフィルターと並べ替え (SQL W HERE /ORDER) を参照してくだ さい。

最後に、入力コンポーネトによりマピングの実行中に実際の、ラメーター値を提供することができるようてなります。MapForce外でマピングが実行される場合、(例えば、異なるマシン上でMapForce Server によりマピングが実行される場合) 上記のデザイン時刻が無 視されるようこ コマ・ドライン パラメーターとしてマピングのランタイムに入力が与えられる必要があります。詳細に関しては、マピングレンテメ <u>ーターを与える</u>を参照してくたさい。

6.2.4.3 例: 複数のテーブルから CSV レポートを作成する

このサンプルは、CSV フォーマトの単一のレポトにデータを抽出する目的のナダの複数のデータベーステーブルのショインの方法について説明しています。このサンプルで使用されているデータベースは、Nanonull.sqliteで、以下の、なで見つけることができます: くマイドキュメント> ¥Altova Map Force 2021 Map Force Examples ¥Tutorial % このデータベースは(注文、製品、および往所か管まれる架空の会社の情報を含んでします。リレーショナルデータベースでは通常、情報が正規化され、複数のテーブルごか割されます。例えばusers テーブルコよ (姓、名、および電子メールアドレスがらくまれる) ユーザーの個人の情報が保管されます。データベースは、ユーザーにより注文された製品の情報が2つの異なるテーブルご含まれます: (注文の一意のID、および注文の時刻を含む) orders および、注文された製品の人で見つけることができまれる) ordered products。更に、製品の名前は products とう異なるテーブルご保管されます。

このサンプルの目的は、特定の製品を誰が、何個注文したかを明確にするため、異なるテーブルより抽出されるデータをベースコンたナポートの生成です。この目的を達成するココントの手順を行います

- 1. 挿入 メニューから データベースをクリックしください。
- 2. データベース型を選択するようにプロレプトされると、「SQLite」をクリックして、「次へ」をクリックします。
- 3. 上記のNanonull.sqlite データベースを参照し、「接続」をクルクします。
- 4. プロンプトされると、テーブルorderedproducts、orders、products おびusers を選択して、「OK」をクルクしま す。

「戸データベース オブジェクトを挿入
テーブル、ビュー、ストアドプロシージャを選択するか、独自の SQLS トはローカルの mfd ファイルに保存されます。
マノース
▼ ▼ ◆ SELECT ステートメントの追加/編集 ■ ■ ■ main ■ ■ ■ テーブル ■ ■ ■ テーブル ■ ■ ■ ddresses ■ ■ orderedproducts ■ ■ □ □ □ □ orders ■ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
プレビューを表示(P) >>

- 5. ジョインエンポーネントをマンピングに追加し、4つのnodes/rows アイテムを「入力の追加」(回)ボタンをクリックして作成します。
- 6. データベースからの4つのテーブルをジョインコンポーネトの対応する入力アイテムコ接続します。



メモ 別のンナオでは、全てのジョクンされたテーブルお同じ「ルト」テーブルの下でネストされるようごショクンエンポーネトにテーブル orderedproducts を接続することができ、おこ、(同じレベルではなく、下のレベルに存在する)テーブルorders などを接続 することができます。データベースルーションシップの扱いも参照してください。マメピングの結果は、テーブルをジョインした結果と同じ にないます。違いとしては、このサンプルではジョイン条件は手動で作成されており、他のシナナオでは、ジョインの条件が MapForce により自動的に作成されています。ジョイン条件の定義を手動で行わずテーブルをジョインするサンプルに関しては、例 : SQL モード内のジョク・テーブルを参照してください。同じ「ルト」テーブルの下でジョインされたテーブルの他のマメピングは、以 下の、次で検索することができます: <マイドキュメント

>\Altova\MapForce2021\MapForceExamples\DB_Denormalize.mfd。

このサンプルではジョインコンポーネントに接続されているテーブルの順序は以下のとおりです。

- 1. orderedproducts
- 2. orders
- 3. products
- 4. users

この順序は「ジョインの条件の定義」Define Join Condition () かなをクトックした時に、どのように対応する構造が「ジョインの条件の定義」ダイアログボックストで表示されるかま、響を及ぼします。すなわち、最初のテーブル(orderedproducts) はデフォルトで構造 1の下に表示され、そのすく後のテーブル(orders) は、構造 2の下に表示されます。

😢 ジョインの条件の定義		
以下の全ての条件が true の場合のジョインの構造:		条件の追加(A)
	= 📀	<定義されていない> X A
条件を定義するには、各構造からアイテムを選択して比較 構造 1:	較します。	構造 2:
III orderedproducts IV id IV order_id IV product_id IV quantity	*	orders

最初のジョイの条件を定義するコよ 左側のペイン内の order_id アイテムをクリックし、右側のペイン内の id アイテムをクリックします。 orderedproducts.order_id と orders.id. が組み合わされます:

以下の全ての条件が true の場合のジョインの構	告:	条件の追加(A)
orderedproducts.order id	= 📲 orders.id AS id2	X
		T

この時点まででは、2つのテーブルがヨインされまた。3番目のテーブルを含むショインの条件を定義するはよ、右側のペインの上で使用する ことのできるドロップダウノストから希望するテーブルを選択します。左側のペインは、この場合、ジョインエンポーネト上でショインの前に発生 する全てのテーブルが表示されています。例えば、のしていたのテーブルは、ジョインエンポーネト上でproductsの前に発生するため productsを右側で選択すると、左側は、orderedproductsとordersを表示します。テーブルproductsのアイールドを 前のテーブルショインすることができます(この場合、orderedproducts.product_idとproducts.id)。

(🕑 ジョインの条件の定義			×	
	以下の全ての条件が true の場合のジョインの構造:			条件の追加(A)	
	orderedproducts.product_id		= 🛙 orders.id AS id2	X	
	AND D orderedproducts.product_id		= 🛙 🖗 products.id AS id3	×	
				-	
	条件を定義するには、各構造からアイテムを選択して比較しま 構造 1と 2:	ます。	構造 3:		
	📮 🎹 orderedproducts		products	▼ < >	
	🗊 id		[F		
			Improducts I	*	

4番目のテーブル(users)をショインするコよ users テーブルをドロップダウノルトから選択します。フィールド orders.user_idとusers.idを組み合わせることができます。

😢 Define Join Condition		×
Join structures when all of the following conditions are true: IP orderedproducts.order_id AND IP orderedproducts.product_id AND IP orders.user_id	= III orders.id AS id2 = III products.id AS id3 = III users.id AS id4	Add Condition
To define a condition, select an item from each structure to compare for eq Structures 1 to 3:	juality. Structure 4:	~ • •
<pre></pre>	USETS	
< >>	5	>
Show <u>a</u> nnotations Show <u>types</u>	OK	Cancel

必要とされるショインの条件かすべて定義されると、ショインエンポーネントのアイテムをターゲットコンポーネントにマップすることができます。マンピングを完了するコよ CSV コンポーネントを追加し、ショインエンポーネントからのアイテムを下に示されるようにターゲット CSV コンポーネントに接続します(CSV とテキストファイルを参照してくたさい):



上で説明されているマピングは、ジョイノに含まれる4つのテーブルからエントリルされたCSV フォーマナのレポートを作成します:

- 注文のID (orderedproducts テーブルから取得)
- 注文されたアイテムの個数 (orderedproducts テーブルから取得)
- 注文の時刻(orders テーブルら取得)
- 注文された製品の名前 (products テーブルから取得)
- 製品を主文したユーザーの氏名(users テーブルから取得)

このサンプル内のすべてのテーブルはINNER JOIN モードを使用してジョインされています。LEFT OUTER JOIN ベジョインモードを変更 するすっかの詳細は、ジョインモードの変更を参照してください。

6.2.5 データベースデータのフィルターと並べ替え (SQL WHERE/ORDER)

データベースデータをフィルターし、並べ替える場合、SQLWHERE/ORDERコンポーネントを使用します。これにより、データをフィルターするSQLWHERE 句をMapForce グラフィカルユーザーインターフェイスから入力することができます。ませよ、特定のデータベースフィールド によるレコードのセナを昇順および降順に並べ替える場合、ORDER BY 句を指定することもできます。 SQLWHERE/ORDER コパーネントはデータベースマピングコンポーネントに接続されている必要があります。ジョインされたセント、おこ はノコードをフィルターする必要がある場合、SQLWHERE/ORDERをジョインコンポーネントに接続することもできます、次を参照してくだ さい、データベースのデータのジョイン)。

SQLWHERE/ORDER コンポーネントをマピングに追加する

1. 「挿入」メニューから SQL WHERE/ORDER をクリックします。デフォルトでは、SQL WHERE/ORDER エンポーネント は次の構造を持ちます:

Sql-where/order ≥table/field result

 ソースデータベーステーブルを接続、まゴゴンSQLWHERE/ORDERのtable/fieldアイテムのフィールドロ接続します。例 えば、〈マイドキュメント〉¥Altova¥MapForce2021¥MapForceExamples¥フォルダーカらマメピング DB_PhoneList.mfdを開きます。このマメピングでは、SQLWHERE/ORDERは、ソーステーブル「Person」から 「B」文字から始まる姓を持つ全てのレコードをフィルターします。



3. SQLWHERE/ORDER コンポーネントのヘッダーをダブルクトックします。(ませよ 右クトックしてコンテキストメニューからプロパ ティを選択します)。これにより、「SQLWHERE/ORDER プロノティ」ダイアログボックスが開かれます。



4. 上のデキストボックスにSQLWHERE 句を入力します。およ、ORDER BY 句を入力します。上のイメージは DB_PhoneList.mfd マメピング内で定義されているWHERE とORDER BY 句を示しています(この設定については、下 で詳しく説明されています)。 例えば、次を参照してくたさい、WHERE とORDER BY 句の作成。

SQLWHERE/ORDER にデメーターを与える

マピング DB_PhoneList.mfd 内で使用されている SQL W HERE /ORDER コンポーネトは、W HERE 句を次のように定義しています:

Last LIKE :Name

「Last」は接続されたテーブル内のデータベースフィールド名を参照しています。「LIKE」は、SQL演算子です。「:Name」はマピング上に「Name」とう名前の、ラメーターを作成します。

SQLWHERE/ORDER コンポーネント内の、ラメーターは狂意です。マンピングからWHERE 句に値をパマする際に役に立ちます。パラ メーター無しでは、上のWHERE 句は以下のように書くことができます:

Last LIKE "B%"

これは「B」から始まる姓を持つ全ての個人を抽出します。クエリニ更に柔軟性を与えるため、「B%」の代わりコップメーターを追加します。 これこよ「マメングから他の文字を与えることができるよう」ています(例えば、「C」、そして、定数、おけまマメング入力・プメーターを変更 するだけで、「C」から始まる姓を持つ個人を抽出できるよう」ています)。

SQLWHERE/ORDER コパーネトの表示

SQLWHERE/ORDER コパーネトの重要な点は設定内の定義により外観が異なる点です。このようしてSQLWHERE/ORDER コパーネトの振る舞いを直接マピングから確認することができます。例:

	W HERE 句 か定義済みです。
<mark>∲Person</mark> ∳table/field (≫Name	定義済みの、ラメーターが存在するWHERE句。 パラメーター「Name」「table/field」アイテムの 下に表示されています。
₩Table/field Name	定義済みの、ラメーターが存在するWHERE 句。更に、ORDER BY 句が定義済みです。並べ 替えば、A-Z 並べ替えアイエイことは表示されています。

SQLWHERE/ORDER へダーニマナスのカーノルをポイントすると、定義されている異なる句を表示するとことの表示されます。

6.2.5.1 WHERE と ORDER BY 句の作成

SQLWHERE/ORDER コパーネトがマピングに追加されると、コパーネトに接続されるデーダをフィルターする方法を定義することのできるWHERE 条件(句)が必要にないます。WHERE 条件は、MapForce の「SQLWHERE/ORDER プロ・ティ」 ダイアログ ボックスに入力される必要があります。

MapForce からW HERE 条件を作成することは、MapForce 外で同じ SQL 句を作成すること類似しています。対応するデータベースの SQL 言語に適応される構文を使用してくたさい。例えば、演算子、ワイルドカード、サブセンクト、または集計関数を使用すること もできます。マンピングから マすることのできる デメーターを作成し、後にン デメーター名が続くセミコロン文字(:)を入力します。

メモ WHERE 句の作成が終わると、「OK」をクリックします。MapForce は最後のSQL ステートメートの整合性を検証します。構 文エラーが存在するとダイアログボックスがプロンプトを表示します。

以下の演算子をWHERE コンポーネントにて使用することができます:

演算子	説明
=	等価
\$	異なる
<	未満
>	を超える
>=	以上
<=	以下
IN	指定した値をカラムから検索
LIKE	指定したとやしを検索
BETWEEN	指定した範囲を検索

%(パーセイ)ワイルドカードを使用して、パターン内の文字数を示します。例えば、lastnameとう名前のフィールドからr」で終わる記録を全て抽出します。

lastname = "%r"

XML データベースデータの保管とフェレをサポートするデータベース(例えば、IBM DB2、Oracle、SQL Server)をフェレする場合、その 特定のデータベースに適用することのできるXML 関数をキーワードに使用することができます。例

xmlexists('\$c/Client/Address[zip>"55116"]' passing USER.CLIENTS.CONTACTINFO AS
"c")

次も参照してください。例 IBM DB2 XML 型列からデータを抽出する。

任意で特定のフィールドにや抽出されオンコードセオを並べ替えるコよ「SQLWHERE/ORDER プロ・ディ」ダイアログボックスの対応 するテキストボックス CORDER BY 句を追加します。複数のフィールドを並べ替えるコよ、フィールド名をエンマで区切ります。並べ替えの順 序を変更するコよ ASC とDESC キーワードを使用します。例えば、次のORDER BY 句は、lastname とfirstname によ降順 に順序付けられオンコードを抽出します。

lastname, firstname DESC

サンプル1

次のWHERE 条件は altova.mdb データベースコンポーネトのPerson テーブルに添付されています。これはFirst とLast が 文字「C」 お大き、レコードを抽出します。すなオな、「Callaby」 以降の名前をすべて抽出します。 SELECT 文により First ならびに Last 内にあるコンテンンがC とう文字より大きなレコードが取得されます。 altova.mdb サンプルファイルの Person テーブルにある Last を例にすると、 Callaby 以降のレコードが取得されます。

First > "C" AND Last > "C"

どのようは接続が作成されているかに注目してくたさい

• 問い合わせを行いたいテーブル(この場合は「Person」)からtable/field、テメーターへの接続を作成します。

• result パラメーターを、問い合わせ/フィルタレグを行オ・フィールドの「親」アイテム(この場合はPerson アイテム) へ接続します。First ならびにLast のフィールドがターゲトコンポーネト に接続され、マメピングの結果に表示されます。



サンプル2

このWHERE ステートメントにより、SQLWHERE/ORDER エンポーネントの、ラメーターとして Name が作成されます。この場合、「Last」 カラムこて、ターンの検索が行われます。 ワイルドカードの%により、任意の数の文字が指定されます。



定数 コンポーネント %S は Name / デメーターの植を与えます。ワイルドカード % は文字数を表示します。これにより、マンピングが列 「Last」 内の ダーン(「S」 て終わる名前)を検索します。

サンプル3

このWHERE 句により、PhoneUpperとPhoneLowerとら2 つのテメーターが作成され、PhoneExtの値が比較されます。上限の値と下限の値は以下のマメピングに示される通り、定数コンポーネントにより与えられます。



この例にあるWHERE 句では BETWEEN 演算子を使用することできます

PhoneExt BETWEEN : PhoneUpper and : PhoneLower

6.2.6 仮想テーブルとしての SQL SELECT ステートメント

MapForce ではデータベースエンポーネントにてい ラメーターを伴う SQL SELECT ステート メントの作成がサポートされています。 SELECT ステート メントはテーブル型の構造をしており、SELECT ステート メントにより生成された結果のフィーリド が含まれます。このような 構造は、通常のデータベーステーブルベビューと同様につ マッピングデータノースとして使用することができます。

- SELECT ステートメントにてINNER/OUTER JOIN が使用されている場合、全てのテーブルのフィールドがセンポーネントには 収められます。
- (SQLの"AS" キーワードには指定された)相関名を伴う表現もコンポーネント内のマッピングアイテムとして表示されます。
- SELECT ステートメートは既存のデータベーステーブルを MapForce へ挿入する際に作成されます。
- SELECT ステートメートには、SQL WHERE/ORDER コンポーネントで使用される構文による、デメーターを含めることができます。

入力パラメーダを含まれ、限り、SQL SELECT ステートメントはC++、C#、Java、およびBUILTHN 言語内でサポートされます。 入力パラメーターを持つSQL SELECT ステートメントはBUILTHN 変換言語内でのみサポートされます。

SELECT がデータベースコンポーネントに追加されると、返されるフィーリドはマンピングのために使用することができます。例

🖯 altova-products	
∋⊞ <mark>∭ Orders</mark>	- 8
SELECT_Statement	SELECT *, (Quantity*UnitPrice) AS FROM Orders INNER JOIN Products
Price OrderID	d L L
Orders.ProductID OrderDate	C) C) C)
Shipping Quantity	1 1 1
Products.Product ProductName	iD (č
UnitPrice	1 1 1

SELECT ステートメトはコンポーネト内部に表示されます。ステートメトにより表示される結果の行数は、メニューコマド「ツール オプション」を選択し、全般タブニで指定することができます。

6.2.6.1 SELECT ステートメントの作成

データベースコンポーネントを含むマンピングでSELECT ステートメントを作成することができます。マンピングがデータベースをおと含んでしない 場合、データベースを先に追加します(データベースへ接続するを参照)。このサンプルのスコープのナンガニ、メニューコマンド「挿入 | データ ベースを挿入」を選択して、〈マイドキュメント〉¥Altova¥MapForce2021¥MapForceExamples¥Tutorial¥フォルダー内で使 用することのできる altova-products.mdb ファイルコンケザードに従い接続します。

SELECT ステートメントの作成:

1. データベースエンポーネントのタイトルを右クリックして、「追加/削除/データベースオブジェクトの編集」を選択します。(代わり に、データベースエンポーネントを選択してメニューコマンド「コンポーネント | 追加/削除/データベースオブジェクトの編集」を選 択します。)



- 2. 以下を行います:
 - 既存のテーブルからSELECT ステートメントを生成するコは、テーブルを右クリックして、コンテキストメニューから、SQL ステートメントの生成と追加を選択します。生成されたステートメントは後に編集することができます。
 - カスタム SELECT ステートメートを作成するづま、SELECT ステートメントの追加/編集 ボタンをクリックします。
- 3. 必要に応じて、ステートメーを編集ませば作成します。例: 下のSELECT ステートメートは、<マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial、フォルダー内のaltova-products.mdb ファイルにとり 有効です。 Price フィールドは、Quantity とUnitPrice の2つのフィールドで構成されており、関連付けられた名前(AS Price とて)宣言されます。

```
SELECT *, (Quantity*UnitPrice) AS Price
From Orders
INNER JOIN Products
ON Orders.ProductID = Products.ProductID
Where Orders.Quantity > 2
```

💑 SQL SELECT ステートメントを入力

ソースデータベースオブジェクトを使用するために単一の SQL SELECT ステートメントを 入力し、正しいルート オブジェクトを選択するように注意してください。 Ctrl+space キーを使うことで自動補完を起動することができます。

SELECT *, (Quantity*UnitPrice) AS Price From Orders INNER JOIN Products ON Orders.ProductID = Products.ProductID Where Orders.Quantity > 2

- 4. 「SELECT ステートメントの追加」をクリックします。SELECT ステートメントは、テーブル、ビュー、プロシージャと同様にデータ ベースオブジェクトとして、表示されます。
- 5. 「OK」をクルクします。データベースコンポーネントにSELECTステートメントが表示され、SELECTクロルにお返されたフィールドのデータをマップすることができます。

ell altova-products	
}⊞ III Orders	ľ
SELECT_Statement	SELECT *, (Quantity*UnitPrice) AS FROM Orders INNER JOIN Products
Price	Ľ
OrderID	1
Orders.ProductID	1
🔲 OrderDate	1
🔲 Shipping	1
Quantity	1
Products.Product	ID (
ProductName	1
UnitPrice	1
Unitln Stock	C.

重要な注意点

- SELECT ステートメント内のすべての計算された式は、マップすることのできるアイテムとて、このサンプルでは "AS Price" のような)一意の関連付けられた名前を持つ必要があります。
- JDBC を使用して、Oracle おはIBM DB2 データベースに接続する場合、SELECT ステートメートでは最後のセミコロン は存在しません。

SELECT ステートメントを削除する

- 1. データベースコンポーネントのタイトルを右クリックして、「追加/削除/データベースオブジェクトの編集」を選択します。
- 2. 削除する SELECT ステートメートを右クトックして、「SELECT ステートメントの削除」を選択します。

6.2.6.2 例:パラメーター付きの SELECT

このサンプルは、MicrosoftAccess データベースからデータを読み込みCSV ファイルに書き込むMapForce マピングの作成方法に ついて説明します。特に、この例のマピングは、パラメーターを持つカスタムデータベースSELECT クロルを使用しています。SELECT ス テート メント は複数のテーブルからのデータを結合します。そして、更なる処理のために結果はマピングへ与えられます。

このサンプルは、以下の、なで見つけることのできるマレングデザイン(.mfd)が伴います: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\select-component.mfd。このサンプルファイルを開き、最初に 分析する、または以下のステップを最初から行います。

このサンプルはMicrosoftAccess データベースを使用しますが、その他のデータベース方でも同じ方法で作動します。<u>データベースへ接続</u> するを参照してくたさい。 目的は、以下のとおりです

- 1. データベースから、オーダーされるアイテムの数が、カスタムの値を超えるオーダーのみを選択する必要かあります。このカスタムの値は、マノピングへの、ウメーターとして与えられます。この目的を達成するために、入カノ、ウメーターを持つカスタムデータベース SELECT ステート・メートを作成します。
- 2. Access データベース内では、日時フォーマトは、YYYY-MM-DD HH-MI-SS です。CSV ファイル内では、時刻の部分が 省かれるため、フォーマトは YYYY-MM-DD しよいます。この目的を達成するために、MapForce内で使用することのできる、 <u>date-from-datetime</u> 関数を使用します。
- 3. 結果 CSV ファイルは Orders Report.csv とう名前を持つ必要が ります。

ステップ 1: SELECT 構造を追加する

- 1. メニューからデータベースの挿入をクリックします。
- Microsoft Access (ADO) を選択して、くマイドキュメント
 >\Altova\MapForce2021\MapForceExamples\Tutorial、フォルダー内で使用することのできる altovaproducts.mdb ファイルに接続するウィザードステップに従います。



3. 「データベースオブジェクトの挿入」ダイアログボックスから「SELECT ステート メントの追加/編集」をクトックし、以下のクロを 挿入します:

```
SELECT *, (Quantity * UnitPrice) AS Price
FROM Orders
INNER JOIN Products
ON Orders.ProductID = Products.ProductID
WHERE Orders.Quantity > :Quantity
```

このカエリOrders とProducts テーブルの結合を使用し、すべてのフィールド (*)、と計算された値 (AS Price)。クエルは 以下も指定します: WHERE 句内のQuantity パラメーター

🙀 Enter a SQL SELECT statement	×
Type in a single SQL SELECT statement to use as a source database object. Please make sure to choose the correct root object.	
Auto-completion can be invoked holding Ctrl+space keys.	
SELECT *, (Quantity * UnitPrice) AS Price	
INNER JOIN Products	
ON Orders.ProductID = Products.ProductID	
WHERE Orders.Quantity > :Quantity	
Parameters:	
To create a parameter, enter a '' followed by its name in the statement above.	
Sample: SELECT (fields) FROM (table) WHERE (field)=:myParam	
Darameter	Datature
Quantity	auto-detect
]	~
Boot Object:	\sim
Name: SELECT_Statement	
Hestore SUL	Add SELEUT Statement Cancel

- 4. 「SELECT ステートメントの追加」をクトックします。
- 5. 「OK」をクリックします。altova-products コンポーネトがマピングエアに追加されます。
- 6. altova-products コパーネト 国から パラメータモーと共に呼び出しを挿入 を選択します。

🖯 altova-products		
SELECT_Statement	E S	ELECT *, (Quantity *
		ソースとしてノードを表示(S)
		ターゲットとしてノードを表示(G)
		パラメーターとともに呼び出しを挿入(A)
		SELECT ステートメントの編集(E)

新規の構造(SELECT_ステートメント)をマピングで使用することができるようごなります。これは2 つの パトに分割されます: 左の パトは入力コネクタを、右の パトは出力コネクタを提供します。 左の パトは前に定義された Quantity パラメーターを含んていること 注意してください。

el altova-products				
SELECT_Statement 🔜 SELECT *, (Quantity * UnitPrice)		SELECT_Statement		
INNER JOIN Products	·····	ELECT statement	*	It result SELECT *, (Qi
	l>•	• Quantity		FROM Orders
				Price
				OrderID
			_	🔲 Orders.ProductIDI
			=	🔲 OrderDate 🛛
				Shipping
				Quantity
				Products.Product
				ProductName
			Ŧ	

ステップ 2: 入力/ テメーターの追加

- 1. 「挿入」メニューカジ入力の挿入」をクルクします。
- 2. 型 "Quantity" を名前とます。
- 3. Design-time Execution 内に、デザイン中にマンピングの実行のオーター値を入力します。(このサンプ ルでは、 ²2) 詳細に関しては、マンピングロン ウメーターを与えるを参照してくたさい。

🕑 入力の作り	成
名前(N):	Quantity
デ <i>ー</i> タ型(D):	string 👻
] 入力は必須(I)
ーデザイン時間	実行
▼ 値を指注	定(S)
值(V): 2	
	OK キャンセル

入力パラメーターをデータベースの呼び出しに接続します。構造は、以下のとおりです。



ステップ 3: ターゲナ CSV コンポーネントを追加する

- 1. 「挿入」メニューカジ テキスト ファイル」をクリックします。
- 2. 「標準 CSVのための単純な処理を使用する...」を選択し、「続ける」をクリックします。
- 3. コンポーネト設定ダイアログボックスから、「フィールドの追加」をクリックし、9つの新し、フィールドを追加します。データベースフィ ールドの名前と同じ名前をCSV フィールドに与えることが奨励されています。これは、マッピング接続を作成する際に時間の節約 しないます。これらの設定に関しての詳しい情報は、CSV オプションの設定を参照してください。

Price		OrderID		Product	ID	OrderDa	ate	Shipping		Quantitiy		ProductNa	me	UnitPrice		UnitInSto	ck
string	•	string	•	string	•	string	-	string	•	string	•	string	•	string	•	string	•
																	- P
			_									_					
フィール	/ドをi	≜力o(A)	12	ィールドィ	を挿入	(N)	71-	ルドを削除	(R)	- <<	$\parallel >>$						
			Ľ														

4. SELECT 構造のresult ノードとSV コンポーネトのRows ノード間の接続を作成します。



CSV コンポーネトのフィールドの多くは、データベースと同じ名前を持ち、マンピング接続は、result を Rows に接続すると自動的に作成されます。これが発生すると、「接続」メニューを選択して、「マッチする子を自動的に接続する」オプションが有効化されていることを確認してくたさい。 SELECT 構造内でこの名前を持つフィールドがえい すっか、手動で接続したマンピングアイテムは ProductID のみです。

ステップ 4: 日付の変換

ライブラリウイドウ内で、date-from-datetime 関数を検索し、マンピングエリアイボラッグします。入力と出力下で表示されるようは接続します。



ステップ 5: 出力ファイルの名前の設定

OrdersReport.csvの出力ファイルの名前を設定するコよ、CSVコンポーネント、をダブルクトックし、出力ファイルボックス内に値を入力します。出力ファイルテキストボックス内に値を挿入することのできるコンポーネント設定ダイアログボックスの開かれます。

6.2.7 XML データから / データベースフィールドへのマッピング

MapForce を使用して、XMLを保管するデータベースフィールド(列)からデータをマップ、おオオデータベースフィールドへデータをマップする ことができます。データベースフィールド(列)に保管されるXML データが、抽出され、MapForce によりサポートされる他の構造に書き込ま れること、おオオその逆か可能しています。データを以下のようこマップすることができます:

- 1. XML 型専用のベベ おは がら フィール (例えば SQL Server 内の Xml、Oracle 内の XMLType)。繰り返しおこ は、おは XML へから書き込む専用の XML フィール は、XML をサポートするネイティブのデータベース ご適用することができま す (IBM DB2、Oracle、および SQL Server など)。
- 2. XML コンテンンを保管する"へ" おけま "から" テキストフィールド (例えば、Text、Varchar)。これはデキストフィールドが XML ドキュメントを保管するけっというな長さかあるデータベース ご適用することができます。

しずれこせよ、有効なXMLスキーマは、マップするデータのデータベース列、おけはデータベース列からのために存在する必要があります。デー タベース列がXMLを保存する場合、MapForceは、(データベースがサポートする場合)XMLスキーでを直接データベースから、割り当 てる、おけばれ部ファイルからスキーマを選択するオプションを与えます。各データベース列につき、1つのXMLスキーでを割り当てることができま す。スキーマイン複数のルート要素が存在する場合、スキーマの単一のルート要素を選択することができます。

XML がデータベース内で文字列フィールドとて保存されている場合、XMLドキュメントの文字エンコードは文字列フィールドをベースとて います。データベースフィールドがテキストをUnicodeとて保存していない場合は、文字の一部を表示できない場合があります。 データベースの一部では、XML フィールドのナダのXML エンコード(このエンコードはデータベースの文字のセルと異なる文字のセルを有す る場合があります)をサポートします。データベースによりサポートされている場合はXMLドキュメント エンコード宣言はXML フィールド内で宣 言されているものと解釈されます。データベースにより与えられた XML エンコードサポートに関する詳細は、それぞれのドキュメントを参照してく たさい。

6.2.7.1 XML スキーマをデータベースフィールドに割り当てる

このトピックはデータベース内でXML型として定義されているフィールドにスキーでを割り当てる方法を説明します。下の説明は、SQL Server 2014 とAdventure Works 2014 データベースを使用しています。後者は、GitHub (<u>https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks</u>)のAdventureWorks サンプルからダウ ンロードすることができます。XML フィールドからの、またはXML フィールドへのデータのマンピングは、XML フィールドをサポートする他のデー タベースの型と同様に動作します。

Adventure Works 2014 データベースをマッピングコンポーネントとして追加する

- 挿入 メニューからデータベースをクリング、接続ウィザドを優先順序(ADO おは ODBC)に従い進めます。更に詳しい情報 に関しては、Microsoft SQL Server (ADO) (注接続 とMicrosoft SQL Server (ODBC) (注接続 を参照してください。 メモ SQL Server Native Client ドライバーを使用する場合、統合されたセキュリティ プロパティをスペース文字に設定す る必要があります(SQL Server Data Link プロパティのセットアップを参照)。
- 2. 「データベースオブジェクトの挿入」ダイアログボックスから、Production スキーマを展開し、ProductModel テーブルを選択します。

▼ ▼ ◆ SELECT ステートメントの追加/編集
ProductDescription
ProductDocument
Productinventory
ProductListPriceHistory
······· 🗹 🎹 ProductModel
ProductModeIIIIustration
ProductModelProductDescr
······ D m ProductPhoto
ProductProductPhoto
ProductReview
ProductSubcategory

3. 「OK」をクルクします。

データベーステーブルがマピングエリアに追加されました。このテーブルコン次の2つのXML型のフィールドがあります: CatalogDescriptionとInstructions:



マピング上にXML フィールドの構造が表示されるようにするコよ、フィールドコンテンソのXML スキーマが必要しないます。 Instructions フィールドを右クルクして、コンテキストメニューから、XML スキーマをフィールドに割り当てるを選択します。

sql_serv_2014 [AdventureWorks]	2014]										
Production.ProductModel	Product model classificatic										
ISP ProductModeIID Primary key for ProductModel recort											
Name Product model description.											
CatalogDescription Detaile	ed product catalog informat>										
> Instructions Manufactur	XML スキーマをフィールドに割り当て										
C rowguid ROWGUIDCOL											
> ModifiedDate Date and t	フィールドから XML スキーマの割り当てを解除										
	データベース クエリの実行(Q)										
	接続の削除(D)	•									
	変数コンポーネントの作成(V)	•									
	コンポーネント(C)	∢									

この特定の別の中では、Instructions フィールドに直接データベースからスキーでを割り当てます。これを行うには、データベースオプタンの横のProduction.ManuInstructionsSchemaCollection アイテムを選択し、「OK」をクレックします。
YML スキーマをフィールドに割り当て Image: State
XML スキーマ ◎ データベース(D): Production.ManuInstructionsSchemaCollecti ▼
◎ ファイル(F): 参照(W) 編集(T)
ルート 要素(R): 選択(C)
埋め込みターゲット XML インスタンス ターゲット 名前空間のプレフィックス(P):
✓ 値をターゲットの型にキャスト(無効な出力を書き込むリスクがある時に数値ま たはデータ値のフォーマッティングの保持を無効化)(G)
■データベースのターゲット XMLを検証(V)
エンコーディング名(E): Unicode UTF-8 ▼ バイト オーガー(の): Ub III Tンディアン ▼ ■ バイト オーガー フーDを合める(M)
✓ min/maxOccursをベースにした入力の最適化処理を有効にする(I)
✓ 全ファイルバスを MFD ファイルへの相対バスで保存(S) OK キャンセル バ

XML フィールドの構造がセパーネトトに表示されます。このフィールドへ、おけてのフィールドへ接続を描くれたびマップを作成することができます。



6.2.7.2 例:XML データを SQLite フィールドに書き込む

この例は、複数のXML ファイルからデータを読み込み、 SQLite データベースに書き込む MapForce マピングを作成する必要な手順 について説明します。マピングの目的は、それぞれのノース XML ファイルのために、 SQLite データベース内に新規のデータベースを作成 することです。それぞれのレコードは、 TEXT フィーリドとして XML ドキュメト を保管します。

使用されているサンプルは、以下の、次で見つけることができます: くマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ファイル名は、以下のとおりです:

マメンクデザインファイル	XmlToSqliteField.mfd
ソースXML ファイル	 bookentry1.xml bookentry2.xml bookentry3.xml
検証に使用されるXML スキーマ	• books.xsd
ターゲナ SQLite データベース	library.sqlite

マノピングのゴールを達成するコよ、以下のステップを踏みます

1. XML コンポーネントを追加し、複数のファイルから読み込むように構成します。

- 2. SQLite データベースコンポーネトを追加し、XML スキーマをターゲト TEXT フィールドに割り当てます。
- 3. マソピング接続の作成とデータベース「アクションの挿入」を構成します。

ステップ 1: XML コンポーネントの追加

- 1. 挿入 メニューから XML スキーマ/ファイルをクリックし、 <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ デルノトリ内のbooks.xsd スキーでを参照します。 XML ファイルを提供するようにプロンプトされると、「スキップ」をクリックします。 ルート 要素を選択するようにプロンプトされると、 Books を選択します。
- 2. コンポーネントへッダーをダブルクトックして、入力 XML ファイルボックス内に bookentry*.xml を入力します。これにより、 MapForce にノースディレクトリ内の "bookentry-" から名前か始まるすべての XML ファイルから読み込むようこ命令します。 この技術に関する詳細については、複数の入力おける出力ファイルを動的に処理するを参照してくたさい。

🕑 コンポーネント語	没定		×
コンポーネント名:	books		
スキーマ ファイル(F) pForce2017¥MapFo	rceExamples¥Tutorial¥books.xsd	参照(W)	編集(T)
入力 XML ファイル(I) e2017¥MapForceEx	amples¥Tutorial¥bookentry*.xml	参昭(B)	
e2017#MapForceEx	amples# r donai#bookendy*.xmi	麥照(B)	編果(E)

ステップ 2: SQLite コンポーネントを追加する

挿入メニューから、データベースをクリックし、ウイザードに従い、くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\デルオリのLibrary.sqliteデータベースファイルに接続します (既存のSQLiteデータベースは接続するも参照してくたさい。)。データベースオブジェクトを選択するようよ問われると、BOOKS テーブルを 選択します。

▼ ▼ ◆ SELECT ステートメントの追加/編集... ∃ 🗹 🎛 Library 🕀 🗹 🖯 main └□ 🗹 向 テーブル 『日図 向 ユーザー テーブル Energy 🖬 Books

XML コンテンンが書き込まれるデータベースフィールドは、metadata と呼ばれます。このフィールドにXML スキーマを割り当てるコよ、右クリックして、コンテキストメニューから XML スキーマをフィールドに割り当てる を選択します。



このチュートリアルで、etadata フィールドに割り当てられているスキーマは、ソースXML ファイルを検証するために使用されているものと同じです。「参照」をクトックし、〈マイドキュメント〉私ltova MapForce2021 MapForceExamples ¥Tutorial ¥ ディレクトリから books.xsd スキーマを選択します:

🕑 XML スキーマを	フィールドに割り当て	×
- XML スキーマ		
データベース(D):	•	
● ファイル(F):	2017¥MapForceExamples¥Tutorial¥books.xsd	参照(W) 編集(T)

books.xsd スキーマコよ 次のグロー シレ宣言を持つ2つの要素があります: book とbooks。この別では、book をデータベースに書き込まれるXML のレート要素として設定します。選択をクリックして、book をレート要素として選択します。

🕑 ルート要素の選択		×
名前	名前空間 URI	
book		
books		

ステップ 3: マッピング接続の作成とデータベース「アケションの挿入」の構成 マピング接続の作成 は、以下の通りです:



上に示されるように、ソースとターゲナカ同じスキーマと子要素の名前を使用するため、 book から book への接続は、 ダイでをエピー、 接続です。このような接続に関する詳細については、 すべてエピー接続を参照してくたさい。

ー番上の接続(books からBOOKS)は、ソース内のそれぞれのbook 要素を反復し、新規のレコードをBOOKS テーブルに書き込みます。データベースエレポーネトの A:In ボタンをクリックして、データベースが、下で表示されるように設定を更新するように設定してくたさい

🕑 データベーステーブルのア	クション - BOOKS
最初のレコードの前に実行する。	SQL ステートメント
○なし(N)	
◎ 全てのレコードを削除(L)	📃 全ての子テーブルにある全てのレ:
〇 カスタム SQL(C) :	
- 各レコードに対して実行するアク:	¢∋ン
以下のオペレーションを使用して 比較の結果が全て真であれば、	「全ての入力データは DB デーブルの: 」指定されたアクションが実行されます
入力データ アクション :	全て挿入
💵 id	max() + 1
💷 metadata 👘	マップされた 値

すべてのレコードを削除オプションは、レコードを挿入する前にBOOKS テーブルのすべてのコンテンンを削除するようにMapForceに命令します。

すべて挿入 アクションは、データベース INSERT クロルが行われるように指定します。フィールド metadata は、マソピングから提供される一方、フィールド id は、データベース自身から生成されます。

実行する前にマピングが保存されていることを確認してくたさい。

マピングを実行し、生成された出力を確認するコよ、出力タブをクトックします。このアクションはデータベースをすくに更新しないとと注意してくたさい。生成されたデータベーススクリプトを実行する準備が整うと、メニューコマイ、出力|SQL スクリプトの実行を選択します。
(オコは シールレーボタンをクトックします)。

6.2.7.3 例: IBM DB2 XML 型列からデータを抽出する

このサンプルはXML 型のIBM DB2 データベース列からのデータの抽出方法、および、ターゲト CSV ファイルへの書き込みの方法について説明しています。また、条件に応じて XML コンテンンを抽出するするために、SQL 内に埋め込まれた XQuery ステートメートの使用方法についても説明されています。このサンプルは、テーブルを作成し、テーブルを表示する許可が存在する、IBM DB2 データベースへのアクセスが必要です。

XML デー効実際に含まれるデータベースを準備します。使用中のデータベースに固有のデータベース管理ソール、おけるMapForce で直接おこなうことができます。以下のステップを行ってください

- 1. 新規マピングを作成し、DBQuery タブをクリックします。
- 2. 「クイック接続」(<->) クリックして、ウィザードステップに従い、新規のデータベース接続を作成します(次を参照してください、デ <u>ータベース接続のサンプル</u>)。
- 3. SQL エディターに次のデキストを貼り付けます。このSQL クエリは、ARTICLES とう名前のデータベーステーブルを作成し、データと共に表示します。



4. 「実行」() がったクリックします。クロの実行の結果がついの結果ウイズウム表示されます。クロの実行に成功すると、 4つの行か新規に作成されたテーブルに追加されます。

次に、条件に応じて作成されたARTICLES テーブルカらXML データを抽出するマノピングを作成します。目的は、ARTICLES 列から 100 よりに高額な商品のみを抽出することです。

ステップ1: データベースを追加する

- 1. マピングペイノに切り替えるために「マパング」タブをクリックします。
- 2. 「挿入」メニューから「データベース」をクリックし、ウィザードステップに従い、データベースは接続します。
- 3. データベースオブジェクトを選択するよう プロンプトされると、以前に作成されている ARTICLES テーブルを選択します。



ステップ 2: フィールドにXML 型スキーマを割り当てる

1. コンポーネントのARTICLE アイテムを右クトックし、フィールドにXML 型スキーマを割り当てるをコンテキストメニューから選択します。

Connection1	S.ARTICLES
	Unassign XML Schema from Field
	Query Database
	Delete Connections
	Create Variable Component
	Component •

2. 「ファイル」を選択して、次のスキーマを参照します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\DB2xsd.xsd。

ステップ 3: SQLWHERE /ORDER コンポーネントを追加する

- 1. 「挿入」メニューからSQL WHERE/ORDER をクリックします。
- 2. ARTICLE XML 型の列 SQL W HERE /ORDER の入力は接続します。



3. SQLWHERE/ORDER プロノディダイアログボックス内に次のテキストを入力します:

XMLEXISTS('\$a/Article[SinglePrice>100]' PASSING ARTICLE as "a")



上のデキストは、SQL クエリの WHERE の部分を示しています。マンピングのランタイムでは、ダイアログボックス上で表示されて いる "SELECT" の部分と結合されています。このステートメートは、XMLEXISTS 関数とIBM DB2 データベース、固有の構 文を使用してます。

ステップ 4: ターゲナ CSV ファイルを追加する

- 1. 「挿入」メニューから「テキストファイル」をクリックします。
- 2. プロンプトされると、「標準 CSV のために簡単な処理を使用する...」、「継続する」を選択します。 3. 「フィールドの追加」3回クルクして、商品番号、名前、価格をそれそれ保管する3つのフィールドを追加します。他の設定を保 別ます。
- 4. 下に示されるようにマピング接続を描きます。



マンピングの結果を、出力 タブ。をクリックして、プレビューすることができます。予想通り、出力内に100 よりに高額の価格の商品のみが出力に表示されます。

1	2,Socks,230
2	4, Jacket, 5750

6.2.8 データベースの参照とクエリ

MapForce はオマピング処理から独立しデータベースをクエリすることのできる専用のデータベースクエリ、まけまDB クエリルペインの搭載されています。このような直接的クエリは、マピング *.mfd ファイルはは保存されず、 MapForce から直接参照まけはデータベースのコンテン ツを変更することのできる便利な方法が与えられます。

個別のDB クエリ ペイノは、現在アウティブなマピングのそれぞれご存在します。DB クエリ ペイン内で異なるデータベースへの複数のアク ティブな接続を作成することができます。DB クエリ ペインから作成された接続は、マピングの一部でけばく、 クローバルリソースとして定義し ない限り、MapForceを閉じた後には保管されません。

🖳 altova	 ■ altova 		-	🖽 🕨 👌 🛔 🗠 🗠	🖡 间		
🕲 + 🔽 + 🕲			1	SELECT[Primaryk	(ey],[Fo	reignKey],	_
🕂 🔠 altova ADO			2	[city],[state],	[street] , [zip]	
🕀 📒 altova	_		2	FROM [Address];		_	
□□□ テーフル い	♥ = -70						501 エディタ-
- H M Addr) –))v ess						
-⊞ III Altov	а						
- 🕀 🎹 Depa	rtment						
-⊞ 🛄 Office	2						
⊕ III Perso	on = ⊐tu						
・・・田 🛑 システム	テーノル						
			,				_
							_
PrimaryKey •	ForeignKey	city •	state 🃍	street	•	zip •	
1 1	1	Vereno	CA	119 Oakstreet, Suite 487	6	29213	クエリ結果
2 2	2	Brenton	MA	9865 Millenium Center, S	uite 456	5985	2 2 2 11 2 12
A a a a b a b b b b b b b b b b				行: 2, 列: 6	0.109 秒	16:25:22	
● 取得の完了							
 ➡ 取得の完了 結果 メッセージ 							
 □ 取得の完了 結果 メッセージ マッピング / vsit 		<u>ш</u> л					
 取得の完了 結果 メッセージ マッピング XSLT 	DB 71IJ	出力					
 取得の完了 結果 メッセージ マッピング ×SLT メッセージ 	DB クエリ ジタブ	出力					

このタブのウィンドウは以下のいから構成されます:

- データベースブラウザー にはデータベーステーブルに対する接続情報が表示されます。
- SQL エディターではSQL クロを作成することができます。
- 結果タブ にはカエリの結果がテーブル形式で表示されます。
- メッセージタブ には警告ならび エラーメッセージ か表示されます。

データベースクエリウィボウの上部には接続エイトローリル)用意されており、作業用データベースや接続ならびにデータベーススキーマを定義することができます。

6.2.8.1 データベースの選択と接続

クローオるそれぞれのデータベースに対して、接続を作成する必要があります。マレングが既にデータベースエンポーネントを含む場合、DB クエリ ペインの上のエリアにある既存のデータベース接続デフォルトでは接続はオフライン)を選択することができ、データベースオブジェクトを探 索し、クローを実行することができます。

マンピングがデータベースエンポーネントを含まれ、場合、おけは新規のデータベースは接続する場合は、クイック接続(場)をクリックし、新 規のデータベース接続を作成するウィザードに従します(サンプルを参照)。接続の内のIつがジロー・シルリンースとして定義されている場合、 グロー・シルノノースから既存のデータベース接続を選択することかできます(グロー・シルノノースを参照)。

Altova MapForce 2021 Professional Edition

データベースは接続すると、次のメンバを使用してデータベースクエレを作成することができます。

- SQL ケエを SQL エディターペインから既存の SQL ファイルインポトする。
- ケエをSQLエディターペイン内で作成する。
- データベースブラウザーペイン内のオブジェクトを右クトックして、クロを生成します。(通常、SELECT)。

SQL エディター ペイン内でクロ を実行する準備が整 オ・ム 実行 🔛 ポタンをクリックします。 データベースデータはテーブルの様式で結果タ ブニ表示されます。 ステータスバーは検索の終了したメッセージ(🖸 検索の終了)を、およびクロルの結果に関連する適切な情報を表示 します。

検索の終了メッセージが表示されると、検索結果を検索、並べ替え、クトップボードにニピーすることができます(<u>クエルの結果のニピー、並べ替</u> え、および検索を参照)。

6.2.8.2 SQL ステートメントの作成と編集

SQL エディターは、SQL ステートメートを作成まりは実行するために使用されます。既存のSQL スクレプトからロードされ自動的に生成された、ませる手動で作成されたすべてのSQL ステートメントが表示されます。SQL エディターは自動補完、領域、ませよラインブロックコメントをサポートします、次を参照してください、自動補完、。

🗟 AltovaTarget 🔹 AltovaTarget	・ 国 田 と 造 造 や つ 縁 (6)
AltovaTarget ADO AltovaTarget ADO AltovaTarget F - ブル D D D D D D D	<pre>1 /* Start of script */ 2 SELECT [PrimaryKey], [ForeignKey], [city], [state], [street], [zip] FROM [Address]; 3 SELECT [PrimaryKey], [Name] FROM [Altova]; 4 SELECT [PrimaryKey], [ForeignKey], [Name] FROM [Department]; 5 region 6 SELECT [PrimaryKey], [ForeignKey], [Desc], [EMail], [Established], [Fax], [Name], [Phone] FROM [Office]; 7 endregion 8 SELECT [PrimaryKey], [ForeignKey], [EMail], [First], [Last], [PhoneExt], [Title] FROM [Person]; 9</pre>

SQL エディターソールドーで、以下のアイコンを使用することができます:

	ブラウザーの切り替え: ブラウザーペインの表示/非表示を選択します。
	結果の切り替え: 結果ペインの表示/非表示を選択します。
	クエリの実行(F5):このボターをクトックすることで、現在選択されているSQLステートメントが実行されます。複数のステート メントが表示されていて、そのうちどれも選択されていない場合、全てのステートメントが実行されます。
2	元に戻す: SQL ウイドウで行われた変更を、任意の回数だけ元に戻すことができます。
2	やり直し:それまで元に戻した変更を再度入力することができます。元に戻すコマイドと併用することで、履歴内にある変更を確認することができます。
•	SQL ファイルのインポート: SQL ファイルをSQL エディターにて開き、実行することができます。

â	SQL ファイルのエクスポート: SQL クエを保存して、将来も使用できるようします。
(現在のSQL スクリプトをDatabaseSpy で開く: DatabaseSpy を起動し、SQL エディターウィボウニ表示されてい るスクリプトを開きます。
%	オプション、オプションダイアログボックスを開き、MapForce 全般ならびにSQL エディターは関する設定を変更することができます。

6.2.8.2.1 SQL ステートメントの生成

SQL ステートメントスクリプトからロードされるはデータベースブラウザーから自動的に生成されることができ、おこ手動で入力することもできます。

データベースブラウザーから SQL SELECT ステート メントを生成するには、以下を行います。

データベースブラウザー内のテーブルおけまビューなどの、データベースオブジェクトおけまプォルダーをクリックし、データベースオブジェクトをブラウザーペインから SQL エディター・ベドラックします。



• データベースブランザーからデータベースオブジェクトを右クトックして、「SQL エディターで表示 | 選択 」を選択します。ブランザーペ イノに表示されているデータベースオブジェクトを右クトックして、コンテキストメニューからオプションを選択

SQL ステートメントを手動で作成する

1. SQL エディター内に SQL ステート メントを入力します。自動補完か自動的に生じる場合は、ステート メントの入力中にドロップダ ウンリスト か表示されます。

SELECT *	F
	型名前
	🔋 FROM
	🖫 FOR

2. カーノルの上と下キーを使用して、提案による選択し、ハイライトされたオプションを挿入するために「Enter」を押します(以下 も参照してくたさい <u>SQL 自動補完の提案</u>)。

6.2.8.2.2 SQL ステートメントの実行

SQL エディターに表示される SQL ステートメントは、データベースに対して使用することができ、即時結果を確認することができます。SQL クロル影響を受けた行は「DB クエリ」ペインの内「メッセージ」ペインに表示されます。

複数のSQLステートメイがSQLエディターに表示される場合、選択されたステートメイのみが実行されます。個々のSQLステートメントを以下のように選択することができます。

- 左マウスボタンをクリックして、カーノ川を特定のステートメントードラックします。
- SQL エディター内のライン番号をクリックします。
- 特定のステートメトをトリプルクトックします。

SQL ステート メントの実行:

- 1. SQL エディターにて SQL ステートメートを入力ませる選択します(SQL ステートメートの生成を参照)。
- 2. 「クエリの実行」 12 ボシをクリックします。

6.2.8.2.3 SQL スクリプトのインポートとエクスポート

SQL エディターウイボウニ表示されている SQL を保存することで、スクリプトを将来再利用することができます。

SQL エディターペインに表示されている内容をファイルへ保存する

• 「SQL ファイルのエクスポート」アイコン 2 をクリックして、保存する SQL スクリプトの名前を入力します。

過去に保存した SQL ファイルを開く:

• 「SQL ファイルのインポート」アイコン をクリックして、SQL ウィドウ 表示する SQL ファイルを選択します。

6.2.8.2.4 SQL コメントの追加と削除

SQL エディターでは、ステートメートやステートメートの一部、おゴはステートメートのグループをコメートアナトすることができます。コメートアナトされたステートメートは SQL スクレプトの実行時にスキップされます。

テキストの領域でコメントアウトを行う

- 1. ステートメトおけはステートメートの一部を選択します。
- 2. 選択されたステートメトを右クリックし、「ブロックコメントを挿入/削除」を選択します。

ラインでコメントアナを行う

 コメトアトを行う位置を右クトックして、「行コメントを挿入/削除」を選択します。カーリルカある位置から、その行の最後まで がコメントアトされます。

ブロックコメントまたはラインコメントを削除する

- 1. コメトアナされた箇所を選択します。ラインコメトの削除を行う場合、コメトの前に挿入されているコメントマーク(一) だき 選択するだけでも同様の効果を持ちます。
- 2. 右クトックして、「ブロック(また」は行)コメントを挿入/削除」を選択します。

6.2.8.2.5 ブックマークを使用する

ブックマークを使用することで、スクリプト内にあるアイテムをマークすることができます。

ブックマークを追加する

1

• ブックマークを行いたい、行を右クリックして、コンテキストメニューからブックマークの挿入削除を選択します。

-	
2	<pre>SELECT [PrimaryKey], [ForeignKey], [city], [state],</pre>
	[street], [zip] FROM [Address];
3	SELECT [PrimaryKey], [Name] FROM [Altova];
4 🤇	SELECT [PrimaryKey], [ForeignKey], [Name] FROM
	[Department];
5	SELECT [PrimaryKey], [ForeignKey], [Desc], [EMail],
	[Established], [Fax], [Name], [Phone] FROM [Office];
6	SELECT [PrimaryKey], [ForeignKey], [EMail], [First],
	[Last], [PhoneExt], [Title] FROM [Person];
7	

ブックマークアイコン 💭 がブックマーク余白に表示されます。

ブックマークを削除する

• ブックマークを削除したい行をクリックして、コンテキストメニューからブックマークの「挿入/削除」を選択します。

ブックマーク間のナビゲーションを行う

- 右クトゥイニて表示されるコンテキストメニューから、「次のブックマークへ移動」を選択することで次のブックマークへカーノルを移動します。
- 右クトックにて表示されるコンテキストメニューから、「前のブックマークへ移動」を選択することで前のブックマークへカーノルを移動します。

全てのブックマークを削除する

• 右クトックして「全てのブックマークを削除」を選択します。

6.2.8.2.6 領域の挿入

領域とはテキストの選択範囲のことで、ユニット単位でSQLスクリフトのマーキングや宣言を行うことができます。領域は展開ならびに縮退することができ、SQLスクリフトの一部を隠すことができます。ある領域を、別の領域内部にネストすることもできます。

領域を挿入すると、選択されたテキストの上部に一regionコメントが挿入され、展開/縮退アイコンか表示されます。

メモーーregionコメトの後に領域の説明をしたテキストを挿入することで、領域の名前を変更することができます。 "region" とう名前を変更したいでください(例: ーregion DB2query)。

領域を作成する

- 1. 領域とて宣言したいステートメートをSQL エディターにて選択します。
- 2. 右クトックして表示されるコンテキストメニューから領域の挿入を選択します。マークされた箇所が領域になり、展開ならびに縮退を 行うことができます。

```
1
2
      SELECT [PrimaryKey], [ForeignKey], [city], [state],
      [street], [zip] FROM [Address];
3
     -- region
4
      SELECT [PrimaryKey], [Name] FROM [Altova];
5
      SELECT [PrimaryKey], [ForeignKey], [Name] FROM
      [Department];
6
      -- endregion
7
      SELECT [PrimaryKey], [ForeignKey], [Desc], [EMail],
      [Established], [Fax], [Name], [Phone] FROM [Office];
8
      SELECT [PrimaryKey], [ForeignKey], [EMail], [First],
      [Last], [PhoneExt], [Title] FROM [Person];
9
```

3. + おけまーボックスをクリックして、領域の展開ませば縮退を行うことができます。

領域を削除する

- region ならびこ-endregion コメント
都除します。

6.2.8.3 データベースオブジェクトの参照

1つまけは複数のデータベースは接続している場合、「データベースブラウザー」 ペイノは、各データベース内のテーブル、ビュー、プロシージャ、などの詳細を含むオブジェクトの概要を表示します。 XML をサポートするデータベースでは、「データベースブラウザー」で個別のフォ ルダー内の登録された XML スキーマも追加して表示します。 データベースオブジェイトをナビゲートするコよ「データベースブラウザー」ペイノニ複数の定義済みのデータベース表示レイアトかあります。データベースブラウザーの上の部分で、定義済みのレイアトを使用することができます。

-	• 🕈 🕈 🖸
• •	フォルダー(F)
-	スキーマなし(S)
	フォルダーなし(N)
Flat	フラット(L)
靤	テーブルの依存関係(T)
	田 🋅 システム テーブル

レイアトを選択するには、フォルダーのレイアナ(10 アプダウオタンをクリックして、リストからエトリを選択します。このおシー は選択されたノイアトを変更することに注意してくたさい。

- フォルダーレイアナは、データベースオブジェケトをオブジェケト型をベースコンたフォルダーを階層構造ツレーに整理します。これはデフォルトの設定です。
- スキーマなし レイアナトは フォルダーレイアナトに類似していますが、データベーススキーマは存在しません。ですから、テーブルはデ ータベーススキーマ別に整理されません。
- フォルダーなしレイアナはデータベースオブジェクトをフォルダーを使用せず」、階層構造に表示します。
- フラット レイアウは、データベースオブシェクトを最初の階層構造で型別に表示します。例えば、対応するテーブルに列か含まれるのではなく、すべての列が個別の列フォルダーに表示されます。
- テーブルの依存関係 レイアナ は、テーブルを他のテーブルとの関係に基づき整理します。外部キーを持つテーブル、他のテーブル との関係を持たないテーブルのオンダのカテゴリーが存在します。

レイアナナビゲーションコルえ、「データベースブラウザー」以下のタスクのために使用することができます。

- SQL ステートメトの生成(<u>SQL ステートメトの生成</u>を参照)。
- 表示されるデータベースオブジェクトのフィルターおよび検索(データベースオブジェクトのフィルターおよび検索を参照)。
- "システム"と"ユーザー" テーブルにテーブルを並び替える
- アクティブなデータノースのルトオブジェクトの更新

ユーザーとシステムテーブルにテーブルを並び替える

- 「データベースブラウザー」内のゲーブル フォルダーを右クルクして、ユーザーとシステムテーブルの並べ替えを選択します。
 - メモ この関数は、次のレイアナが選択されていると使用することができます。フォルダー、スキーマなし、おとはフラット。

アクティブなデータソースのルートオブジェクトの更新

• 「データベースブラウザー」の上にある、更新(🧿)をクリックします。

6.2.8.3.1 データベースオブジェクトのフィルタリングと検索

「データベースブラウザー」に表示されるデータベースオブジェクト(スキーマ、テーブル、ビューなど)を名前おけれ名前の一部によりフィルターすることができます。オブジェクトは文字を入力中にフィルターされます。フィルターはデフォルトで大文字と小文字を区別します。フィルターは「No Folders」レイアウトを選択している場合サポートされません。

データベースオブジェクトのフィルタレング

1. ツールトーカジ フォルダーコンテンツのフィルターアイコノン(2) をクリックします。現在のレイアトで表示されているフォルダーの隣にフィルターアイコムが表示されます。



2. フィルタレクを行うフォルダーの隣にあるフィルターアイコンをクリックして、ポップアップメニューからフィルタレクオプション(例えば含 む」)を選択します。



3. 空のフィールドがフィルターアイコンの隣に表示されます。検索を行う文字列(例: "G")を入力します。入力中に検索結果が表示 されます。



データベースオブジェクトの検索

名前によりデータベースアイテムの検索を行うには、フィルター機能の他にもオブジェクトロケーターを使用することができます。

- 1. \vec{J}
- 2. ドロップダウノリストがブラウザーペイン下部に表示されます。検索を行う文字列を入力します(例: "Off")。



3. 「データベースブラウザー」内のノストからオブジェクトをクリックします。

6.2.8.3.2 データベースブラウザー内のコンテキストオプション

データベースブラウザー内で使用することのできるコンテキストメニューオプションは選択されたオブジェクトにお異なります。例えば

- 「ルート」オブジェクトを右クリックすると、データベースを「更新」することができます。
- フォルダーを右クルクすると常に同じオプションの表示されます:「展開 | 兄弟 | 子」と「折りナナみ | 兄弟 | 子」。
- データベースオブジェクトを右クトックすると、下で説明される「SQL エディターで表示」コマンドとサブメニューアイテムが表示されます。

複数のデータベースオブジェクトを選択する場合、「Shift + クリック」または「Ctrl + クリック」を押します。

メモ SQL ステートメートの構文は、使用するデータベースにお異なります。下の文法はMicrosoft SQL Server 2014 に適用することができます。

ルトオブジェクトのナックに「SQL エディターで表示」コンテキストメニューの次のオブノョンを使用することができます。

- **CREATE**: 選択されたデータベースのレートオブジェクトのためにCREATE ステートメートを作成します。例えば CREATE DATABASE [MYDB]
- DROP: 選択されたデータベースのルートオブジェクトのためにDROP ステートメートを作成します。例えば DROP DATABASE [MYDB]

テーブルビューのために「SQL エディターで表示」コンテキストメニューの次のオプタンを使用することができます

- SELECT: ソーステーブルの全ての列からデータを抽出し SELECT ステートメントを作成します。例えば SELECT [DepartmentID], [Name], [GroupName], [ModifiedDate] FROM [MYDB]. [HumanResources].[Department]
- Name: 選択されたテーブルの名前を返します。
- Path: テーブルのフル 次を以下のフォーマトで返します: ¥
 DataSourceName.DatabaseName.SchemaName.TableName.

複数のテーブルを選択した場合、名前おゴコンマで区切られ、異なるラインに表示されます。

列のために「SQL エディターで表示」コンテキストメニューの次のオプションを使用することができます:

- SELECT: 親テーブルの選択された列からデーダを取得する SELECT ステート ベトを作成します。例えば SELECT [DepartmentID] FROM [MYDB].[HumanResources].[Department]
- Name: 選択された列の名前を返します。
- Path: 列のフル 文を以下のフォーマトで返します: DataSourceName.DatabaseName.SchemaName.TableName.ColumnName.

複数の列を選択した場合、名前ませお、マレコンマで区切られ、異なるラインに表示されます。

制約のために「SQL エディターで表示」コンテキストメニューの次のオプションを使用することができます::

- Name: 選択された制約の名前を返します。
- Path: 制約のフル 次を以下のフォーマトで返します:
 DataSourceName.DatabaseName.SchemaName.TableName.ConstraintName.

複数の制約を選択した場合、名前おゴンジはエレマで区切られ、異なるラインに表示されます。

次のオプションを使用することができます。「SQL エディターで表示」コンテキストメニューからインデックス

- Name: 選択されたインデックスの名前を返します。
- Path: インデックスのフルレンを以下のフォーマトで返します: DataSourceName.DatabaseName.SchemaName.TableName.IndexName.

複数のインデックスを選択した場合、名前おゴコンマで区切られ、異なるラインに表示されます。

データベースがXML スキーマをサポートする場合、 "XML スキーマ" フォルダーム表示される全てのスキーマのために次のオプションを使用することができます:

- XMLSpy 内のビュー: XMLSpy 内でデータベーススキーマを開きます。
- XML スキーマの管理:新規データベース XML スキーマを追加し、既存のデータベース XML スキーマを削除することのできるダ イアログボックスを開きます。

6.2.8.4 クエリの結果のコピー、並べ替え、および検索

DB クエリペインの結果タブは、データベースクエリの結果として抽出されたシュードセルを表示します。

	PrimaryKey •	ForeignKey 🔹	EMail •	First •	Last •	
1	1	1	v.callaby@nanonull.com	Vernon	Callaby	
2	2	1	f.further@nanonull.com	Frank	Further	
3	3	1	I.matise@nanonull.com	Loby	Matise	
4	4	2	j.firstbread@nanonull.com	Joe	Firstbread	
5	5	2	s.sanna@nanonull.com	Susi	Sanna	
6	6	3	f.landis@nanonull.com	Fred	Landis	
7	7	3	m.landis@nanonull.com	Michelle	Butler	
8	8	3	t.little@nanonull.com	Ted	Little	

ッシール・ーにより、結果とSQL ステートメトの間をナビゲートすることができ、クロル結果内の検索を促進します。

	検索:結果ウイドウニある文字列を検索します。F3 キーを押下することで検索を継続することができます。
1	ステートメントへ移動: SQL エディターウィンドウへ移動し、その結果を得るために使用された SQL ステートメントのグループが ノイライトされます。

クエリの結果からセルを選択する

- 列ヘッダーをクリックして、全体の列を選択します。
- 列全体を選択するためこ、列番号をクリックします。
- 個別のセルをクルクします。複数の選択を行うことができる「Ctrl」キーを押します。列ませるセルがXMLデータを含む場合、このデータモニピーすることができます。

メモ コンテキストメニュー「選択 | 行 | 列 | すべて」 もデータの選択に使用する子ができます。

クリップボードに選択されたセルをコピーする

• 右クトックして、コンテキストメニューから「選択されたセルをコピーする」を選択します。

データの並べ替え

- 並べ替えるためこ列を右クルクレ、「並べ替え | 昇順 おっよ降順」を選択します。
- 列のヘッダー内の並べ替えアイコンをクリックします。

	ID	• NAME
1	3227	KElla Kir

データは並べ替えられた列のコンテンソニとり並べ替えられます。

デフォルトの並べ替え順序の復元

• テーブル内を右クトックし、コンテキストメニューから「並べ替え | デフォルトの復元」を選択します。

6.2.8.5 実行されたクエリの状態を確認する

DB クエリペインのメッセージ タブは、最後に実行された SQL ステートメントに関する特定の情報を提供し、また、エラーおけは警告メッセージを報告します。

🔻 🔹 🔺 🐔 🐔 🖍 🗄	dia 1	×
Q 2 row(s) affected A sector description of the sector des		
Elapsed time: 0.016 sec		
Results Messages		

メッセージ タブのカスタマイズされた異なるフィルターを使用するか、おけよ 「次へ」および「前へ」 ポタンを使用して、データを行から行へ と参照することができます。 上の代タンは、メッセージをナビゲート、クリップボードへコピー、メッセージの一部を非表示するけっかに使用されます。 これらのオプションは、メッセージタブ内を右クリックすると表示されるコンテキストメニュー内でも使用することができます。

	フィルター: 個別のメッセージのペト (概要、成功、警告、エラー) を開いた箇所からポップアップメニューを表示のために開き ます。「自動挿入」は、 SQL ステートメント、ませは SQL 構造が SQL エディターにより自動的に挿入されると挿入されるメッ セージを指します。「進行状況」メッセージは、データベース接続の結果、および SQL 解析とデータ構造のロードの結果もを報 告します。
	「全てチェックする」 おさま「チェックを全て解除する」 をポップアップメニューから選択し、マウスをクトックするささてこれらのオ プションを全て選択する、おさは選択しない ことかできます。
	次へ、次のメッセージニ移動してノイライトします。
	前へ前のメッセージン移動していイライトします。
陷	選択されたメッセージをクリップボードにコピーします。
臣	選択されたメッセージとその子をクリップボードにコピーします。
錮	すべてのメッセージをクリップボードにコピーします。
#\$	検索:検索ダイアログボックスを開きます。
₩ ₂	前を検索:検索ダイアログボックス内で指定された文字列の前の発生まで移動します。
dig	次を検索:検索ダイアログボックス内で指定された文字列の次の発生まで移動します。



6.2.8.6 データベースクエリ設定

このセクションでは、入力されたSQLステートメイ、おゴコロードされたSQLエディターに適用することのできるその他の設定の構成についての情報と、クロが実行された後に表示されるクロリの結果について説明されています。

6.2.8.6.1 SQL ファイルエンコードの設定

作成されたSQL ファイルませはSQL エディターで開かれたファイルのためのエンコードオプションの指定方法は、以下の通りです:

- 1. 「DB クエリ」 タブをクトックします。
- 2. ペイの上はある「オプション」 2 をりかんて、「全般 | エンコード」をりかんます。

☺ 全般: エンコード	ページ デフォルトにリ
新規 SQLファイルのデフォルト エンコーディング Unicode UTF-8 ▼ ◎ リトルエンディアン バイト オーダー ○ ビッグエンディアン バイト オーダー	BOM (バイト・オーダー・マーク) ◉ UTF-8 でない場合、常に BOM を作成 ◎ 保存時に検出した BOM を保持
デフォルト エンコーディング Unicode UTF-8	

新規のSQLファイルのためのデフォルトのエンコーディング

新規ファイルで使用するデフォルトのエンコーディングを指定することで、新たっ作成されるドキュメントには、そのエンコーディング指定が追加さ れます。2 ませま4 バイトのエンコーディング(UTF-16、UCS-2、ませまUCS-4)が選択された場合、リトルエンディアンませまビッグエンディ アンを選択することができます。

既存のファイルで使用されているエンコーディングは、勿論そのまま保持されます。

未知のエンコードを使用して SQL ファイルを開く

エノコードの指定がない、まけはエノコードを検出することのできないSQLファイルを開く場合使用するエノコードを選択することができます。

メモ エノコード指定のないSQL ファイルは、UTF-8 エノコードで保存されます。

6.2.8.6.2 SQL エディターの一般的な設定

SQL エディターに適用することのできる全般的な設定を以下のように変更することができます。

- 1. 「DB ケエリ」 タブをクトックします。
- 2. ペインの上にある、「オプション」 2 をクリックして、「SQL エディター」をクリックします。

鹵 SQL エディター		ページ デフォルトにリ
< 全般		
☑ 構文の色分けを有効化	📃 実行時にデータ ソースに打	妾続
取得		
☑ タイムアウトダイアログを表示	実行タイムアウト(秒):	60
	バッファー量(行数):	200
入力ヘルパー バッファー 	入力ヘルパー バッファーは自動補完な れ、投入のためにある程度の時間を。	と自動挿入で使用さ 必要とします。 バッファーをクリア
圆 テキスト ビューの設定		

全般

「構文の色分けを有効化」により、SQL構文における要素が異なる色により表示されるよう」ないます。

「タイムアウトダイアログを表示」チェックボックスにより、SQLファイルが実行される際ならびにデータノースが接続されて以内場合、対応するデータノースへの接続が自動的に行われます。

取得

SQLの実行が許可される最大時間(実行タイムアナ)を秒単位で定義します。

「タイムアウトダイアログを表示」チェックボックスを有効にすることで、タイムアウトが発生した際にタイムアウトの設定を変更することができるようにないます。

入力ヘルレーバッファー

SQL ステートメトの入力中の自動補完の提案を有効化するコよ「自動的に開く」チェックボックスを選択します(次も参照する 自動 補完)。 自動補完のための入力ヘルレー、シファーをデータノースは接続する際、おけは最初に使用する際に入力することができます。バッファーの入力には時間がかかる可能性があります。「バッファーのクリア」ボタンを使用して、バッファーをルセルすることができます。

テキストビュー設定

余白やタブ、視覚的な補助やテキストビューナビゲーションのホットキーといたテキストビューの設定を定義することができます。

6.2.8.6.3 SQL ステートメント生成の設定

SQL ステート 火トの生成構文を複数のデータベースの型のために指定することができます。

- 1. 「DB クエリ」 タブをクトックします。
- 2. ペインの上にある「オプション」 2 をクリックして、「SQL エディター」をクリックします。

 SQL エディター: 生成 ステートメント生成オプション データベースを選択: MS Access MS SQL Server Oracle Sybase IBM DB2 IBM DB2 IBM iSeries MySQL PostgreSQL Informix 全てのデータベースに適用 	ページ デフォルトに! ▼ ステートメントの終わりにセミコロンを追加 ▼ 識別子をエスケープ文字で囲む ▼ 列全てのリストに対する SELECT ステートメントを生成
- 確認メッセージ	SQL生成が適用しない場合警告を表示

特定のデータベースのための構文の優先順序を定義するコよリストから選択して、右の8つのチェックボックスを有効化ませま無効化します。

すべてのデータベースのナダの一意の構文を定義するコよ、全てのデータベースに適用を選択して、右の3つのチェックボックスを有効化または無効化します。全てのデータベースのナダの共通の設定は、JDBC 接続を使用した Oracle とIBM DB2 とiSeries データベース 内のデーダを編集する際、不都合か生じる可能性かあることに注意してくたさい。

6.2.8.6.4 クエリの結果ビューの設定

DB クエリペインの結果タブの外観の構成の方法は、以下の通りです:

- 1. DB クエリタブをクトックします。
- 2. ペインの上の オプション 2 をクリックし、「SQL エディター | 結果ビュー」をクリックします。

■ SQL エディター: 結果ビュ	_		[/	ページ デフォルトにり.
● ▼ グリッドを異なる色で表示				
表示オプション マ 水平線を表示 ・ 重直線を表示	✓ 行番号を表 ✓ 結果ツール/	示 (~を表示	▼ソートの方向	を表示
データの編集 ▼ 変更を保存するためにトランサ ▼ 失敗時にトランザクションを	ドクションを使用 をロールバック	■ デフォルトイ ▼ ータの修	直でセルを初期化 正が制限されているとき	をにヒントを表示。

色を変更してグリッドを表示するを選択して結果タブを単純なグリッド表示にするか、または色を使った行の表示を行うか指定することができます。

表示オプショングループでは水平ならびに垂直のグリメド線、行番号や結果ツールレーなどの表示を指定することができます。対応するチェックボックスを有効/無効にすることで、これらの表示/非表示を切り替えることができます。

データの編集グループでは、トランザクションに関する設定や、デフォルトの値でセルを埋めるのか、おエントの表示を行うかといた設定を行うことができます。

6.2.8.6.5 SQL エディターフォントの設定

SQL エディターに表示されるSQL ステートメトの色とフォントの設定を構成する方法は、以下の通りです。

- 1. 「DB クエリ」 タブをクトックします。____
- 2. ペインの上にある「オプション」 a をクリックして、「SQL エディター | フォント」をクリックします。



フォント設定リストボックスに表示されているフォント設定はSQL ステートメントの要素になります。SQL エディターに表示されるフォントフェイスやスタイル、サイズ、おこ色の指定を行うことができます。同一のフォントサイズが全てのテキストに対して使用されます。

個々のテキストタイプごとは指定することのできる設定はスタイルだけです。この機能によい構文による色分しか可能になります。「ページデフォールトにリセット」ボタンをクリックすることで、オルジナルの設定に復元することができます。

6.2.9 ストアドプロシージャ

ストアドプロシージャとはデータベースサー・イー上で動作するプログラムのことで、ケライア・トアプレケーションから呼び出されること」ないます。通常これらのプログラムはSQLを拡張した方言により記述され、データベースによっては、Javaや.NET CLRといオンプログラング言語による実装をサポートしているものもみります。

ストアドプロシージャの典型的な使用方法には、データベースのケエルビデび出し元のケライアントにデータを返す、おけは入力パラメーターの追加検証の後にデータベースを変更するなどか挙げれます。ストアドプロシージャは電子メールの送信などデータベース外で他のアケションを実行することができます。

ストアドプロシージャイは、ゼロ、まけはコーン以上の入力と出力パラメーターをデフォルトの戻い値に加え持つことができます。この結果、 MapForce 内では、ストアドプロシージャを複数の方法で呼び出すことができます。

- マビング上のノースコンポーネントとしてデーダを抽出するオータリニストアドプロシージャを呼び出します。入力、ラメーターを取らない プロシージャのオータリのみ適用することができます。マビングが実行されると、プロシージャが呼び出され、レコードセナ、まけは出力パラメーターが返されます。レコードセナ、まけは出力、ラメーター、まけは両方をMapForce によりサポートされるデータ型にマップすることができます。例えば、ストアドプロシージャとデータノースを参照してください。
- ・ パラメーターを持つ関数としてストアドプロシージャを呼び出します。この場合、マンピングから必要とされるすべての入力パラメーターを 提供し、MapForceによりサポートされる他のターゲナトに返されたシュードセナ、おけは出力パラメーターをマップすることができます。例えば、入力と出力を持つストアドプロシージャを参照してくたさい。
- マメピング上のターゲトコンポーネトとしてストアドプロシージャを呼び出します。一般的なユースケースはデータベースを変更するためコン ちメーターを持つストアドプロシージャを呼び出す場合です(例えば、レコードの挿入など)。このアプローチはストアドプロシージャから出力を呼び出す必要がない場合に適しています。おここのアプローチによりエラーが発生した場合ロール シクすることのできるデータベーストランザンタン内でストアドプロシージャを実行することができます。例えば、ターゲナムコンポーネント内のストアドプロシージャを照してくたさい。

(最初に挿入して、更新するなど)特定の順序でストアドプロシージャを呼び出す、封コはデータベーステーブル上でアグシュンを実行する必要 かある場合かあります。例えば、ストアドプロシージャの出力・デメーターを他のストアドプロシージャイン なする必要かある場合かあります。まこ は、ストアドプロシージャイこと、返されたデーダをテーブルからのデータと組み合わせる必要かある場合かあります。基 はなるデータベースかデーブ ル間の主/外部キー・リレーションシップを強制しては、場合でも、MapForce内で定義されているローカルリレーションを使用してこのようなアク ションは可能しています。詳細に関しては、ストアドプロシージャとローカルリレーションを参照してくたさい。

メモ MapForce かどのようにストプトプロンージャを実行するか示すために、このチャプターはMicrosoft SQL Server 2016 と 「AdventureW orks 2016」データベースを使用しています。後者は<u>https://github.com/Microsoft/sql-server-</u> samples/releases/tag/adventureworks からダウノロードすることができます。

ポートに関するとモ

- BUILT-IN 実行エンジン内でのみストフドプロシージャを使用することができます。C++、C#、ませよ Java 内でのコード生成は サポートされていません。
- ユーザー定義型、カー祖す型、変数型、多種の "エキゾチッグ" データベース固有のデータ型(配列、形状、CLR型) 通常は入 カ、おけは出力/ デメーター型としてサポートされません。
- プロシージャと関数のオー・トーレード(同じ名前と異なる、ラメーターを持つルーチンの複数の定義)はサポートされません。
- データベースによっては入力パラメーターにデフォルト値を定義することもできますが、この機能は現在サポートされていません。マパンプレンプにて入力パラメーターへの接続が行われなかオーとしても、デフォルト値が使用されることはありません。
- 複数のコードを返すストアドプロシージャは、ドライバーとデータベースAPI(ODBC/ADO/ADO.NET/JDBC)の組み合わせ によってはサポートされます。固定されたナラム構造を持って目じ数のレコードセナを返すプロシージャだけがサポートされます。
- 可能な場合、データベースのシンダーには管理されているネイティブのドライバーの最新のバージョンを使用することが奨励されています。ODBC からADO へのブリッジ、ODBC からJDBC へのブリッジなどのブリッジドライバーの使用を避けてください。
- データターゲトとして呼び出されるホアドプロシージャのオダのデータベーストランザグションを任意で有効化することができます。タ <u>ーゲトコンポーネト内のホアドプロシージャ</u>を参照してくたさい、(入力パラメーターを持たさい)データノースとして呼び出されるス トアドプロシージャ、まけば関数同様に呼び出される(入力と出力を持つ)関数のオカメコよ・ランザグションはサポートされていません。

次のテーブルノストはデータベース固有のサポートメモをリストしています。

データベース	ህポー እድ
Access	 MicrosoftAccess データベース内のストフトプロシージャイコは制限された機能のみを持ち、 MapForce 内ではサポートされません。
DB2	 MapForce にてサポートされる対象:ストアドプロシージャ、スカラー関数、テーブル値関数。 MapForce 内で使用されるデータベース API により読み取ることができない ケムシ DB2 ストアドプロシージャからの戻り値はサポートされていません。 値関数 (RETURNS ROW) はサポートされません。 エラーと計画を読み取る際に、確認されている JDBC ドライバーの問題を避けるために、少なくとも 「IBM_DB2 9.7 Fix Pack 3a" をインストールすることが受励されます。不足する結果セット行を 引き起こす ADO プロ・ディダーに関する問題をフィッグスすることができます。
Firebird	• MapForce にてサポートされる対象: ストアドプロシージャ、テーブル値関数
Informix	• MapForce にてサポートされる対象: ストプドプロシージャ、テーブル値関数。
MariaDB	• MapForce ストアドプロシージャ、スカラー関数によりサポートされています。
MySQL	 MapForce にてサポートされる対象: ストアドプロシージャ、スカラー関数 MySQL バージョン 5.5. 以降は、ストアドプロシージャイン対する完全なサポートを含みます。これ以前のバージョンを使用する場合、MapForceの機能性は制限されます。
Oracle	• MapForce にてサポートされる対象: ストアドプロシージャ、スカラー関数、テーブル値関数。

データベース	ቻኛ ት	
	 Microsoft OLE DB Provider for Oracle の代わりにネイティブな Oracle ドライバーの使用が奨励されます。 Oracle ではREF CURSOR 型の出力バラメーターを使用することで結果セナ が返されます。この仕様はストアドプロシージイマ対して MapForce でサポートされるものの、関数に対してはサポートされません。このさめ、Oracle ストアドプロシージャイマ対して常に固定された数と名前が使用されることではます。 	
PostgreSQL	 MapForce にてサポートされる対象: スカラー関数、row-valued 関数、テーブル値関数。 PostgreSQL では、関数にて定義された任意の出カパラメーターにより、結果セナのカラムか記述されます。この情報はMapForce にて自動的に使用され、情報取得のための実行やレコードセナの入力を手動で行う必要はありません。refcursor 型の、ラメーターはサポートされていません。 	
Progress OpenEdge	• MapForce にてサポートされる対象: ストアドプロシージャ。	
SQL Server	 MapForce にてサポトされる対象: ストアドプロシージャ、スカラー関数、テーブル値関数。 Microsoft OLE DB Provider for SQL Server の代わりに最新のSQL Server Native Client ドライバーを使用することが奨励されます。 SQL Server 2008 にて導入されたくつかのデータ型(datetime2, datetimeoffset) に対して、ADO API は限られたサポートしか提供していません。ADO とSQL Server ネイティ ジケライアントを使用するよかで、新たな期間型にてデータの切り捨て問題に立面した場合、接続 文字列にDataTypeCompatibility=80 とう値を加えるか、ODBC を使用するようにしてくたさい。(datetime2, datetimeoffset)。 SQL Server プロシージイには開発がな型 int null の戻り値・ウメーターが含まれておし、こ れらの値をマンピングに使用することもできます。プロシージイにて RE TURN ステートントが省略さ れた場合、戻り値は0 どいます。 	
SQLite	• SQLite はながプレンージャを使用しません。	
Teradata	 MapForce にてサポートされる対象: ストアドプロシージャ、マクロ。 スカラー関数、集計関数、およびテーブル関数はサポートされません。 既知の問題: Teradata ODBC ドライバーがプロシージャロールの後、出カパラメーターの値を作成することを拒否する問題。 	

6.2.9.1 マッピングにストアドプロシージャを追加する方法

マピングエルア内で、ストアドプロシージャは属するデータベースコンポーネントの一部として表示されます。ストアドプロシージャをデータベースコンポーネント上で可視化するコよ、マピングにデータベースコンポーネントを追加する際に、下に示されるように、明示的に選択する必要があります。このサンプルでは、SQL Server 上で作動する「AdventureW orks」データベースを接続します。命令は他のデータベースの種類
ど類似しています。

Oracle データベースの場合、ストアドプロシージャオスタンドアロン、おけま Oracle パッケージの一部として保管されます。マンピングに両方のカテゴアを追加することができます。パッケージン属するストアドプロシージャまけは関数は下に表示される「データベースオブジェクト」ダイアログボックス対応する、シケージ名の下に表示されます。

マッピングにストアドプロシージャを追加する方法:

- 1. 以下を行います:
 - •「挿入」メニューで、「データベース」をクルクします。
 - •「データベースの挿入」(目))ツールレーボタンをクリックします。
- 2. 「データベースオブシェクトの挿入」ダイアログボックスレオンジー着くまでデータベースウィザードのステップに従ってください。各データベース型に適応することのできる詳細に関しては、データ接続のサンプルを参照してください。

7-7. ピュー、ストアドプロシージルを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに描入してください。ユーザーにより定義された SQL SELECT ステートメント位ローカルの midファイルに保存されます. ソース マ マ ゆ SELECT ステートメントの追加/編集 (リー・ションの追加/編集 (ロ)ー レコードセット構造の追加/編集 日 ● AdventureWorks2016 日 ● Yestow 日 ● Yestow 日 ● Yestow 日 ● Yestow 1 ● Zestow 1 ●	☞ データベース オブジェクトを挿入		×
ソース マ マ キ SELECT ステートメントの追加/編集 (0)… レコードセット構造の追加/編集 □ ● AdventureWorks2016 □ ● AdventureWorks2016 4 □ ● Ad	テーブル、ビュー、ストアドブロシージャを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SQL SELECT ステ mfdファイルに保存されます。		
 ▼ ● SELECT ステートメントの追加/編集. リレーションの追加/編集 □ ■ ■ AdventureWorks2016 □ ■ ● スキーマ /ul>	y-z		
Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016 Image: AdventureWorks2016	▼ 🔽 🗢 SELECT ステートメントの追加/編集_ リレーションの追加/編集 (0)_ レコードセット構造の追加/編集		
	De 🖪 🔢 AdventureWorks2016	<u></u>	
	AdventureWorks2016		
1			
HumanResources HumanResources W HumanResources W Person Poduction W Production W Production W Sales			
	⊕ ₩ manResources		
Image: Second seco	- 🖽 🗹 🏦 INFORMATION_SCHEMA		
Image: Production Image: Purchasing Image: Sales Image: Image: Image: Sales Image: Im	田 🗹 🏂 Person		
 Burchasing Burchasing Jut2- Jut2-(P) Jut2-(P) DK キャンセル 	The Production		
□□□ = 3 saies フレビュー フレビュー(P) □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □			
フレビュー フレビュー(P) フレビュー(P) フレビュー#表示(P) << OK キャンセル		Ψ.	
ブレビュー(P) OK キャンセル			
ブルビュー(P) ブルビュー(P) OK キャンセル			
フレビュー非表示(P) <	7 \nu \mathcal{L}_2 \nu (P)		
ブレビュー非表示(P) <			
フレビュー非表示(P) << OK キャンセル			
フレビュー非表示(P) << OK キャンセル			
<u>フレビュー非表示(P) <<</u> OK キャンセル			
	ブレビュー非表示(P) <<	OK	キャンセル

3. マッピング上で表示されるデータベースオブジェクトの横のチェックボックスを選択し、「OK」を選択します。このサンプルでは、全てのテ ーブル、ビュー、および「Human Resources」スキーマ内で使用することのできるストアドプロシージャが選択されています。

メモ データベースエレポーネントのタイトルバーを右クリックし、、をコンテキストメニューから「データベースオブジェクトを追加/削除編集」を選択して、選択済みのオブジェクトを後に変更することもできます。 データベースユーザーアカウントは、データベース内のストアドプロシージャをビューして実行する権利が必要です。
--

データベースコンポーネントがマンピングに追加されます。
盟アイコンによりストアドプロシージャが識別されていることに注意してください。更に、
テーブル、ビュー、プロシージャイはデータベースコンポーネント内で、アルファベット順に並べ替えられていることに注意してくたさい。

0	AdventureWorks2016 [AdventureWorks2016]		
	HumanResources.Department Lookup table containing the department		
	Im HumanResources.Employee Employee information such as salary, depar		
	Im HumanResources.EmployeeDepartmentHistory Employee department tr		
	III HumanResources.EmployeePayHistory Employee pay history.		
	III HumanResources.JobCandidate Résumés submitted to Human Resource		
	III HumanResources.Shift Work shift lookup table.		
	🗮 HumanResources.uspGetAllEmployees 🗮	¢	
	ImanResources.uspUpdateEmployeeHireInfo 🔠 Updates the Emploit		
	🔠 HumanResources.uspUpdateEmployeeLogin 🔠 Updates the Employed		
	HumanResources.uspUpdateEmployeePersonalInfo	Updates the Er	
₽	■●●HumanResources.vEmployee Employee names and a	ソースとしてノードを表示(S)	
Þ	■ ■ ■ ■ ■ HumanResources.vEmployeeDepartment Returns em	ターゲットとして ノードを表示(の)	
Þ		7 7 7 EUC/ TERM(0)	
₽	■●●HumanResources.vJobCandidate Job candidate nam	パラメーターとともに呼び出しを挿入(A)	
Þ	⊕ ●HumanResources.vJobCandidateEducation Displays		
₽	HumanResources.vJobCandidateEmployment Displa	レコードセット構造の編集(R)	
Þ	⊡ ♥♥INFORMATION_SCHEMA.CHECK_CONSTRAINTS	プロシージャ設定(P)	
	■■INFORMATION_SCHEMA.COLUMNS	P.	

各ストアドプロシージャの横の「コンテキストメニューの表示」 国 ボタイこよりストアドプロシージャかどのようコ呼び出されるか、お日は他の プロシージャイ 理連した設定を構成することができます:

オプション	使用法
ソースとしてノードを表示	データベースからデーダを抽出し、MapForce によりサポートされる(XML、テキスト、EDI などの他のエンポーネントにマップするオカコニ、ノ <i>ラメーター無しのストアドプロシージ</i> をを呼び 出す場合このオプションを選択します。例えば、ストアドプロシージャとデータノースを参照して くたさい。
ノードをターゲットとして表示	ストアドプロシージャの出力を必要としたい個所でデータベースを変更、おけま他の特定のアク ションを実行するストアドプロシージャを呼び出す場合、このオプションを選択します。例えば、 ターゲナンポーネント内のストアドプロシージャを参照してくたさい。
・ ラメーター付きの呼び出しの挿入	MapForce によりサポートされる他のエンポーネントに返されたデータをマップするセンド、パ ラメーターカあるストアドプロシージンを呼び出す場合、このオプションを選択します。 例え ば、 入力と出力パラメータを持つストアドプロシージャを参照してくたさい。
レコードセル構造の編集	レコードセトを返すストアドプロシージャのナめに適用することができます。MapForce が返 されナユコードセトの構造を決定し、マピングを表示するストアドプロシージャを実行するた めこ、このオプションを選択します。 おけよ、デザインタイムにストアドプロシージャを実行しない 場合、レコードセト構造を手動で定義することができます。
プロシージャの設定	(データベースを更新する「ターゲト」として構成されたストアドプロシージャのためにつみ適用することができます。プロシージャを呼び出す前にカスタム SQL クロを実行、おけよデータベーストランザブションを有効化するなど追加プロシージャ設定を構成するためにこのオプションを選択します。

6.2.9.2 データソースとしてのストアドプロシージャ

このサンプルは入力、ラメーターを取らない、データベースからデータを抽出するおけのプロシージンを呼び出す方法を示しています。このシナリ オでは、ストプドプロシージンはマッピングへのノースエンポーネントとしての役割を果たし、MapForce によりサポートされる他のターゲットコン ポーネントによい抽出されるデータをマップすることができます。入力、「ラメーターを持つストプドプロシージンを呼び出す必要がある場合、入力と 出力を持つストプドプロシージンを参照してくたさい。

「AdventureW orks」データベース内にデモのストアドプロシージャを最初に作成しましょう。これを行うには、下のスクリプトをデータベースに対して実行します。Microsoft SQL Server Management Studio のプロリウイドウ、おけまDatabaseSpy (https://www.altova.com/ja/databasespy) などのソールからこれを行うことができます。どちらの場合でも、データベースユーザーアカウントがストアドプロシージャを作成する。トミッションを有していることを確認してください。

```
CREATE PROCEDURE HumanResources.uspGetAllEmployees
AS
SELECT LastName, FirstName, JobTitle, Department
FROM HumanResources.vEmployeeDepartment
```

上のストアドプロシージャイは vEmployeeDepartment ビューから従業員の情報を返します。 次のステップはこのプロシージャから返された データを消費するマッピングを作成する方法を表示しています。

- 1. MapForce からAdventureWorks」データベースは接続し、マルングレントアドプロシージャを追加する方法で説明されている とおり、ストアドプロシージャをマルングに追加します。データベースユーザーアケナナトにストアドプロシージャをビューし実行するパー ミッションが存在することを確認してくたさい。
- 2. ストアドプロシージャの横の「コンテキストメニューの表示」 🖾 ボタンをクリックし、「ソースとしてノードを表示」を選択します。



3. 「コンテキストメニューの表示」 国 ボタンをもう一度クリックし、「レコードセル構造の編集」を選択します。「レコードセルの構造」 ダイアログボックスが表示されます。

😢 レコード セットの構造 - [AdventureWorks2016].[HumanResources].[uspGetAllEmployees]	×
ストアドプロシージャにより、出力パラメーターに加え、MapForce のデータソースとして使用することができる1 レコードセットが返されることがあります。これらレコードセットの構造はプロシージャの実装に依存しており、ラ スのカタログから呼び出すことができません。以下のオプションが利用可能です:	つ以上の データベー
・ ユーザー定義のパラメーターによりプロシージャを呼び出すことで、レコードセットを自動的に抽出する	
入力パラメーターと呼び出しプロシージャの定義(D)	
□ 構造の更新が必要な時には、常にこれらのパラメーターでプロシージャを呼び出す(A)	
 列名とデータ型を入力することで、レコードセットの構造を手動で定義する 	
レコード セット(R):	
	×
名則 型 Nullad ここで新たなレコードセット名を指定。	le
呼び出しによりレコード セットの構造が変化するストアド プロシージャはサポートされないことに注意してくだる	5(10
OK +t	シセル
	배스 며드
ンデン・ディン・ディン・シーン・シーン・シーン・シーン・シーン・シーン・シーン・シーン・シーン・シー	コギセッ す。「レコ

4. 「入力パラメータと呼び出しプロシージャの定義」をクトックし、「OK」をクトックします。「ストアドプロシージャの評価」ダイアログ ボックスが表示されます。

ードセト構造」ダイアログボックス内の「レコードセットの追加」ませま「列の追加」ボタンを使用します。

😢 ストアドプロシージャの評価 - [AdventureWorks2016].[HumanResources].[uspGetAllEmployees]			
 ストット・フロシージャには入力、または入力/出力パラメーターが含まれていません。 実行ボタンをクリックすることでプロシージャを実行し、レコードセットの構造を抽出します。 ☑ トランザクションによりプロシージャを実行し、直ちにロールバックする(U) 警告:プロシージャの依存関係によっては、ストアドプロシージャを実行することでロールバックできない 副作用が発生する可能性があります。 			
副作用が発生する可能性があります。 実行(E) 実行結果ステータス: 未実行			
@uspGetAllEmployees	호 int	1里 <i>未抽出</i>	
,		OK キャンセル	

5. 「実行」をクリックし、「OK」をクリックします。レコードセナ構造("RS1")が「レコードセナの構造」ダイアログボックスとマッピン グ上に表示されます。



 この時点では、抽出されたデー効害き込まれるターゲトコンポーネトを追加することができます。このサンプルでは、データは CSV ファイルに書き込まれます。「挿入」メニューから「テキストファイル」をクリックし、データはCSV コンポーネントをマンピング に追加します。詳細に関しては、CSV とテキストファイルを参照してください。



マピングをプレビューすることができます。「出力」ボタンをクリックして、「出力」ペイン内でマピングの結果を確認します。例

1	Sánchez,Ken,Chief Executive Officer,Executive	
2	Duffy,Terri,Vice President of Engineering,Engineering	
3	Tamburello,Roberto,Engineering Manager,Engineering	
4	Walters, Rob, Senior Tool Designer, Tool Design	
5	Erickson,Gail,Design Engineer,Engineering	
6	Goldberg, Jossef, Design Engineer, Engineering	
7	Miller,Dylan,Research and Development Manager,Research and Development	
マッピング	DB クエリ 出力	
Boost All Employees.mfd		

6.2.9.3 入力と出力を持つストアドプロシージャ

このサンプルは、入力・ウメータを取り、データベースから出力を取得するプロシージャを呼び出す方法を示しています。このシナリオでは、ストアドプロシージャイはWebサービス、ませば関数と同様に呼び出されています。MapForceによりサポートされる他のターゲットコンポーネントによい抽出されたデータをマップすることができます。

「AdventureW orks」データベース内にデモのストアドプロシージャを最初に作成しましよう。これを行うには、下のスクレプトをデータベースに 対して実行します。Microsoft SQL Server Management Studio のクエリウインドウ、おけはDatabaseSpy (<u>https://www.altova.com/ja/databasespy</u>) などのソールからこれを行うことができます。 どちらの場合でも、データベースユーザーアカウン トがストアドプロシージャを作成する ペーミッションを有していることを確認してください。

```
CREATE PROCEDURE Production.uspSearchProducts
  @SearchString nvarchar(50)
  ,@MaxPrice money
  ,@ComparePrice money OUTPUT
AS
BEGIN
  SET NOCOUNT ON
  SELECT pr.[Name], pr.ListPrice FROM [Production].[Product] pr
  WHERE pr.[Name] like @SearchString AND pr.ListPrice < @MaxPrice
  SET @ComparePrice = @MaxPrice
  RETURN @ComparePrice
END</pre>
```

上のストアドプロシージャは製品の情報を含むレコードセナを抽出します。入力として2つの、ラメーターを取ります:製品名 (@SearchString)を持つ文字列、および、製品の価格(@MaxPrice)の最高額。レコードセナとデフォルトの戻り、ラメーターコルえ、 出力、ラメーター(@ComparePrice)も抽出します。

次のステップは、このプロシージャイことし返されたデータを使用するマッピングの作成方法を表示します。

- 1. 「AdventureW orks」データベースを MapForce から接続し、マルングレストアドプロシージャを追加する方法で説明されているとおり、ストアドプロシージャをマルングに追加します。データベースユーザーアカウストがストアドプロシージャをビューして実行する パーミッションの存在することを確認してくけざい。
- 2. ストアドプロシージャの横の「コンテキストメニューの表示」 国 ボタンをクリックし、「パラメーター付きの呼び出しの挿入」を選択します。ストアドプロシージャか 左側に入力パラメーターを表示し、右側は戻り値と出力パラメーターを含むマンピングの個別のコンポーネント内に表示されます。

AdventureWorks2016 [AdventureWork]	Production.uspSearchProducts	
Production.uspSearchProducts 🔜	procedure	🛱 🕪 result
	>>> @SearchString	
2	(>++ @MaxPrice	@ComparePrice >

3. 「コンテキストメニューの表示」 国 ボタンをもう一度クリックし、「レコードセット構造の編集」を選択します。これは、プロシー ジャイニよし返されたシュードセナの構造に関する MapForce 情報を提供するために必要です。「レコードセナの構造」 ダイアロ グボックスが表示されます。

😢 レコード セットの構造 - [AdventureWorks2016].[Production].[uspSearchProducts]	×
ストアドプロシージャにより、出力パラメーターに加え、MapForce のデータソースとして使用すること レコードセットが返されることがあります。これらレコードセットの構造はプロシージャの実装に依存し スのカタログから呼び出すことができません。以下のオプションが利用可能です:	パできる1つ以上の ており、データベー
・ ユーザー定義のパラメーターによりプロシージャを呼び出すことで、レコード セットを自動的に抽	出する
入力パラメーターと呼び出しプロシージャの定義(D)	
□ 構造の更新が必要な時には、常にこれらのパラメーターでプロシージャを呼び出す(A)	
 列名とデータ型を入力することで、レコードセットの構造を手動で定義する 	
レコード セット(R):	
	X
名則 至	Nullable
RS1	
ここで新たな列名を指定。	
- 「胚び申しによりしつードセットの構造が変化オスフトアドプロシージョはサポートされたいことに注意	1.7ください。
**************************************	1001/20010
OK	
	キャンセル
	キャンセル
	キャンセル
デザイン時にストアドプロシージャを呼び出すとプロシージャの実装により副作用が発生する可能性が	キャンセル
デザイン時にてみアドプロシージャを呼び出すとプロシージャの実装により副作用が発生する可能性があ ストアドプロシージャを実行する場合、下記で説明されているとおり、「実行」をクトックしては でくけざい。 イ	キャンセル ります。デザイン時に ドカンドニ レコードセッ

4. 「入力パラメータと呼び出しプロシージャの定義」をクリックします。「ストアドプロシージャの評価」ダイアログボックスが表示されま

ードセト構造」ダイアログボックス内の「レコードセットの追加」ませま「列の追加」ボタンを使用します。

Altova MapForce 2021 Professional Edition

す。
入力パラメーター(I):			
名前	型	值	
@SearchString	nvarchar(100)	%Frame"	-
@MaxPrice	money	500	–
・ 実行ボタンをクリックすることで上の入 「トランザクションによりプロシー 警告:プロシージャの依存関係 副作用が発生する可能性があ 実行(E) 実行結果ステータス: 未実行	、力値によりプロシージャを実行した -ジャを実行し、 直ちにロールバックす &によっては、 ストアド プロシージャを ります。	コード セットの構造を抽出します。 する(U) 実行することでロールバックできな(,
出力パラメーター(0)			
名前	型	值	
@uspSearchProducts	int	未抽出	
@ComparePrice	money	未 抽 出	
		ОК キャン セ	204

- 5. 上記の通り デメータの植を入力し、「実行」をクリックします。
- 6. 「OK」をクリックします。レコードセナ構造("RS1")が「レコードセナの構造」ダイアログボックスとマノピング上に表示されます。
- 7. この時点では、抽出されたデータが書き込まれるターゲトコンポーネントを追加することができます。このサンプルでは、データは CSV ファイルに書き込まれます。「挿入」メニューから「テキストファイル」をクリックし、データはCSV コンポーネントをマンピング に追加します。 CSV とテキストファイルを参照してくたさい、次に、下に示されるようにマンピング接続を描きます。プロシージャの入 カバラメータが定数によい提供されていることに注意してくたさい。詳細に関しては、マンピングに定数を追加するを参照してくたさい。



マピングをプレビューすることができます。「出力」ボタンをクリックして、「出力」ペイン内でマピングの結果を確認します。例

1	500,"LL Mountain Frame - Black, 40",249.79,
2	500, "LL Mountain Frame - Black, 42", 249.79,
3	500, "LL Mountain Frame - Black, 44", 249.79,
4	500, "LL Mountain Frame - Black, 48", 249.79,
5	500, "LL Mountain Frame - Black, 52", 249.79,
6	500,"LL Mountain Frame - Silver, 40",264.05,
7	EAA "II Mountain Ename Silven 42" 264 AE
マッピン	グ DB クエリ 出力
🔁 usp S	earchProducts.mfd*

6.2.9.4 ターゲットコンポーネント内のストアドプロシージャ

このサンプルは、入力・ウメータを取るプロシージャの呼び出しとデータベースの更新の方法にていて説明しています。プロシージャをこの方法で 呼び出すことにより、トランザグションを有効化し、エラーの場合アグションをロールレック、お日コプロシージャの呼び出し前にカスタム SQL ステ ートメントの実行を可能にすることができます。このシナリオは、ストアドプロシージャがMapForce内のターゲトコンポーネントのよう1振る 舞い、返される出力に興味がない場合を想定しています。ノラメーターの、父の方法と、ストアドプロシージャイにより返されたデータをマップする 方法に関しては、入力と出力を持つストアドプロシージャを参照してくたさい。

「AdventureW orks」データベース内にデモのストアドプロシージャを最初に作成しましょう。これを行うには、下のスクリプトをデータベースに対して実行します。Microsoft SQL Server Management Studio のクロリウイドウ、おけはDatabaseSpy (<u>https://www.altova.com/ja/databasespy</u>)などのソールからこれを行うことができます。どちらの場合でも、データベースユーザーアカウン トが文トフドプロシージャを作成するパーミッションを有していることを確認してください。

END

,GETDATE())

上のストアドプロシージャ入力とて2つのプロシージャ(@ModelName, @Inst)を取り、データベースにと生成されたデータと共に、 AdventureWorks データベース:ProductModel テーブルに対応する値を挿入します。

次のステップはこのプロシージャイことし返されるデータを使用するマンピングの作成方法について示しています。

- 1. 「AdventureWorks」データベースをMapForceから接続し、マルシノリニストアドプロシージャを追加する方法で説明されるとおり、ストアドプロシージャをマルシノガに追加します。データベースユーザーアカウストがストアドプロシージャをビューして実行する、トミッションが存在することを確認してくたさい。
- 2. ストアドプロシージャの横の「コンテキストメニューの表示」 国 ボタをクトックし、「ノードをターゲットとして表示」を選択します。 左側に入力・ ラメーターが」ストされてい るマメピング上でストアドプロシージャがターゲットコンポーネトとして表示されます。



- 3. 「コンテキストメニューの表示」 国 ボタンをもう一度クリックし、「プロシージャの設定」を選択します。この任意のステップにより、 ロールレジク可能なトランザジョンの内部でストアドプロシージャを実行することができます。 プロシージャが呼び出される前にカスタ ムSQL ステートメートの実行を追加することができます。
- 4. 「トランザウションの使用」チェックボックスを選択します。

😢 データベース ストアド プロシーシ	ジャ設定 - Production.uspSearchProducts	×
ーストアドプロシージャ呼び出しの ◎ なし(N) ○ カスタム SQL (C):)前に実行する SQL ステートメント	
ーデータベーストランザクションの扱 ✓ トランザクションを使用(T) エラーが発生した場合:	及() rollback top transaction and stop ン	
	OK キャンセル	

メモ このサンプルでは、データベースのトレースがデータベースコンポーネントのレベルで無効化されており、トレースか実行されません。しかしなから、必要な場合、ストアドプロシージャのすっとにデータベースのトレースを有効化することができます。詳細に関しては、データベースのトレースを有効化することができます。詳細に関しては、データベースのトレースとエラーのログを参照してください。

5. データベースに挿入されるデータを提供するノースエンポーネントを追加します。このサンプルでは、ソースデータが定数により提供されますが、MapForceによりサポートされる他のノースエンポーネントは入力としての没割を果たします。詳細に関しては、マンピングプニ定数を追加するを参照してくたさい。



このマピングはデータベースを更新し、他のマピングのように出力を直接プレビューしません。代わりに、「出力」 ポタンをクリックして、データベ ースかどのように変更されるかについてのピートを含む疑似 SQL を表示します。トランザグションを有効化すると、これらはコメントにより示され ます。

7		
8	SET QUOTED_IDENTIFIER ON	
10	begin transaction	
11		
12	<pre>EXECUTE NULL = [Production].[uspAddProductModel] 'Tractor part 4711', 'coupling'</pre>	
13	commit transaction	
15		Ŧ
Mappi	ng DB Query Output	
🔊 uspA	ddProductModel.mfd	×

「出力」ペイン内に表示される疑似 SQL は、実際のトランザクションコマンドを表示せず、(コメントとして) ビトのみを表示します。実際の SQL コマンドは基 になるデータベース API に送信されます。

データベースコマオレてマピングを実行するコよ以下を行います

- 「出力」メニューから SQL スクリプトの実行」をクトックします。
- 「SQL スクリプトの実行」 🥵 ツールレーオタンをクリックします。

ストアドプロシージャと入力の重複

同じストアドプロシージャイニマッピング上の複数のノースからのデータをマップする場合、複数の入力を受け入れるようニストアドプロシージャを複 製します。これを行うコは、コンポーネント上のストアドプロシージャを右クトックし、コンテキストメニューから「入力の複製を追加する」を選択 します。入力の複製を参照してくたさい、マンピング実行される個所で、このようなストアドプロシージャの複製にそれぞれの複製された入力の ためニー度呼び出されます。

各ノラメータは動的な値(ませま「nullable」)であることができなさめ、「入力の複製を追加する」はストアドプロシージャのクラメーターのために無効化されていることに注意してください。

6.2.9.5 ストアドプロシージャとローカルリレーション

ローカルルーションは、基になるデータベースを変更することなくMapForce内で作成することのできるデータベースフィールド間のロジカルな リルーションシップです。ローカルルーションシップの定義を参照してくたさい。ローカルルーションをデータベースフィールドのナメガオナではなく、ストアドプロシージャのナメカニントマとターゲトコンポーネント内で定義することができます。

ソースコンポーネント内で、ローカルルレーションにとり関連したオブシェクトからデータを簡単に読み取れることができます。例えば、データベース テーブルからID を読み取り、関連した情報を取得するために、ID のそれそれからストアドプロシージャを呼び出します。

ターゲトコンポーネント内で、ローカルレーションによ複数の関連したプロシージャが呼び出される階層の順序を定義することができます。 例えば、ID 値を作成するストアドプロシージャを最初に呼び出すことができ、テーブルは関連した情報を入力するものを呼び出すことができ ます。また、ストアドプロシージャとローカルレーション内のテーブルをミックスすることもできます。例えば、他のプロシージャの呼び出しの代わり に直接関したテーブルは挿入することもできます。<u>キーを生成するためにストアドプロシージャを使用する方法</u>を参照してくたさい。

ローカルルーションの作成方法:

- 1. データベースコンポーネントのタイトルレーを右クトック、「データベースオブジェクトを追加/削除/編集」をコンテキストメニューから選択します。「データベースオブジェクトを追加/削除/編集」ダイアログボックスか開かれます。
- 2. 「リレーションの追加/編集」をクリックします。
- 3. 「リレーションの追加」をクリックして、リレーションシップを作成するオブジェクトを選択します。

💁 リレーションの追加/編集			×
このダイアログではデータベースにて宣言されたものカ キー オブジェクトと外部キーオブジェクト、そして少な	ロえて、新たなリレーションを宣言すること くとも1つのデータ タイプの列ペアを選択	ができます。 してください。	各リレーションに対して、主
主キー/一意キーオブジェクト	外部キーオブジェクト		OK
□ [テーフル オフシェクト] ■ [最初にオブジェクト選択]	[テーフル オフジェクト] [最初にオブジェクト選択]		キャンセル
			リレーションを追加
			リレーションを削除
			列の追加
			列の削除
		-	

上記のとおり、「主/一意のキー」オブジェケトと外部キーオブジェケトによりローカルルレーションは構成されています。親子リレーションシップと同様と考えて下さい。マピングエンポーネント上では、主/一意のキーかある(テーブル、ビュー、プロシージャ、などのオブジェケトは親として示され、外部キーのあるオブジェケトは下にネストされ表示されます。例えば、下記のデータベースエンポーネント内では、ローカルルーションはレコード列(RS1.Department)とテーブル列(Department.Name)との間で定義されています。この結果、Department テ

ーブルはマピング上のストアドプロシージャの子として表示されます。このサンプルの詳細は、ソースコンポーネント内のローカルルレーションで詳しく説明されています。

AdventureWorks2016 [AdventureWorks2016]
🕀 🧱 HumanResources.uspGetAllEmployees 🔝 🔅
, @uspGetAllEmployees
□
LastName 🗦
FirstName 🔅
JobTitle 😒
Department >
HumanResources.Department Name
DepartmentID Primary key for Department
Name Name of the department., RS1.Dep[>
GroupName Name of the group to which
ModifiedDate Date and time the record v

次のテーブルはコーカルルーションを定義することのできるすべての可能なフィールドをリストしています(例えば、ストアドプロシージャの出力を データベース列にマッピングするなど)。リルーションシップ内の一部のフィールドは、同じ、まけは互換性のあるデータ型を持つ必要があります。

主/ユニークキー	外部キー
 データベーステーブル おけビューの列 ストアドプロシージャの出力・ラメーターおけま見り値。ストアドプロシージャを参照してくたさい。 ストアドプロシージャイこよ返されたシコードセトの列* ユーザー定義 SELECT ステートメントの列。仮想テーブルしてのSQL SELECT ステートメントを参照してくたさい。 	 データベーステーブル おゴビューの列 ストアドプロシージャの入力/ ラメーター ユーザー定義 SELECT ステートメントの入力/ ラメーター

*(パラメータを持たない) データソースまたは、(入力と出力パラメーターを持つ)関数としてストアドプロシージャが呼び出された場合 適用することができます。セクションのためコンコードセットが使用できるようになるには、レコードセットを取得するためにストアドプロシ ージャを一度実行する必要があります。

6.2.9.6 ソースコンポーネント内のローカルリレーション

このサンプルは、ローカルルレーションシップを使用して、ストアドプロシージャイことし返されたデータを同じデータベース内のテーブルからのデータと組み合わせる方法について説明しています。

前のサンプルで既に行っていない場合、次のスクレスを実行して、「AdventureW orks」データベース内にデモのストアドプロシージャを作成します。これをクロリウイドウMicrosoft SQL Server Management Studio のクロリウイドウ、封まはMapForceの「DB クロリ」 ペインから行うことができます。データベースの参照とクロノを参照してくたさい。 どちらの場合でも、データベースユーザーアカウストがスト アドプロシージャを作成する パーミッションを持っていることを確認してくたさい。

CREATE PROCEDURE HumanResources.uspGetAllEmployees AS

SELECT LastName, FirstName, JobTitle, Department
FROM HumanResources.vEmployeeDepartment

上のストアドプロシージャイは、vEmployeeDepartment ビューから従業員の情報を返します。次のステップはこのプロシージャイによし返されるデータを使用するマメピングの作成方法について説明しています。

- 1. マレングレストアドプロシージャを追加する方法で説明されているとおりMapForce から「AdventureWorks」データベースに 接続します。データベースユーザーアカウトカジトアドプロシージャをビューして実行するパーミッションが存在することを確認してくだ さい。
- 2. データベースオブンエクトを選択するようにプロンプトされると、Department テーブルと uspGetAllEmployees ストアドプロン ージャを選択します。



- 3. みアドプロシージャの横の「コンテキストメニューの表示」 🔤 ボタンをクリックし、「ソースとしてノードを表示」を選択します。
- 4. 「コンテキストメニューの表示」 国 ボタンをもう一度クリックし、「レコード・セット構造の編集」を選択します。「レコード・セット 構造」 ダイアログボックスが表示されます。
- 5. 「入力/ラメータと呼び出しプロシージャの定義」をクリックし、「OK」をクリックします。「ストフドプロシージャの評価」ダイアログ ボックスカ表示されます。

😢 ストアドプロシージャの評価 - [AdventureWorks2016].[HumanResources].[uspGetAllEmployees] 🛛 🗲		
 ○ ストット・フロシージャのお干価 - [Adventurevvorks2010].[Indinankesources].[uspGetAllEmployees] このプロシージャには入力、または入力/出力パラメーターが含まれていません。 実行ボタンをクリックすることでプロシージャを実行し、レコード セットの構造を抽出します。 ○ トランザクションによりプロシージャを実行し、直ちにロールバックする(U) 警告: プロシージャの依存関係によっては、ストアド プロシージャを実行することでロールバックできない 		
実行(E)		
未美行 出力パラメーター(0)		
名前	型	值
@uspGetAllEmployees	int	未抽出
		OK キャンセル

6. 「実行」をクリックし、「OK」をクリックします。レコードセット構造((RS1 () が レコードセットの構造」 ダイアログボックスとマッピン グ上に表示されます。

AdventureWorks2016 [AdventureWorks2016]	
	con
🕀 🧮 HumanResources.uspGetAllEmployees 🔜	ļ
	¢
🗔 🥅 RS1	¢
LastName	¢
III FirstName	¢
III JobTitle	¢
Department	¢

「ローカルルーションシップの定義」

返されたノコードセナのDepartment 列とDepartment テーブルのName 列の間のローカルルレーションシップを定義します。

1. データベースエレポーネトのタイトル、を右クトクレ、「データベースオブジェクトを追加/削除/編集」をエレテキストメニューから選択します。

2. 「リレーションの追加/編集」をクトックし、「リレーションの追加」をクトックします。以下に示されるようにルーションシップを定義します。



3. 「OK」をクリックして、ダイアログボックスを閉じます。Department テーブルはRS1 レコードセナの子になったことに注意してくたとい。



マピングを完了する方法

今しかた作成されたルーションシップにより、テーブルからのデータと結合されたシコードセルからのデータをマップすることができるようこなりました。サンプルのスコープのために、データを次のようにターゲル XML ファイルに書き込むことしましょう

- 1. 「挿入」メニューカら XML スキーマ/ファイル」をクリックし、次のファイルを選択します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\EmployeesWithDetails.xsd。
- 2. XML インスタンスファイルを提供するようコプロンプトされると「スキップ」をクトックします。
- 3. 下に示されるようこマンピングの接続を描きます。



上で示されるマピングはデータベースからターゲト XML ファイルにデータを書き込みます。ソースデータはストアドプロシージャから抽出された データと直接テーブルから抽出されたデータの組み合わせです。マピングは、concat 関数を使用して、ダッシュ文字、とグループ名が後に 続く部署名を含む文字列を作成します。

マピングをプレビューするコは「出力」ボシをクリックして、「出力」ペイン内でマピングの結果を確認します。例



6.2.9.7 キーを生成するためにストアドプロシージャを使用する方法

ローカルルーションを使用して、ストアドプロシージャインは生成されたキー(ID)の一部を挿入する方法について説明されています。

「AdventureW orks」データベース内にデモのストアドプロシージャを最初に作成しましょう。これを行うには、下のスクレプトをデータベースに 対して実行します。Microsoft SQL Server Management Studio のカエリウイドウ、おけまDatabaseSpy (<u>https://www.altova.com/ja/databasespy</u>) などのソールからこれを行うことができます。どちらの場合でも、データベースユーザーアカウン トがストアドプロシージャを作成するパーミッションを有していることを確認してくたさい。

```
CREATE PROCEDURE Production.uspAddProductModelEx
  @ModelName nvarchar(50)
  ,@Inst xml
  ,@ProductModelID int OUTPUT
AS
BEGIN
INSERT INTO [Production].[ProductModel]
            ([Name]
           ,[Instructions]
           ,[rowguid]
           ,[ModifiedDate])
     VALUES
           (@ModelName
           ,@Inst
           ,NEWID()
           ,GETDATE())
   SELECT @ProductModelID = SCOPE IDENTITY()
END
```

上のストアドプロシージャは、入力とて2つのプロシージャ(@ModelName, @Inst)を取り INSERT オペレーションを ProductModel テーブル内で実行します。生成された @ProductModelID を出力パラメータを返します。必要条件はストアドプロシージャインと返された @ProductModelID を ProductModelllustration テーブル内に挿入することです。

次のステップは必要条件を満たすマッピングの作成方法を示しています。

- 1. マレンクレストアドプロシージャを追加する方法で説明されているとおりMapForce から「AdventureW orks」データベースに 接続します。データベースユーザーアカウトカジトアドプロシージャをビューして実行するパーミッションカ存在することを確認してくだ さい。
- 2. データベースオブシェクトを選択するようにプロンプトされると、ProductModelllustration テーブルと uspAddProductModelEx ストアドプロシージャを選択します。

	JAdventureWorks2016 [AdventureWorks2016]	
[→ Improduction.ProductModellIlustration Cross-re	≱
	Production.uspAddProductModelEx 📑	ł
		4

3. ストアドプロシージャの横の「コンテキストメニューの表示」 国 ボタンをクリックし、「ノードをターゲットとして表示」を選択します。 左側に入力・ デメーターがノストされてい るマピング上でストアドプロシージャがターゲットコンポーネトとして表示されます。

AdventureWorks2016 [AdventureWorks2016]

AdventureWorks2016 [AdventureWorks2016]

Production.ProductModelIllustration Cross-re
Production.uspAddProductModelEx

AdventureWorks2016 [AdventureWorks2016]

- 4. 任意で、トランザジョン内でストアドプロシージャを実行するコよ「コンテキストメニューの表示」 国 ポタンをもう一度クリック し、「プロシージャの設定」を選択し、「トランザジションの使用」、チェックボックスを選択します。 ストアドプロシージャのナックトラン ンザジションを定義することは、同じトランザジンョン中にキーの取得とコードの挿入が同時に発生することを保証することができます。
- 5. データベースエンポーネトのタイトルバーを右クトクレ、「データベースオブジェクトを追加/削除/編集」をエンテキストメニューから選択します。
- 6. 「リレーションの追加/編集」をクトックし、「リレーションの追加」をクトックします。以下に示されるようにルーションシップを定義します。

1 リレーションの追加/編集	×
このダイアログではデータベースにて宣言されたもの加えて、新たなリレーションを宣言することができます。 キー オブジェクトと外部キーオブジェクト、そして少なくとも1つのデータ タイプの列ペアを選択してください。	各リレーションに対して、主
主キー/一意キーオブジェクト 外部キーオブジェクト	OK
□ Production.ProductModellin	キャンセル
	リレーションを追加 リレーションを削除
	列の追加
	列の削除
-	

7. 「OK」をクトックして、ダイアログボックスを閉じます。ProductModelllustration テーブルはストアドプロシージャの子とて表示されていることに注意してくたさい。ストアドプロシージャ 出力パラメーター(@ProductModelID) はローカルルレーション内で使用されるインジケーターに表示されます。入力、おけは出力コネクタを持ちません。

AdventureWorks2016 [AdventureWorks2016]
Description Production.ProductModellIlustration Cross-reference table I →
🕞 🧮 Production.uspAddProductModelEx 🔣
> ModelName
>
@ProductModelID
→ ← III Production.ProductModellllustration ProductModellD Cross-reference tat
ProductModelID Primary key. Foreign key to ProductMo
>
ModifiedDate Date and time the record was last updat

8. このサンプルでは、@Inst パラメーターはXMLの型です。 コンポーネント上の@Inst パラメーターを右クリックし、「XML スキ ーマをフィールドに割り当てる」をコンテキストメニューから選択します。

Production.ManuNstructionsSchemaCollection スキーマをデータベースから選択します。ルート要素を選択するよう コプロンパされると、デフォルトの値をそのままし、「OK」をクルクします。データベース XML フィールドへのデータのマメピングの 詳細に関しては、XML データから/データベースノイールドへのマメピングを参照してくたさい。

😢 XML スキーマをフィールドに割り当て				
XML スキーマ ④ データベース(D): ○ ファイル(F):	Production.ManuInstructionsSchemaCollecti	参照(W)	編集(T)	

9. データベースは挿入されるデーダを提供するノースエンポーネトを追加します。このサンプルでは、ソースデータが定数により提供されますが、MapForceによりサポートされる他のノースエンポーネントは入力としての役割を果たします。定数の詳細に関してはマンピングに定数を追加するを参照してくたさい。



このマピングはデータベースを更新し、他のマピングのように出力を直接プレビューしません。代わりに、「出力」 ポタンをクリックして、データベ ースかどのよう 変更されるかについてのビートを含む疑似 SQL を表示します。トランザグションを有効化すると、これらはコメントにより示され るように発生します。

10	begin transaction							
11								
12	<pre>NULL = {{{[Production].[uspAddProductModelEx]}}} ('Tractor part 4711','<ro xmlns="http://schemas.microsoft.com/sqlserver/2004/07/adventure-works/ ProductModelManuInstructions"><location locationid="17"><step>coupling<!--/pre--></step></location></ro </pre>	oot						
	<pre>step>',NULL)</pre>							
13	>>> %@uspAddProductModelEx1%	1.1						
14	>>> %@ProductModelID1%							
15	·							
16	<pre>INSERT INTO [Production].[ProductModelIllustration] ([ProductModelID],</pre>							
	[IllustrationID]) VALUES ('%@ProductModelID1%', 11)							
17								
18	commit transaction							
19		-						
Mappi	ng DB Query Output							
🔊 uspA	ddProductModelEx <	4 Þ ×						

「出力」ペイン内に表示される疑似 SQL は、実際のトランザクションコマンドを表示せず、(コメントとして) ビナのみを表示します。実際の SQL コマンドは基 になるデータベース API に送信されます。

データベースコネルてマピングを実行するコよ、以下を行います

- 「出力」メニューから SQL スクリプトの実行」をクトックします。
- 「SQL スクリプトの実行」 5 ツール ードタンをクトックします。

6.2.10 トランザクションロールバックの処理

データベースを変更するマンピングを実行する場合、MapForce(おけよMapForce Server、おけよMapForceーにより生成されたプロ グラム)ではデータベースに関連したエラーが発生する場合があります。例えば、データベースアカウトには、特定のデータベースアクションを実 行するけうか特権がない、おけは無効なデータを入力、おけは更新することを回避するデータベース制約を実行する特権がない場合がありま す。後者は、列か必須で、マンピングが価を適用しない場合に発生する場合があります。

データベース例外が発生した場合にコールレジクを可能にするコは、マピングがデータベーストランザクションを使用するように構成します。下て説明されるとおり、データベースコンポーネントレベルとテーブルアクションレベルてトランザクションを有効化することができます。

データベースコンポーネントレベルでのトランザクション

データベースコンポーネントレベルでトランザクションを有効化するコよ、以下を行います

1. データベースコンポーネントを右クトックし、コンテキストメニューからプロパティ」を選択します(代わりに、データベースコンポーネントのタイトルレーをダブルクトックします)。これによりデータベースコンポーネント設定ダイアログボックスが開かれます。

データベーストランザクションの扱い	
☑ トランザクションを使用(T)	
エラーが発生した場合:	トップトランザクションをロールバックし停止します ~

2. 「トランザクションの使用」チェックボックスを選択します。

これを行うことにより、データベースコンポーネントにより行われるすべての変更がエラーを発生した場合ロールレックされるトランザクション内に囲まれます。マンピングの実行中にエラーが発生すると、出力はドロップダウンノストから選択されたオプションにより異なります。

- トップトランザクションをロールバックし停止 データベースコンポーネントにより行われるすべての変更を囲むトランザクションはコールレッシンされ、マンピングは停止されます。
- トップトランザクションをロール、シグし継続 上記同様ですが、マピングはロールシグ後も継続されます。(例えば、存在する場合、2番目のターゲトコンポーネントを処理する場合など)。

マピングの一部は複数のデータベースコンポーネントを含んでいる可能性があります。これらは、同じデータベース接続を使用する、おけままっ ナメ異なるデータベースへの接続を使用している場合があります。このようなマピングの出力は、データベースエラーが発生する場合、実行エ ンジンイこよにとないます:

マピングがMapForce で実行される場合マピングの実行時、1つのターゲトコンポーネトカ実行されます。「プレビュー」
 ボタンが有効化されている個所のコンポーネントです。そのコンポーネント内でデータベースエラーか発生し、「トランザグションの使用」チェックボックスが有効化されていると、コンポーネントによる全ての変更がロールレジンされます。この場合、データベースエラーか発生すると、エラーか発生する箇所でロール・シングが発生し、マピングは停止されるか、エラーか発生する箇所でドロップダウノノストで選択されたとおり、次のターゲトコンポーネントの実行を継続します。

データベースコンポーネント設定ダイアログボックスで使用することのできるすべての設定に関数の参照に関しては、データベースコンポーネント 設定を参照してください。

テーブルアケションレベルでのトランザケション

トランザンョン at テーブルアクションレベルでのトランザンションは、個別の、挿入、更新などのテーブルアクションを含みます。このセトアップでは、レコード上で挿入、おけま更新アクションが不足する場合、現在失敗したアクションのみをロール、シグし、他のレコードの処理を続行することができます。テーブルアクションレベルでトランザンションを実行するには、データベーステーブルアクションダイアログボックス「トランザクションの使用」チェックボックスから選択します。(次を参照: データベーステーブルアクション設定).

😢 データベーステーブルアクション - product	ts	×
- 最初のレコードの前に実行する SQL ステー ● なし(N) ○ 全てのレコードを削除(L) ○ カスタム SQL(C) :	-トメント 全ての子テーブルにある全てのレコードも削除	
- 各レコードに対して実行するアクション ここで定義された演算子を使用して、す/ NULL 値を等価として扱うように「NULL ec 全ての比較が true の場合、特定のアクラ	べての入力データはDBテーブルデータと比較されます。 qual」チェックボックスをチェックしてください。 ションが実行されます。	
レコードに対するアクション NULL 等作 「「」 name 「」 quantity 「」 description 「」 created_at 「」 updated_at	 全て挿入 □ 全て挿入 □ DB 生成 □ マップされた値 □ □ □ □ マップされた値 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	ヘ アクションを追加(A) アクションを挿入(I)
子テーブルのデータ削除 入力子データを無視 データベーストランザクションの扱い		へ ↓ アクションを削除(D)
エラーが発生した場合: トップトラン トップトラン トレース トレースレベル(E):	ザクションをロールバックし停止します ザクションをロールバックし停止します ザクションをロールバックし得止します ンザクションをロールバックし得止します ンザクションをロールバックし継続します ンサクションをロールバックし継続します	× 21 701 07
「バルク転送を使用 (MapForce Server バルク転送はデータベース接続によりサポ	r のみ)(B) バッチ サイズ: 1000 レ ペートされていません。]−ド
		OK キャンセル

使用することのできるオプションは以下の通りです

トップトランザクションをロールバックし停止 – 親子リレーションシップを持つマッピングが複数のテーブルを更新する場合、親テーブルのオックガイナではおく、子テーブルのオックコン・サクションを有効化することができます。トランザクションが複数のレベルで有効化することができます。これは、そのレベルでのトランザクションが有効化

されている場合、データベースエンポーネトレベルからかトランザジョンです。それ以外の場合、エンポーネトはは複数のネストされたテーブルアグションな存在することが仮定され、一番上のテーブルアグションで定義されナトランザジョンの取られます。データを複数コンクされているテーブルは挿入するで説明されるとおり、このオジョンを理解するは、エンポーネトが複数のテーブルの階層を更新することができることを思いせえてくたさい。例えば、エンポーネトはは、個人のレコードを挿入する制「挿入」アグション、および、個別のテーブルに各個人のレコードを挿入する」つませは複数のネストされた「挿入」アグションが存在します。両方の挿入アグションのためはトランザジョンを有効化することができ、「トップ」は階層レベルの最も高いレベルを指しています(このサンプルでは、個人がアグションを挿入しています)。このため、アドレスを挿入する際に初外が発生すると、これはこのサンプルでは、個人のテーブルベルであるトップのトランザジョンをローフレッグし、実行は停止されます。すなわち、個人のアドレスを挿入できない場合、個人のレコードは挿入されません。

- トップトランザクションをロールバックし継続する 上記同様ですが、ロールジクの後もマッピングは継続されます(例えば、存在する場合、他のターゲットコンポーネントを処理するために継続されます)。
- 現在のトランザウションをロールバックし停止する 例外が発生すると、このロールバックは、現在のトランザウション内で発生した変更のみをロールバックします。現在のトランザウション外で発生した変更はコミットされます。
- 現在のトランザクションをロールバックし継続する 上記同様ですが、マピングはロールバック後も継続されます。

ストアドプロシージャレベルトランザクション

ストアドプロシージャのすめのデータターゲトと呼ばれるデータベーストランザンションを有効化することができます。例えば、<u>ターゲトコンポーネ</u> ント内のストアドプロシージャ</u>を参照してくたさい。(入力パラメーターを持たない)データノースとして呼び出される、おけば関数のようコ呼び出 される(入力と出力を持つ)ストアドプロシージャのためには、コントザンションはサポートされていません。

ロールジク

トランザションがマピングランタイムご発生するかを確認するコよ「出力」ダブをクリックして、MapForce により生成された疑似-SQL codeコードを確認します。下のイメージは、エンポーネントレベリビテーブルアグタンレベリビトランザンタンが有効化されているマピングのプ レビューを示しています。下に示されているとおり、メインのトランザンタンは各オペレーションのためご発生します(コメント --begin transaction と--commit transaction にお意味的されます)。また、複数の小さ、トランザンタンは各入力オペレーションのた めご発生します(SAVEPOINT コマンドにお意識別されます)。

7 8	begin transaction
10	PRAGMA foreign_keys = ON;
11 12	SAVEPOINT PRODUCTS
13 14	INSERT INTO "products" ("name") VALUES ('Product A')
15 16	SAVEPOINT PRODUCTS
17 18	INSERT INTO "products" ("name") VALUES ('Product B')
19 20	SAVEPOINT PRODUCTS
21	INSERT INTO "products" ("name") VALUES ('Product C')
23	
25	commit cransaction
マッピン	グ DB クエリ

マルングがMapForce Server おけまMapForce により生成されたプログラムによし実行されると、上記のとおり、マルングのデザイン中 に構成されたオプションに基づきロールレジックが実行されます。例えば、オプション「トップトランザクションをロールレジックし停止」を選択する と、他の挿入か成功した場合でも、失敗した入力オペレーションにより全ての変更がロールレジックされます。しかしながら、「現在のトランザク ションをロールレックし継続する」を選択すると、失敗した挿入か最も近し保存ポイトにコールシンされ、マンピングは他のレコードを挿入しようと試みます。

メニューコマボ「出力 | Run SQL-スクリプトの実行」を使用して MapForce 内でトランザブションか有効化されたマッピングを直接実行すると、ダイアログボックスはデータベースの別外が発生したかを通知します。例:

😢 MapForce: データベース トラ	ンザクション例外	\times
トランザクション内のエラー:	PRODUCTS	
エラーの原因:		
NOT NULL constraint failed: pr	oducts.quantity	0
		~
SQLステートメント:		
- どのように続けるか選択		
○このトランザクションをロール	バックして停止	
○このトランザクションをロール	バックして続ける	
○ 今後のエラー全てに対	してロールバックして続ける	
◉ トップをロールバックして停」	Ł	
○ トップをロールバックして継続	売	
	ОК	

以降の振る舞いを変更することができます

- このトランザクションをロールバックし停止する これは、現在のトランザクションに含まれる変更のみをロールバックし、マピングの 実行を停止します。現在のトランザクションの外部で前に行われた変更はエミナトされます。
- このトランザクションをロールバックし継続する 上記同様ですが、マンピングロールバッグ後も実行されます。チェックボックスロールバックして全てのエラーのために継続するを選択して、更にエラーが発生した場合、ダイアログボックスを抑制します。
- すべてをロールバックし停止する-このデータベース呼び出し内のすべてのトランザクションは、ロールバックされ、マンピングの実行 は停止されます。

6.2.10.1 例:トランザクションロールバック

このサンプルは、データベースを変更するマメピングを実行する際にデータベースの例外を処理する可能な方法について説明しています。このサ ンプル内で使用されているデータベースはユーザーレコード(users テーブル)とアドレス(addresses テーブル)を保管します。各ユーザー はゼス、1つ、おけはつのアドレスを持つことができます(例えば、自宅の住所と職場の住所)。具体的けて、アドレス内の各アドレスけて users テーブル内のidフィールドを指す user_id フィールドが含まれています。 users テーブルはこのかめaddresses のかかの「親」 テーブルです。下のノスティングは両方のテーブルのかかの SQL 作成スクリプトを示しています(構文が SQLite データベースに適用できること に注意してくたさい)。

CREATE	TABLE
	users (id INTEGER NOT NULL PRIMARY KEY,
	first name TEXT NOT NULL ,
	last name TEXT NOT NULL,
	email TEXT UNIQUE NOT NULL);
CREATE	TABLE
	addresses (id INTEGER NOT NULL PRIMARY KEY,
	user id INTEGER NOT NULL,
	is shipping INTEGER,
	is billing INTEGER,
	type TEXT NOT NULL,
	city TEXT NOT NULL,
	street TEXT NOT NULL,
	number INTEGER NOT NULL,
	FORFICN KEY (user id) REFERENCES users (id)) .

ソーステーブル

必要条件はusers とaddresses テーブルからのすべてのデータを新規のテーブルすなわち new_users とnew_addresses テー ブルニピーします。これらのテーブルはusers とaddresses テーブルに類似しています。唯一の違いはテーブルnew_addresses は 下のコードリスト内でいくライトされている通り列is_shipping とis_billing のよりに値を許可しません。これは、値が発生すると 例外が発生することを意味するよったれば重要です。

CREATE	TABLE new_users (id INTEGER NOT NULL PRIMARY KEY, first_name TEXT NOT NULL, last_name TEXT NOT NULL, email TEXT UNIQUE NOT NULL);
CREATE	TABLE new_addresses (id INTEGER NOT NULL PRIMARY KEY, user_id INTEGER NOT NULL, is_shipping INTEGER NOT NULL, is_billing INTEGER NOT NULL, type TEXT NOT NULL, city TEXT NOT NULL, street TEXT NOT NULL, number INTEGER NOT NULL, FOREIGN KEY (user id) REFERENCES new users (id)) :

ターゲットテーブル

古しテーブルが新しいテーブルに全てのデータをエピーするマンピングは以下に示されています。このマンピングを次の、次で見つけることができます。< マイドキュメント > #Altova #Map Force 2021 #Map Force Examples #Tutorial #Database Exceptions.mfd。



上で表示されている通り、このマンピングはデータをそのまま同じデータベースに属するソースデータベースエンポーネントからアーゲルデータベース コンポーネントにコピーします。唯一の違いは、ソースとターゲントテーブルが異なる点です。ターゲントコンポーネントのタイトルドーをダブルクシッ クすると、各入力アウンコンの前に全てのレコードセントが削除されていることに注目してください。これにより、ターゲントテーブルが各挿入の前に 空であることを保証することができます。

😢 データベーステーブルアクション - new_users	×
最初のレコードの前に実行する SQL ステートメント ○なし(N) ●全てのレコードを削除(L)	カレコードも削除

上記の通り、ソースaddresses テーブルこnull 値か含まれていない場合、例外が発生します。以下の通りテーブルこnull 値か含まれて いないかをチェックすることができます:

- 1. 「DB クエリ」 タブをクトックします。
- 2. addresses テーブルの横のボタンを右クトックし、「SQL エディター| SELECT」内で表示を選択します。
- 3. 「クエリの実行」 と をクリックします。

ExceptionsDemo 💌 main						•		> č	à	n വ 👫	
P. → 🗛 🕈			1	SE	LECT "i	ł",	"user	_id", "is	_shipping",		
ExceptionsDemo SQLite				"i	s_billi	י", "מי	"typ mher"	e", "city FROM	<i>"</i> ,		
Ē	📒 ma	in _				"n	ain"."a	ddre	sses"	;	
i		テーブル	7.0		2						
		address	/ <i>//</i>								
	··. (±	mew_ado	dresses								
	÷	🔟 new_use	ers								
	 ₽	III users									
I AS	L				_						
1 "0				1	_						
	id •	user_id •	is_shipping •	is_billing	•	type 🔹	city	•	street	•	number •
1	1	1	1	0		work	Bridgede	II	Maple	Lane	1
2	2	1	[NULL]	1		home	Bridgede	II	Olive	Street	6
3	3	3	1	[NULL]		home	Roseford		Everg	reen Lane	34
4	4	4	1	1		work	Beardale		Route	44	9
5	5	5	[NULL]	1		home	Johnson	City	Frank	lin Avenue	11
C)取得の	完了						行:5	, 列: 8	0.062 秒	15:32:00
精	果 メッ	セージ									

上で表示されている通り、ソースaddresses テーブル内に複数のNULL フィールドが存在します。この点を考慮して、マピングを実行する際に、例外を扱うよの複数のオプィンを使用することができます。例:

- A)例外が発生すると、全ての変更がロールバンだれます。すなわち、ソースデータに無効なレコードが含まれているとコードが挿入されません。
- B) 例外が発生すると、例外が発生する箇所でしコードをスキップし、有効なしコードを挿入し続けます。

シナオ A: 例外の場合に全ての変更をロール シクする

必要条件:例外が発生するとデータベースへ加えられた全ての変更をロールックします。この目的を達成するためにマンピングを構成する方法は以下のとおりです:

- 1. ターゲトデータベースエンポーネトのタイトル・モモカクリックし、コンテキストメニューカら「プロパティ」を選択します。ませま、タイトル・モをダブルクリックします。これにより、データベースエンポーネント設定ダイアログボックスが開かれます。
- 2. 「トランザクションの使用」チェックボックスを選択し、「トップトランザクションをロールバックして停止します」を選択します。

マピングが上記のように構成されていると、例外の発生は、データベースコンポーネントレベルでのトップレベルのトランザクションのロール・シグを引き起こします。これらの設定を使用してマピングを実行するココンドを行います:

1. 「出力」ダブをクリックします。

2. 出力メニューから「SQL-Script の実行」をクリックします。ソースアドレステーブル内のnull 値のため、このステージでは、マパンプは別外に遭遇します。データベーストランザグション例外 ダイアログボックスか表示されます。

😢 MapForce: データベース トランザクション例外	\times
トランザクション内のエラー:	
エラーの原因:	
NOT NULL constraint failed: new_addresses.is_shipping	~
	~
SQLステートメント:	
INSERT INTO "new_addresses" ("user_id", "id", "is_shipping", "is_billing", "type", "city", "street", "number") VALUES (?, ?, ?, ?, ?, ?, ?)	
- どのように続けるか選択	
 ●このトランザクションをロールバックして停止 	
○ このトランザクションをロールバックして続ける	
○ 今後のエラー全てに対してロールバックして続ける	
OK	

3. デフォルトの選択済みのオプションを変更せず、「OK」をクリックします。

結果は以下のようないます

- 全ての変更がロール シクされます。
- new_users テーブル内にコードは挿入されません。
- new_addresses テーブル内はコードは挿入されません。

シナオB)「現在のトランザクションをロールレックし、継続する」を実行する

必要条件: 例外を生成するレコードをスキップし、有効なレコードを挿入し続ける。この目的を達成するためにマッピングを構成する方法は以下のとおりです:

- 1. new_users テーブルの横の DEL All ポタンをクトックし、「トランザクションの使用」 チェックボックスを選択し、「現在のトランザクションをロール ックし、継続する」を選択します。
- 2. new_addresses テーブルの横の All ボタンをケルクレ、「トランザクションの使用」 チェックボックスを選択し、「現在のトランザクションをロール ivクし、継続する」を選択します。

マピングをこれらの設定(データベーストランザブション例外ダイアログボックス)で実行します。以下に示されるようにオプタンを選択して、現在のトランザブションをロールバックし、マピングの実行を継続します:

😢 MapForce: データベース トラ	ンザクション例外	\times
トランザクション内のエラー:	NEWADDRESSESUSERID	
エラーの原因:		
NOT NULL constraint failed: no	ew_addresses.is_shipping	0
SQLステートメント:		
INSERT INTO "new_addresses "street", "number") VALUES (?	s" ("user_id", "id", "is_shipping", "is_billing", "type", "city", ?, ?, ?, ?, ?, ?, ?, ?)	
」 - どのように続けるか選択		
○ このトランザクションをロール	バックして停止	
● このトランザクションをロール	いが少して続ける	
□ 今後のエラー全てに対	してロールバックして続ける ,	
	£	
○ トップをロールバックして継続	范	
	ОК	

結果は以下のようてなります

- エラーを伴うトランザクションはすべてロールレックされます。
- (データベース例外によりユーザーレコードが生成されていないかかり5人のユーザーが挿入されます
- (3 つのドレスか値を含み例外か生成されているため 5 つのアドレス中2つのアドレスか挿入されます

シナリオC)「トップトランザクションをロールバックし、継続する」を実行する

必要条件: フドレスの挿入に失敗した場合、親ユーザーレコードも挿入される~きではありません。これを達成するためにマピングを以下のようご構成します:

- 1. データベースコンポーネントのタイトルドーをダブルクトックし、「トランザクションの使用」チェックボックスをクリアします。
- 2. new_users テーブルの横の_DEL, A:In」 ポタンをクトックし、「トランザクションの使用」チェックボックスを選択し、「現在のトランザクションをロールレッシックし、継続する」を選択します。
- 3. new_addresses テーブルの横の All ボタンをクリックし、「トランザクションの使用」 チェックボックスを選択し、「トップトラン ザクションをロールバックし、継続する」を選択します。

この設定でマピングを実行すると、データベーストランザクション例外ダイアログボックスか表示されます。以下に示されるようにオプタンを選択し、プロンプトされる回数とおり選択を繰り返します:

😢 MapForce: データベース トラ	ンザクション例外	×
トランザクション内のエラー:	NEWADDRESSESUSERID	
エラーの原因:		
NOT NULL constraint failed: ne	ew_addresses.is_shipping	Ô
L SQLステートメント:		Ť
INSERT INTO "new_addresses "street", "number") VALUES (?	s" ("user_id", "id", "is_shipping", "is_billing", "type", "city", ?, ?, ?, ?, ?, ?, ?, ?)	
」 - どのように続けるか選択		
	パックして行き上	
- 今後のエラー全てに対	// 1990 Clist (1)る してロールバックして続ける	
○ トップをロールバックして停」	E	
● トップをロールバックして継続	売	
	OK	

結果は以下のようてなります

- 2 人のユーザーが挿入されます(Sharda Junker、Tobie Hughey)
- 1 つのアドレスが挿入されます

説明: ユーザー「Sharda Junker」はアドレスを持たない唯一のユーザーです。このレコードのために例外は発生しません。2番目のユーザ 一である「Tobie Hughey」はこのアドレスレベルで例外が発生していない唯一のユーザーであるため挿入されています。他のユーザーレコー ドは、例外が発生しトランザブションがロールバックされているアドレスを持っているため挿入されていません。

nulls を持たれすめ、例外をスローしない2つのアドレスが存在します。これらはid=1とid=4.を持つアドレスです。最初のアドレスのための挿入トランザブションは しかしながら、ユーザートランザブションがロールレッジンされる必要があっすため、ロールレッジンされています。このため、id=4を持つアドレスのみが挿入されています。

同じ結果を以下のように達成することができます。

- 1. new_users テーブルの横の DEL, A:In ボタンをクトックし、「トランザクションの使用」 チェックボックスを選択し、「現在のトラン ザクションをロールル ジックし、継続する」を選択します。
- 2. new addresses テーブルの横の All ボタンをクリックし、「トランザクションの使用」 チェックボックスをクリアします。

MapForce Server を使用してマピングを実行する方法

ライセンス供与済みのMapForce Server を使用する場合、コマイドライン、Linux、macOS、ませはWindows マシン上でマピン グを以下の通り実行することができます:

- 1. メニューコマナド「ファイル | Map Force Server 拡張ファイルニンパイルする」、を使用して Map Force Server 拡張ファ イル(.mfx) ヘマナングをコッパルにます。 Map Force Server 拡張ファイルニマナングをコッパルするも参照してくたさい。
- 2. .mfx ファイルと Exceptions Demo.sqlite データベースをサゲーマンとのディレオリニピーします。これを「作業ディレオーリ」と呼ぶことします。
- 3. コマボプロンプトを作業ディレクトリニ変更し、以下のコマボでMapForce Server を実行します:

mapforceserver run DatabaseExceptions.mfx

灹

• mapforceserver は使用中のオペーティングシステムのオメのMapForce Server 実行可能ファイルへの 次です。.mfx ファイルへの 次を適用できるように変更します。例えば Linux マシンでは、パスは以下のとおりです

:/opt/Altova/MapForceServer2021/mapforceserver.

サーバーの実行の場合、オンデマイ、おけま再帰的にAPI 呼び出し、おけま low Force Server ジョブとしてマピンがを実行することが てきます。詳細に関しては、MapForce Server を使用して自動化するを参照してください。

6.2.11 データベースのトレースとエラーのログ

マピングがデータボースに書き込む場合、オプションでデータベーストレースとエラーログを有効化することができます。マピングの作動 時データベースへのマピングにより全ての変更をトラックする場合、役に立ちます。トレースがオンの場合、データベース挿入、おけは更新アク ション、おけはエラーがXML ファイル内でログされている場合、これらを後に分析おけよ、更に自動化された方法で処理することができます。 XML 以外の書式でログファイル作成する場合、任意でデータを MapForce によりサポートされるエンポーネントの種類からマップすることが できます(例、テキストファイル、おけは他のデータベース)。

メモマップがトレースとログをサポートするコよマノピング言語がC#、C++、お」ま」ava に設定されている必要があります。

データベースのトレースを構成することは可能です。構成により、すべてのメッセージ、おとはエラーのみをトレースし、エラーのみを、おとなレースを完全に無効化することもできます。更に、次のレベルてトレースを使用することができます。

- データベースコンポーネントレベル。このレベルでは、そのデータベースコンポーネントのナダンコンレースを効果的にオンとオスに切り替えることができます。これは複数のターゲットデータベースコンポーネントを持つマンピングにとい役に立ちます。そして、一部のナダにのアナレースが必要となります。
- テーブルおけはストアドプロシージャレベルレデータベースエンポーネント上では、各テーブル、おけはストアドプロシージャのナムリエレースを行うかを決めることができます。有効化されると、このレベルでのトレースは "Insert All"、 "Update If"、 "Ignore If"、 "Delete If" などのテーブルアグションのナムリこのイベントを含みます。ストアドプロシージャイ 理連した・イベント がトレースされます。
- データベースフィールドレベル。デフォルトで、全てのフィールドがトレースされますが、特定のフィールドをトレースから除外、またココンラー時にのアホレースファイルに含まれるよう、指定することができます。

重要な点は、上記の3つのレベルは階層で、トレース設定は上から下に伝達されます。このため、データベースエポーやホトレベルでトレースを 無効化します。テーブルおけはストアドプロシージャレベルでは有効化することは無理です。同様に、テーブルレベルでトレースを無効化するこ とができますが、データベースフィールド(列)レベルでは、設定することはできません。同様がトレースレベルを狭める際に適用することができま す。例、データベースコンポーやホトレベルでのエラーにのチャトレースを制限する場合、テーブルまけはストアドプロシージャレベルでフルトレース を使用することは不可能です。

データベースコンポーネントレベルでのトレース

データベースコンポーネントレベルでトレースを有効化する方法

1. データベースエレポーネントを右クリックし、「プロパティ」をコンテキストメニューから選択します(おけよ、データベースエレポーネントの タイトルレデーをダブルクリックします)。これにより、データベースエレポーネント設定ダイアログボックスか開かれます。

1	- FD-2		
	トレースレベル (E) :	常に ~	
	ファイルのトレース <mark>(F)</mark> :	log.xml	参照

- 2. 希望するトレースレベル(常に、エラー、無効化済み)を選択します。「エラー」を選択すると、データベースエラーがマピングの 実行時に発生した場合にトレースファイルを作成します。
- 3. 「ファイルのトレース」の横はレースのすべての詳細が書き込まれるXMLファイルの名前、ませま やを書き込みます(例、 Log.xml)。フル ゆの代わりにファイル名を入力すると、トレースファイルがマンピングが作動する同じディレクトリ内に作成されます。

トラッキングを有効化した後、データベースエンポーネントはマッピング上の構造を変更します。具体的はは、データベース構造に加え、階層 内のトップアイテムがトレースファイルであるアイテムの新規のノードの階層が表示されます。詳細に関しては、ファイル構造のトレースを参照し てください。

データベースコンポーネントレベルでのトレースの有効化により、マンピング接続の存在するそのコンポーネント上のすべてのテーブルおけまなア ドプロシージャのかかに有効化されます。必要であれば、下記の通りテーブルおけまなトアドプロシージャレベルでトレースをトラックすることができ ます。マンピング上では役割を果たされ、コンポーネント上に表示されているテーブル、おけまストアドプロシージャイホトレースされません(すなわ ち、マンピング接続は存在しません)。

テーブルまたはストアドプロシージャレベルでのトレース

テーブルませまなトアドプロシージャレベルでのトレースの有効化

- 1. データベースコンポーネトレベリ出のトレースレベリが常におはエラーに設定されていることを確認してくたさい(上記参照)。
- 2. 以下のつを行います:
 - トレースを有効化するテーブルの横のテーブルアケションポタン(例、▲□)をクックします。
 - ストアドプロシージャの場合、「コンテキストメニューの表示」 国 ボタンをクルクし、「プロシージャの設定」をコンテキストメ ニューから選択します。

- トレース			
トレースレベル (E) :	コンボーネント設定の使用	\sim	フィールド <mark>(F)</mark>

3. トレースレベルを選択します。「コンポーネント設定の使用」オプションはコンポーネントレベルで定義された同じ設定を継承しま す。「エラービ制限」オプションはトレースをエラーイベントに制限します。「常に無効化」はエのテーブル、おけよストバプロシー ジャではトレースが発生しないてとを意味します。

データベースフィールドレベル出のトレース

データベースエンポーネトレベルとテーブルまけまなアドプロシージャレベル出のトレースを有効化すると、全てのフィールド(データベース列)は デフォルトでトレースレポートに含まれます。任意で、データベースフィールド、おけまストアドプロシージャ・デメーターイトレースを制限すること ができます。これを行うコま、テーブルアグンョン、おけまデータベースプロシージャ設定ダイアログボックス上のフィールド、ボタンを押します。

ame O O O Jantity O O O scription O O O
antity O O O
escription
reated_at O O
updated_at O O

このダイアログボックス上の選択は、ファイル構造のトレースに影響を与えます。例、隠すフィールドは、トレースファイル上で表示されません。 これを行う必要があるかもしれません。例えばログファイルのサイズを減らす場合などが挙げられます。

複数のフィールドを同時に隠す、おけまきも場合、ヘッダー内にそれぞれのオプィンをクリックしてくたさい。

6.2.11.1 ファイル構造のトレース

データベースコンポーネントのためコンレースか有効化されている場合、トレース構造がエンポーネントの下の部分で使用できるようことがます。 例:



上記の通り、トレース構造のトップノードはマノピングが実行される際に書き込まれるトレースファイルの名前を示しています。トレース構造は マノピング内で発生するデータベーステーブル、おけはストアドプロシージャをモデルとしています。このサンプルでは、データベースの名前は "NanonullTarget"ですので、トレース構造内に同様のアイテムが存在します。

"NanonullTarget"の子アイテムは"products" データベーステーブルに対応する products です。このサンプルでは、データベースエンポ ーネントはコテーブル "products" が つのみ存在します。しかしながら、多くのテーブルのエンポーネント内では、トレース構造はエンポーネント 上に存在する各テーブルを含みます。テーブル、おけコントアドプロシージンをデータベースエンポーネントに追加、おけゴ消除すると、トレース構造が対応して更新されます。命令に閉しては、データベースオブジェントの追加、編集、および、削除を参照してくたさい。

このサンプルでは存在しませんが、trace:init構造も表示される場合があります。これは、マンピングがダインの挿入、更新、おけれ他のアクションの前に初期アウションの構成背れている場合に発生します。例、新規のテーブルを挿入する前にすべての既存のレコードを削除する場合、trace:init構造が存在します。(削除、おけよカスタムSQL クロルだどの)初期アウションのトレース情報が収集されます。

trace:values 構造は、データベースのテーブルのすべての列を表示します(この場合、「products」テーブルの列)。ストアドプロシージャの ために、この構造はストアドプロシージャの、ラメーターを表示します。デフォルトで、全ての列はトレースに設定されており、上記の通りどの列 がトレースされるかを構成することができます。

trace:actions アイテムは、度のアケンコンカデータベーステーブル内で現在設定されているかを表示します。このサンプルでは、データベース 更新アウション A:Up が発生するように設定されており、trace:update アイテムを使用することができます。データベース挿入アウション A:In を実行するようにコンポーネントを変更すると、トレース構造に地のアウションのオンガにtrace:insert に変更されます。ストアドプロシ ージャの場合、trace:execute アウションが代わりに表示されます。 各トレースアグションコン対応するデータベースアグションによ影響を受ける行数を指定する rows-affected 属性が存在します。

trace:error アイテムはマピング実行中にデータベースエラーか発生した場合に作成されます。以下の2つの属性が存在します: code と state。エラーのテキストと属性の値はデータベースドライバーにと、提供され、各データベースの種類にと、異なります。

trace:summary アイテムコよ発生したエラーの数量をレポートする errors 属性が含まれます。エラーが発生しない場合、属性の値は0 です。

6.2.11.2 例: データベースエラーのログ

このサンプルは、SQLite データベースにデータを挿入するためにデータベーストレースとエラーログの構成方法にていて説明されています。命令はMapForce によりサポートされる他のデータベース型、および、("update"、"ignore"、"delete" などの他のデータベースアクション型に類似しています。

次の、なでこのサンプルで参照されているデモマングを見つけることができます: くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\Tutorial\DatabaseExceptions.mfd。このマピングはデータベースの2 つのテーブルからのデータを同じデータベースのテーブルの新規のペアに挿入し、例トランザグションロールレック内で更に詳しく説明されています。データベース内のレコードの一部は例外を生成するためになる的にnullに設定されています。

		_		
0	ExceptionsDemo		🖯 ExceptionsDemo	
ģ⊕ [111 addresses		≩⊞ III new_addresses	₽
¢⊜[11 users	 1	🕀 🛄 new_users 🛛 DEL, A:In	Þ
¢	🗊 id 🛛 👘	ا ا	• 📲 id	¢
¢	III first_name	ا ا	▶ ····· III first_name	¢
¢	Iast_name	ا ا	III last_name	¢
¢	🗊 email	ا ا	email	¢
¢ ⊷	🕀 🛄 addresses user_id	ا ا	← 🕂 🔟 new_addresses 🔥 A:In] user_id	¢
¢	🗊 id 🛛 🖉	ا ا	►	¢
	user_id = users.id	>		¢
Þ	is_shipping	Þi	is_shipping	¢
Þ	🛄 is_billing	Þi	is_billing	¢
Þ	🔲 type	Þi	• TT type	¢
Þ	🔲 city	Þi	city	¢
Þ	🔲 street	Þi	street	¢
¢	🔲 number	<u>ا</u>	number	¢
		d		

DatabaseExceptions.mfd

このサンプルでは、ビジネスの必要条件は生成されるエラー数とエラーが発生する箇所を決定するために、ターゲトデータベースエレポーネトのためによし、レースを有効化することです。下のステップはこの目的を達成する方法を説明しています。

ステップ 1: トランザクションロール シクの有効化

このデモマピングでは、データベースエンポーネントレベルでのトランザグションロール・シノを使用するオークは「構成されています。これは、最初の例外が発生すると、マピングは実行を停止し、最初の状態まで全てのデータをロール・シクルます。トランザグションロール・シクオプションを変更し、マピングの実行がしコードの一部のナークは別外が発生しても継続するようします。

1. new_users テーブルの横の DEL A:In ボタをクトックし、「トランザクションの使用」 チェックボックスを選択し、「現在のトランザクションをロール 、シックし継続する」を選択します。

エラーが発	修生した場合:	現在のトランザクションをロールバックし継続します ~	

2. new_addresses テーブルの横の All ボタンをクリックし、「トランザクションを使用」 チェックボックスを選択し、「現在のトランザクションをロール デックし継続する」を選択します。

トランザンコンロール シンを完全に無効化することができます。しかしなから、これを行うとエラーか最初に発生するとマピングの実行は停止され、変更をロール シンすることはできません。ですから奨励されません。この場合トレースとログは、最初に発生した例外の時点の実行のみを反映します。トランザンシュンロール シンオニ関する詳細は データベース例外の扱いを参照してくたさい。

ステップ 2: エラーログの構成

トレースを有効化するコよ以下を行います

- 1. データベースコンポーネントのタイトルドーを右クリックし、「プロパティ」をコンテキストメニューから選択します(おけよ、データベース コンポーネントのタイトルドーをダブルクリックします)。これによりデータベースコンポーネント設定ダイアログボックスが開かれます。
- 2. 「エラー」をレースレベルとして選択します。
- 3. 「トレースファイル」の横に、トレースファイルの名前を入力します。(この場合 Errors.xml)。他のマシン、お台オペレーティングシステム上のMapForce Server でマンピングを実行する場合は、絶対パンを入力しないでください。

- トレース		
トレースレベル (E) :	Ij- ~	
ファイルのトレース <mark>(F)</mark> :	Errors.xml	参照

ターゲナデータベースコンポーネントは構造を変更し、トレースファイル構造を使用して追加コンパトメントを表示します。



データベースコンポーネントレベルでのトレースが有効化されています。今後、トレース設定はエのデータベースコンポーネント上のテーブルアクショ ンコンと継承されます。このサンプルでは、new_usersとnew_addressesテーブルの挿入アクションコンと継承されます。これらのテーブ ルのそれぞれのオンガニ、アクションボタンAin をクリックすると、設定はエの事実を表示します。



任意で、フィールドボタンをクトックして、どのフィールドがトレースファイル内に含まれるか、非表示にされるかを構成します。このサンプルでは、全てのフィールドが含まれています。

ステップ 3: マパングの出力の構成方法

このステップは、他のエレポーネントの種類、この場合、単純型出力コンポーネント)ヘトレースファイルからの値をマップする方法を示しています。このステップは狂意です。エラーログファイルは他のコンポーネントしトレースファイルをマップしない場合でも作成されます。このサンプルでは、マッピングを明示的に構成して、出力内に発生したエラーの総数を返すように構成します:

- 1. 「挿入 | 出力の挿入」メニューコマドを選択して、単純型出力コンポーネトを追加します。単純型入力コンポーネトに関する詳細は、マレングから文字列の値を返す方法を参照してください。
- 2. 「挿入 | 定数」メニューコマバを選択して、定数を追加します。 定数は、マソピング出力テキストを明確にするために静的なテキスト「エラーの総数」」を提供します。
- 3. concat 関数をライブラルらマッピングゴンラッグします。ビルトイン関数に関する詳細は、マッピングゴンバトイン関数を追加する 方法を参照してください。

下に示される通り、concat 関数は、定数のテキストとトレースファイルにより与えらえるerrors 値を結合します。



上記の構成の通り、MapForce Server を使用してマピングが実行されると、出力はデキストを表示します: value か実行されると、ランタイムに遭遇したエラーの数量を示す "エラーの総数: {value}"。

この時点では、「結果」コンポーネト上のプレビュー・ オタンを押して、MapForce内でマンピングを実行することは不可能で、 エラーを引き起こします。単純な出力はレースファイルからデータを読み取るため、トレースファイルが最初に存在することが重要です。 次のステップはレースファイルの作成方法を示しています。

ステップ 4: マピングの実行

MapForce を使用してマピングを実行する方法

- 1. データベースコンポーネント上の「プレビュー」 1 パタをクリックします。
- 2. 「出力」 ダブをクトックします。
- 3. 出力メニュー上で、「SQL-スクリプトの実行」をクトックします。この時点では、マンピングで最初の例外が発生します。これは期 待されるものです。データベーストランザンシュン例外ダイアログボックスが表示されます。

😢 MapForce: データベース トラン	ザクション例外	×
トランザクション内のエラー:	NEWADDRESSESUSERID	
エラーの原因:		
NOT NULL constraint failed: nev	v_addresses.is_shipping	0
squステートメント:		
INSERT INTO "new_addresses" "street", "number") VALUES (?, "	("user_id", "id", "is_shipping", "is_billing", "type", "city", ?, ?, ?, ?, ?, ?, ?)	
どのように続けるか選択		
○このトランザクションをロールノ	「ックして停止	
●このトランザクションをロール/	「ックして続ける	
□ 今後のエラー全てに対し	てロールバックして続ける	
○ トップをロールバックして停止		
○ トップをロールバックして継続		
	ОК	

- 4. 上記の通りロールバックオプションを選択し、「OK」をクリックします。
- メモ MapForce Server によりマピングが実行されると、ダイアログボックスは表示される、MapForce 内で構成されたロールレック 設定に従い、エラーのトランザジションが自動的にコールレックされます。

このサンプルでは、マンピングの実行が完了すると、失敗したすべてのトランザグションがロールレックされ、成功した実行のみがデータベースコンミントされます。発生したエラーは、Errors.xml ファイルに書き込まれます。以下にこの実行の一部が表示されています。



ログファイルが生成されたので、MapForce実行内のエラーの総数を返す出力をプレビューすることができます:

- 1. マピングに戻り、「結果」コンポーネト上の「プレビュー」 🤷 ポタンをクリックします。
- 2. 「出力」ダブをクリンクして、結果を確認します。

1					
				-	
マッピ	ング DB クエリ 田力				
DatabaseExceptions.mfd*					

マンピングのこの部分は、以前に作成されたログファイルからのデータを読み取ります。ログファイルか存在したい場合、エラーか生成されます。

6.3 CSV とテキストファイル

MapForce は、CSV (コンマで区切られた値) とFLF (固定長フィールド) テキストファイルなどのテキストベースのファイルから、またまへの、データのマッピングのためのサポートを含みます。例:以下のようなデータ変換を作成することができます:

- XML スキーマとフラナ ファイルフォーマナ
- データベースとフラオファイルフォーマオ

CSVの場合、ファイルはコンマだけでなく、ダブ、セミコロン、スペースませま他のカスタム化された植などの区切り記号を含むことができます。

CSV とFLF ファイルの他に、MapForce FlexText (このモジュールはMapForce Enterprise Edition で使用することができます) を使用すると、複雑おけよカスタム化された構造のナメのテキストファイルから、おけよへのマナングが可能しています。FlexTextを使用する と("FlexText テンプレート"を使用して) 他のフォーマトにマナビングするけっかに、カスタムテキストデータを定義することが可能しています。

ファイルテキストから、おさまテキストファイルへのデータのマンピングは以下の言語でサポートされています: Java、C#、C++、おさま BUILT-IN。

マピングされたフラルなファイルデータ生成/保存するコは2 つの方法がみます:

- 出力ダブをクトックすることで、内蔵の実行エンジンを使オニプレビューか生成され、「出力 | 出力ファイルの保存」メニューオプションを 選択するか、 ・ アイコンをクトックすることで結果を保存できます。
- メニューオブションの「ファイル|コードの生成 | Java、C#、おゴはC++」を選択し、生成されたコードをコンパイル、実行します。

6.3.1 例: CSV ファイルから XML へのマッピング

この例ではMFCompany.xsd スキーマファイルをベースに単純なCSV ファイルをXML ファイルヘマメピングします。以下の例で使用されるファイルは全てくマイドキュメント>WiltovaWapForce2021WapForceExamplesWutorial¥フォルダーゴ収められています。

- 1. ツールドーにて Java、C#、C++ おけま BUILT-IN のアイエンが選択されていることを確認します。
- 2. メニューオプションの「挿入 | テキストファイル」を選択するか、テキストファイルを挿入アイコン(
)をクトックします。「テキストコン
 ポーネントを挿入」ダイアログボックスが表示されます。
- コンポーネント設定ダイアログボックスから、入力ファイルをクリックして、Altova_csv.csv ファイルを参照します。ファイルのコンテンソは、ダイアログボックスの下の部分に表示されます。プレビューモードの場合最初の20列のみが表示されることご注意してくたさい。

🕑 コンポーネント設	定							.	3
コンポーネント名:	テキスト ファイル							OK(O) キャンセル(C)]
○入力/出力ファイル―									J
Altova¥MapForce2016¥MapForceExamples¥Tutorial¥Altova_csv.csv 入力ファイル(I) 出力ファイル(U)									
🔽 MFDファイルに対す	する相対パスで保存する	5 <mark>(</mark> V							
- 入力/出力 エンコーディ	んり								
エンコーディング名(E)	: Unicode UTF-8					-			
バイト オーダー(B):	リトル エンディアン		•		バイト オ・	-ダーマー	りを含める(M)	
CSV 設定 フィールドの区切り文 ○タブ ○セミコロ	:字 コン	ペース ○カ	አቃሪ፡		引用文字 ●なし	©'	● CSV(S) ● 固定長	(F)	
 ■ 先頭行をフィールド ▼ 空のフィールドを存 	名として使う 在しないものとして扱う			(● 必要な ● 常に追	場合追力 加	0 EALA		
コノニルド1	コノールドク	コスール格3	3		LKA				1
string •	string	▼ string	, •	string	ייוש ז	-	string	^	
Vernon	Callaby	582		v.calla	by@nano	null.com	Office Mar	nager	
Frank	Further	471		f.furth	er@nano	null.com	Accounts	Receivable	
Loby	Matise	963		l.matis	e@nanor	ull.com	Accountin	g Manager	
Joe	Firstbread	621		j.firstb	read@nar	nonull.com	Marketing	Manager E 👻	
•		111						Þ	
フィールドを追加(<u>A</u>)	フィールドを挿入(<u>N</u>)	フィールド [:]	を削除(<u>R</u>)	<<	>>				

4. フィールド 1 のヘッダーをクリックして、テキストの内容を(例えば Firstname に) 編集します。他のフィールドも同様の方法で編 集します: フィールド 2 => Lastname、フィールド 3 => Tel-extension、フィールド 4 => Email、フィールド 5 => Position。ビナ: ダブキーを押下することで、フィールド間を移動することができます: ヘッダー1、ヘッダー2 など。

First-name	Last-name	Tel-extension	Email	Position	
string 🗾 💌	string 💌	string 💌	string 💌	string	
Vernon	Callaby	582	v.callaby@nanonull.com	Office Manager	
Frank	Further	471	f.further@nanonull.com	Accounts Receivable	
Loby	Matise	963	l.matise@nanonull.com	Accounting Manager	
				·····	

- 5. 「OK」ボタンをクトックします。
- 6. (プロンプトにて表示される) エパーネト名を変更ポタンをクトックすることで、エンポーネト名がセナされます。 CSV エンポーネトがマノビングに挿入されます。
- 7. メニューオプションから挿入 | XML スキーマ/ファイル」を選択し、MFCompany.xsd を選択します。
- 8. スキップをクルクし、ルート要素とてCompanyを選択します

9. 両コンポーネトで対応するアイテムをマングします。行アイテムをスキーマターゲトのPerson アイテムへ接続し、出力ダンを クリックしてマンピングの結果を確認します。

CSV コンポーネント内の行アイテムからのスキーマ内の Person アイテムへの接続は、どの要素が反復するか定義するける 必要不可欠です。 CSV ファイル内の各行に対して、新しい Person 要素が XML 出力ファイル内で作成されることを意味します。



10. 出力ダブをクリックして結果を確認します。



CSV ファイルからのデータがXML ファイルへマパングされまた。

6.3.2 例: アイテムの反復

ターゲト CSV ファイル内でアイテムの反復(複数の行)の方法を説明します。このサンプルで使用されるマメピングデザインファイルは、くマ イドキュメント> Altova Map Force 2021 Map Force Examples ¥Tutorial ¥ フォルダー内にある Tut-xml 2 csv.mfd で確認で きます。


Tut-xml2csv.mfd

このマピングは、意図的に不完全し作成されています。このサンプルを、メニューコマボ「ファイル | マッピングの検証」を使用して検証しようと試みると、検証エラーが生じる可能性があります。また、マノピング出力をプレビューすると、目的に合った、まけは合わない単一の行か作成されます。

XML ファイル内のアイテムのシーケンスから CSV ファイル内に複数の行を作成することが目的だと仮定します。これはターゲオ CSV ファイルの Rows アイテムは接続を描くことで達成することができます。

全てのオフィスコ関するデーダをCSV ファイルへ出力するコよ Office を行アイテムへ接続する必要があります。MapForce で以下の命 令を与えます: Office アイテムを行アイテムへ接続することで、各 Office (とマソピングされたアイテム)かターゲオ CSV ファイルに表示さ れます。



CSV コンポーネント内のRows アイテムは、接続されたアイテムのシーケンスのオメの反復子としての役割を果たします。ですから、Office アイテムを接続すると、出力はノースXML内で見つかる各のオメリン行を作成します。

1	"Microtech, Inc.",Level 1 support,Major Ave 1,Vancouver,558833
2	"Microtech Partners, Inc.", Level 2 support, Perro Bvd 1324, Otto
з	

Department から Rows アイテムへのマピングを行うことで、ソース内の各 department のためご行が作成されます。



出力は、以下のようています

1	"Microtech, Inc.", Admin, Major Ave 1, Vancouver, 5588339, Clive, Clo
2	"Microtech, Inc.", Sales and Marketing, Major Ave 1, Vancouver, 558
з	"Microtech, Inc.", Manufacturing, Major Ave 1, Vancouver, 5588339, K
4	"Microtech, Inc.",Level 1 support,Major Ave 1,Vancouver,5588339
5	"Microtech Partners, Inc.",Admin,Perro Bvd 1324,Ottowa,3549202,
6	"Microtech Partners, Inc.", Sales and Marketing, Perro Bvd 1324,0
7	"Microtech Partners, Inc.",Level 2 support,Perro Bvd 1324,Ottow
8	

最後に Person をRows アイテムニマンピングすると、結果はすべてのPerson が出力になります。この場合、MapForce は、記録を 以下のようこ反復します: 各 Department 内の各 Office 内の各 Person.

6.3.3 例: CSV と FLF ファイルから階層を作成する

このサンプルは、以下の、なで見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Tut-headerDetail.mfd。このサンプルは次の書式を持つCSV ファイル(Orders.csv)を使用します:

- フィールド1:Hとり値はヘッダーか記述されたシュードを表し、D は詳細情報が記述されたシュードを表します。
- フィールド 2: ヘッダーと詳細レコードの共通キーです。
- 各ペダー/詳細レコードは別々の行にて表されます。

Orders.csv ファイルの内容を以下に示します。

H,111,332.1,22537.7,,Container ship,,, D,111,A-1579-227,10,3,400,Microtome,, D,111,B-152-427,7,6,1200,Miscellaneous,, H,222,978.4,7563.1,,Air freight,,, D,222,ZZ-AW56-1,10,5,10000,Gas Chromatograph,,

マピングの目的を以下に示します

- フラオファイルCSV を皆層構造のXML ファイルへマピングする
- Hにお表示されるへダーレコードをフィルタレグする
- 各へダーレコードに対してDにお表される詳細レコードを関連付ける



tut-headerDetail.mfd

上の目的を達成するコよ、ヘッダーと詳細レコードに共通のフィールドが与えられている必要があります。共通のフィールド(おオオキー)とよこの場合 CSV ファイルのレコードである OrderNo のこと てなります。 CSV ファイルの内容を見ると、最初のヘッダーレコードとそれ以降にある 2 つの詳細レコードに 111 という共通した言か与えられていることが分かります。

Order.csv ファイルが2度挿入され、マピングが出植感的に表示されています。

Tut-headerDetail.xsd スキーマファイルは階層構造にと構成されています。Order はルート要素で、子要素にHeader を持ち、更にHeader 以下にDetail 要素が配置されています。

最初のOrders.csv エンポーネントからは Header レコード(ならびにその他のフィールド)がスキーマターゲットファイルの Header アイテム にマッピングされます。フィルターエンポーネントを使用することで、H レコードのフィルタリングを行います。行アイテムにより、フィルタリングされたシ コードがスキーマファイル内にある Header アイテムに与えられます。 2 番目のOrders.csv エンポーネントからよ Header レコードにある OrderNo の値にマッチした Detail レコードからえられます。以下の処理が行われます:

- ヘッダーレコードのOrderNo フィーリドを詳細レコードの同じフィーリドとequal 関数を使用して比較します(優先コンテキスト)は拡張された。シオーマンスのオングの、ラメーター上で設定されています)。
- Logical-and 関数を使用することで、Header レコードのOrderNo と同じ値を含むDetail レコードだけが渡されるようしなります。

行アイテムからはフィルターコンポーネントのon-true パラメーターを介して、フィルタルングされオシコードがスキーマコンポーネントのHeader ならびにDetail アイテムへ渡されます。

出力ダブをクトックすることで、以下に示されるようなXML ファイルが表示されます。各ヘッダーレコードには対応するデータが、そしてDetail レコードには同じ OrderNo を持って詳細情報が表示されます。2番目の別では、別のCSV ファイルを使用します。

1	1 xml version="1.0" encoding="UTF-8"?				
2	2 📮 <order td="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemalocation<=""></order>				
3	3 🔶 <header></header>				
4		<recordtype>H</recordtype>			
5		<orderno>111</orderno>			
6		<total veight="">332.1</total>			
7		<totalunitcost>22537.7</totalunitcost>			
8		<currency></currency>			
9		<shipping-details>Container ship</shipping-details>			
10	ø	<detail></detail>			
11		<recordtype>D</recordtype>			
12		<orderno>111</orderno>			
13		<productno>A-1579-227</productno>			
14		<unit veight="">10</unit>			
15		<unitno>3</unitno>			
16		<unitcost>400</unitcost>			
17		<unit-description>Microtome</unit-description>			
18	ŀ				
19	ø	<detail></detail>			
20		<recordtype>D</recordtype>			
21		<orderno>111</orderno>			
22		<productno>B-152-427</productno>			
23		<unit veight="">7</unit>			
24		<unitno>6</unitno>			
25		<unitcost>1200</unitcost>			
26		<unit-description>Miscellaneous</unit-description>			
27	ŀ				
28					

若干異なるCSV ファイルを使用しくマイドキュメント>Altova MapForce 2021 MapForce Examples Autorial ギフォルダー内でHead-detail-inline.mfd として使用することのできる他のサンプルを見てみましょう。以下に特徴を示します:

- レコードの識別子(H お台はD)は利用できません。
- 共通/キーフィールドはCSV ファイル内の最初のフィールドに存在しており、 ヘッダーならびに詳細レコード(Head-key、Detail-key)が収められています。フィールドはスキーマターゲナのOrderNo ヘマノビングされます。
- ヘッダーと関連する詳細フィールドは全て同じ行に収められています。

111,332.1,22537.7,,Container_ship,,,111,A-1579-227,10,3,400,Microtome,,111,B-15 222,978.4,7563 1.,Air_freight,,,222,ZZ-AW56-1,10,5,10000,Gas_Chromatograph,,

マピングは以下のようこデザインされます

- キーフィールドがスキーマターゲット内の対応するOrderNo アイテムへマンピングされます。
- スキーマターゲオファイル内の詳細アイテムが複製され、以下のように表示されます: Detail (2)。これによど詳細レコードの2番目のセオを正確なアイテムにマップすることができます。
- このマッピングの結果として、上にあるサンプルと同様のXML ファイルが作成されます。



Head-detail-inline.mfd

6.3.4 CSV オプションの設定

テキストコンポーネントをマッピングエリアに追加した後、コンポーネント設定ダイアログボックス空適用できる設定を構成することができます。コ ンポーネント設定 ダイアログボックスを以下の方法で開くことができます:

• コンポーネトを選択して、「コンポーネント」メニューからプロノティ」をクリックします。

- エンポーネトへッダーをダブルクトックします。
- コポーネトヘッダーを右クリックして、「プロパティ」をクリックします。

)コンポーネン	/卜設定				
コンポーネント 名	3 : People				OK(O) キャンセル(C)
入力/出力ファイ	΄JU				
People.txt				入 カファイル(1)	
				(70327130(4)	
				出力ファイル(U)	
▼ MFDファイル	に対する相対パスで保存す	する(V			
入力 / 出力 エン:	コーディング				
エンコーディング	名(E): Unicode UTE-8				-
バイト オーダー	(B): リトルエンディアン		-	📃 バイト オーダー	マークを含める(M)
✓ 空のフィール	ドを存在しないものとして扱	ئ تارىپ اللام	77.484		٦ اللاد
J1 - JVI1 string	J1 = JV1 ≤ string	J1 = Mr∋	J1 - MI ⁴	y1=Wr5	J1 = JUINO ▼ string
Company	Department	First	Last	Title	EMail
Nanonull, Inc.	Administration	Vernon	Callaby	Office Manager	v.callaby@nanonull.cor
Nanonull, Inc.	Administration	Frank	Further	Accounts Receivable	f.further@nanonull.cor =
Nanonull, Inc.	Administration	Loby	Matise	Accounting Manager	l.matise@nanonull.com
Nanonull, Inc.	Marketing	Joe	Martin	Marketing Manager Euro	pe j.martin@nanonull.com
Nanonull, Inc.	Marketing	Susi	Sanna	Art Director	s.sanna@nanonull.com
Nanonull, Inc.	Engineering	Misballa	Butler	Program Wanager	r.iandisenanonull.com
Nanonull, Inc.	Engineering	Ted	Little	Software Engineer	t little@nanonull.com
Nanonull Inc	Engineering	Ann	May	Technical Writer	a wav@nanonull.com
Nanonull, Inc.	Engineering	Liz	Gardner	Software Engineer	l.gardner@nanonull.cor
Nononull Inc	Engineering	Dout	Smith	Software Engineer	nemith@papapull.com
•	111				4

テキストコンポーネント設定ダイアログボックス(CSV モード)

使用することのできる設定は、以下のとおいです

ユンボーネント名	コンポーネント名はコンポーネントを作成する際に自動的に生成されます。ですが、後に 名前を変更することが可能です。
	コンポーネント名はスペースを含むことができ、おっフルストップ文字も含むことができます。 コンポーネント名は、スラッシュ、バックスラッシュ、コロン、二重引用符、行頭と末尾スペー

	スを含む場合があります。 全般的に、 エンポーネントの名前を変更する際、 以下の影響を考慮してくたさい	
	 マピングをFlowForce Server にデプロイする場合、エポーネト名は一意である必要かあります。 コマ・ドラインに入力できる文字のみを使用することが奨励されます。コマ・ドラインとWindows 内では、それぞれの文字で異なったニンコードを使用する場合かあります。 	
入力ファイル	MapForce がデータを読み込むファイルを指定します。このフィールドはノースエンポーネントにとり重要で、最初のエンポーネントを作成すると入力され、テキストファイルに割り当てられます。マメピングのナッシュンケゲトとしてテキストファイルの使用されている場合は、フィールドは空のままし、ておくことができます。	
	ソースコンポーネント内で、MapForceは、このフィールドの植を使用し、列名を読み込み、 インスタンステキストファイルのコンテンンをプレビューします。	
	新しいファイルを選択するコよ「入力ファイル」をクリックします。	
出力ファイル	MapForce がデータを書き込むファイルを指定します。このフィールドはターゲナ・コンポー ネントにと」重要です。	
	新しいマアイルを選択するコよ「出カファイル」をクリックします。	
MFD ファイルニ相対的なすべてのファイル パズを保存する	このオプションが有効化されると、MapForce は、コンポーネト設定ダイアログボックス に表示された、MapForce Design (.mfd) ファイルの場所にす相対したファイル びを 保存します。この設定は、テキストコンポーネントにより使用された入力と出力ファイルを 影響します。以下も参照: <u>コンポーネントにて相対バンを使用する</u> 。	
入力/出力 エンコード	出カインスタンスファイルの次の設定を指定することを許可します	
	 エンコード名 バイトオーダー バイトオーダーマーク(BOM) 文字が含まれるか否か。 	
	デフォルトでは「新しいコンポーネントのためのデフォルトのエンコード」オプション内で 定義されたエンコードを持つ新しいコンポーネント。「ツール オプション」からこのオプショ ントアクセスし、タブを生成します。	
フィールド区切り記号	テキストファイルの区切りを選択します(CSV ファイルの場合、デフォルトの区切りである 、、、、が使用されます)。このオプションにより、タブ、セミコロン、スペース文字を区切り記号と して選択できます。独自の区切り文字をカスタムフィールドにて入力することもできます。	
最初の行をフィールドの名前と見なす	テキストファイル内にある最初のレコードの値をカラムのヘッダーとして定義します(プレビュ ーウインドウに表示されます)。テキストコレポーネントがマンピングに表示された際にコよカ ラムヘッダーがアイテム名として表示されます。	
空のフィールドを存在しないものとして扱う	このオプションが有効化されると、ソースファイル内の空のフィールドは対応する空のアイテム (要素おけば属性)をターゲットファイル内に作成しません。	
	例: CSV レコード ^W General outgassing pollutants,,,,"は4つのフィーールドから構成されており、少なくとも3つのフィールドは空です。	

	出力がXML ファイルと仮定し、このオプションが無効化されている場合、空の値を持つ 空のフィールドが出力に作成されます(このサンプルでは、要素 Last、Title、と Email):			
	33			
	38 <person></person> <person></person> <person></person> <person></person> <person></person> <person></person> <inst>100</inst> <person></person> <inst>1100</inst> < /ul>			
囲み文字	入力ファイル内のフィーリド値が囲み文字で囲まれている場合、ソースファイルと同じ囲み 文字を選択してくたさい。同じ設定が出力ファイルにき使用されます。			
	引用文字 ②なし ② " ③ " ④ 必要な場合追加 ③ 常に追加			
	出力ファイルのために、追加設定を指定することができます。			
	必要な場合追加する フィールド区切り記号、おけは改行を含むテキストのフィー みに選択された囲み文字を追加します。			
	常に追加する 生成されたCSV ファイルのすべてのフィールドに選択され み文字を追加します。			
CSV / 固定值	コンポーネトの型をCSV おけまFLF (固定値フィールド)に変更します。			
ĴĽī-IIJT	ダイアログボックスの下の部分では、入力ます」は出力として選択されたファイルの20の行を表示します。 必要であれば、以下のようこ、ファイルの構造を作成(ます」」は既存の構造を変更)することができます。			

フィールドの追加	最後のCSV レコードの後に、新規のフィールドを作成しま
フィールドの挿入	現在選択されているCSV レコードの前に、新規のフィー、 すくに生成します。
フィールドの削除	現在選択されているフィーリドを削除します。
<<	現在選択されているフィーリドをにつつ移動します。
>>	現在選択されているフィールド右につ移動します。
フィー・ルドの名前を変更する 入力します。最初の行をフィ 合、フィー・ルド名は編集するこ	コよ、ヘッダーをクリックして(例: フィーリド1)、新しい値を テーリドの名前とみなすオプションが有効化されている場 とかできません。
フィールド1 string	•
ドロップダウノルトから必要と MapForce は 入力データ す。フィーナド フォーマナがー	される値を選択し、フィールドのデータ型を変更します。 とフィールドフォーマナが一致するかデータ型をチェックしま ・致してみ、場合、はデータか防では以来とされます。
Company decimal Nanonull, Inc. Nanonull, Inc. Nanonull, Inc.	
フィー・ド型はデフォルのXM 以下のフォームで表示されます	ML ス キーマデータ型をベー ス ユ、て ます。 例: データ型は 非 YYYY-MM-DD 。

6.3.5 例: 固定長テキストファイルを(データベースへ)マッピング

このサンプルでは固定長テキストファイル(FLF)とMS Access データベースの間のマンピングを行います。このサンプルで使用されるファイル は、〈マイドキュメント〉 Wilto va Wap Force 2021 Wap Force Examples Wiltorial ¥フォルダーイコ収められています。ソーステキス トファイルとターゲート データベースは従業員のノストを保管しています。ソースファイル内では、記録にお明示的にサイズにより以下のようにカン マで区切られています。

フィールドの位置と名前	サイズ (文字数)
Field 1 (First name)	8
Field 2 (Last name)	10
Field 3 (Phone extension)	3
Field 4 (Email)	25
Field 5 (Position)	25

マルングの目的は、データベース内の各社員の内線番号を、各内線番号のプレフィックスに ~100 ~ を追加し、ソースファイル内のものに更新 することです。このゴールを達成する コよ、以下を行います:

ステップ 1: テキストコンポーネントの挿入と構成

- 1. メニューオプィンの挿入 | テキストファイル」を選択するか、テキストファイルを挿入アイエン
- スカファイルボタンをクトックしてくマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Altova-FLF.txt ファイルを選択します。ファイルが1つの、大きな)文字列と、#の充填文字から構成されていることが確認できます。

● コンポーネント設定	×
コンポーネント名: テキスト ファイル	OK(O) キャンセル(C)
入力/出力ファイル	
¥Altova-FLF.bxt 入力ファイル(I)	
出力ファイル(U)	
✓ MFDファイルに対する相対パスで保存する(V	
入力/出力 エンコーディング	
Iンコーディング名(E): Unicode UTF-8 ▼	
バイト オーダー(B): リトル エンディアン 🔹 🔲 バイト オーダー マークを	:含める(M)
CSV 設定 フィールドの区切り文字)CSV(S) 〕固定長(F)
マイールド1 string Vernon##Callaby###582v.callaby@nanonull.com###Office Manager#########Frank###Further###47*	1 f.furthe r@nano null.co
< III	4
フィールドを追加(<u>A</u>) フィールドを挿入(<u>N</u>) フィールドを削除(<u>R</u>) << >>	1.

3. 固定長を選択します。

● コンポーネント設定	۲.
コンポーネント名: テキスト ファイル をやいてル(C)	
- 入力/出力ファイル	
Altova-FLF.txt 入力ファイル(I) 出力ファイル(U)	
☑ MEDファイルに対する相対パスで保存する(V	
- 入力/出力 エンコーディング	
Iンコーディング名(E): Unicode UTF-8 ▼	
バイト オーダー(B): リトル エンディアン マ 「バイト オーダー マークを含める(M)	
固定長フィールド設定 - 充填文字	
● 本 (× ×) () / ()	
▼ 空のフィールドを存在しないものとして扱う	
フィールド1 string	Ţ
1	
V	
フィールドを追加(<u>A)</u> フィールドを削除(<u>R</u>)>>	//

4. レコードの区切り文字があるものとして処理チェックボックスをクリアします。

フィールド1	
string	I
1	
V	
e	
r	.
•	•

5. 黄色で イライトされた3行は編集することができます。 およ、以下を指定することができます: i) フィールド名、ii) データ型、iii) フィールドのサイズ。 値を8 に変更した後 Enter キーを押下します。 8文字ごとこフィールドが区切られるため、 より多くのデータが 最初のレコードに表示されます。

フィールド1		
string	-	
8		
Vernon##		
Callaby#]
##582v.c		-
٩	Þ	1

6. フィールドを追加ボタンをクリックして、新たなフィールドを追加します。2番目のフィールド長さを10文字に指定します。

フィールド1	フィールド2		
string	✓ string	-	
8	10		
Vernon##	Callaby###		
582v.cal	laby@nano n		
ull.com#	##Office M		Ŧ
		Þ	

7. 同様の方法を使用して、更に3つのフィールドを作成します。長さはそれぞれ3、25、25 文字にないます。更にフィールドの名前を それぞれ First、Last、Tel.-Ext、Email、Title とすることで、簡単にマッピングを行えるようにします。 ブレビューは以下のように表 示されます:

First	Last	TelExt	Email	Title	*
string 💌	string 💌	string 💌	string 💌	string	
8	10	3	25	25	
Vernon	Callaby	582	v.callaby@nanonull.com	Office Manager	
Frank	Further	471	f.further@nanonull.com	Accounts Receivable	
Loby	Matise	963	l.matise@nanonull.com	Accounting Manager	
Joe	Firstbread	621	j.firstbread@nanonull.com	Marketing Manager Europ	
Susi	Sanna	753	s.sanna@nanonull.com	Art Director	
Fred	Landis	951	f.landis@nanonull.com	Program Manager	Ŧ
•		III		Þ	

8. 固定長フィールド設定グループのカスタムテキストボックスをクリックして、ハッシュ文字(#)を入力します。この操作により、充填文字とて使用されている文字が入力文字列から取り除かれます。

- 固定長 - 充埴	フィール 文字	ド設定		
) () ス	~-, ~-,	⊚ピリオド	◎ カスタム	: #

9. 「OK」をクルクします。

MapForce
インスタンス ファイル名を変更しましたが、コンポーネント名は変更されていません。新たなインスタ ンスファイルにマッチする様に、コンポーネント名を 'Altova-FLF' に変更するか、そのままにしておく か選択して下さい。
コンポーネント名を変更 コンポーネント名を変更しない キャンセル

10.「コンポーネト名を変更」ボタンをクトックして、名前を変更します。テキストファイルコンポーネトがマンピングウイドウン表示されます。このコンポーネントを使オーデータのマンピングが行えるようコンジます。

	I Altova-FLF	
	回日ファイル: Altova-FLF.txt <u>ファイル/文字列</u> に	>
j	> □ () 行 () > □ = コイールK1	>
լ	> =	>
ľ	> ー=フィールド3	>
j	> ====================================	>
ן	י דות רכ	

ステップ 2: データベースコンポーネントの挿入

- 1. メニューオプションの「挿入 | データベース」を選択し、Microsoft Access デジオドタンを選択して「次へ」をクルクします。
- 2. <マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥Tutorial¥フォルダー内にある altova.mdb デ ータベースを選択し、「接続」をクトックします。
- 3. Person テーブルのチェックボックスを選択し、「OK」をクトックします。

(⇒データベース オブジェクトを挿入	×
テーブル、ビュー、ストアドブロシージャを選択するか、独自の SQL SELECT ステートメントを作成してマッピングに挿入してください。ユーザーにより定義された SQL SELECT ステートメントはローカルの mfd ファイルに保存されます。 ソース	
 ▼ ● SELECT ステートメントの追加/編集 リレーションの追加/編集 (0) レコードセット構造の追加/編集 □ ○ altova □ □ □ ○ altova □ □ □ ○ altova □ □ □ □ ○ altova □ □ □ ○ altova □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
プレビューを表示(P) >> OK	キャンセル

ステップ 3: マッピングのデザイン

- 1. core | concat 関数をライブラリウイドウからマピングダブイドラッグします。
- 2. メニューオブションの「挿入 | 定数」を選択し、数値ラジオボタンをクリックした後に、新たな内線番号のプレフィックスとして使用する 100 を入力します。この定数が内線番号のプレフィックスを保管します。

🕑 定数を挿入		×
100		 ○ 文字列(S) ● 数値(N) ○ その他(O)
	ОК	キャンセル

3. 以下のスクノーンショナに示されるマピングを作成します。



- 4. Person エトリーの隣にあるテーブルアケションボタン All をクリックします。
- 5. 「入力データアクション」のエンボボックスをクトックし、「以下の条件で更新..」エトリーを選択し、First とLast のフィールドが一致するようご設定します。この操作により、First の内容がシーステキストファイルとデータベースにて一致する場合のみ、 Person テーブルが更新されるようゴンリます。条件が真の場合に取られるアクションはマッピングにより定義されており、この場合は "100" と、ラプレフィックスが追加された内線番号がPerson テーブルのPhoneExt フィールドに与えられます。

🕑 データベーステーブル	Dアクション - Person			×
最初のレコードの前に実行す ◎ なし(N) ◎ 全てのレコードを削除(L) ◎ カスタム SQL(C) :	する SQL ステートメント	ある全てのレコー	ドも削除	
●各レコードに対して実行する 以下のオペレーションを使用 比較の結果が全て真であれ	アクション 犯て全ての入力データは I れば、指定されたアクション፣	DB デーブルのデー が実行されます。	タと比較され	ます。
入力データ アクション 『PrimaryKey 『ForeignKey 『EMail 『First 『Last 【	全て挿入 マップされた値 マップされた値 マップされた値 マップされた値 マップされた値			アクションを追加(A) アクションを挿入(I) アクションを削除(D)
 ■トランザクションを使用(■バルク転送を使用 (Ma バルク転送はデータベース) 	T) pForce Server のみ)(B) 接続によりサポートされてい	バッチ サイズ ません。	с 1	000 レコード キャンセル

ステップ 4: マピングの実行

出力ダブをクトックして、生成されたSQLステートメントをプレビューします。SQLスクリプトの実行パタン 🔒 をクトックして、SQLステートメートを実行します。

6.3.6 FLF オプションの設定

テキストコンポーネントをマッピングコレア追加した後、コンポーネント設定ダイアログボックスから適用することのできる設定を構成することができます。 ます。コンポーネント設定 ダイアログボックスを以下の方法で開くことができます:

- コンポーネントを選択して、「コンポーネント」メニューから「プロノティ」をクリックします。
- コンポーネントへッダーをダブルクトックします。
- コポーネトへダーを右クルクして、「プロパティ」をクルクします。

🕑 コンポーネント部	定			
コンポーネント名:	テキスト ファイル			OK(O)
┌入力/出力ファイルー				++7211(C)
C:¥Users¥	¥Documents¥Altova¥MapFor	ce2016¥MapForceExamp 入	カファイル(I)	
		上 出	カファイル(U)	
▼ MFDファイルに対	する相対パスで保存する(v			
○入力/出力 エンコーデ	イング			
Iンコーディング名(E): Unicode UTF-8		•	
バイト オーダー (B) :	リトル エンディアン	- //	イトオーダーマークを含める	(M)
- 固定長フィールド設定 - 充填文字	2			
●スペース ○t	ደሀオド 🔘 ከスタム:		() CSV(() 固定	S) '長(F)
▼レコードの区切り	文字があるものとして処理		0 2.4	
☑ 空のフィールドを存	昇在しないものとして 扱う			
フィールド1	フィールド2	フィールド3	フィールド4	
string	✓ string	✓ string	✓ string	_
1 V	1	1 r	1	
0	e	I	pi	
4				Þ
フィールドを追加(<u>A</u>)	フィールドを挿入(<u>N</u>) フィ	ィールドを削除(<u>R</u>) (<く)>	>	1.

テキストコンポーネント設定ダイアログボックス(固定値フィールドモード)

使用することのできる設定は、以下のとおりです

コンポーネント名はコンポーネントを作成する際に自動的に生成されます。 ですが 後に 名前を変更することか可能です。
コンポーネント名はスペースを含むことができ、おっフルストップ文字も含むことができます。 コンポーネント名は、スラッシュ、バックスラッシュ、コロン、二重引用符、行頭と末尾スペー スを含む場合があります。全般的に、コンポーネントの名前を変更する際、以下の影響 を考慮してくたさい
 マピングをFlowForce Server にデプロイする場合、コンポーネント名は一意である必要があります。 コマ・ドラインに入力できる文字のみを使用することが奨励されます。コマ・ドラインとWindows 内では、それぞれの文字で異なオミエンコードを使用する場

	合かがます。
入力ファイル	MapForce がデータを読み込むファイルを指定します。このフィールドはノースコンポーネントにと)重要で、最初のコンポーネントを作成すると入力され、おこテキストファイルに割り当てられます。テキストファイルコンポーネントをマンビングのためにターゲントとして、使用する場合、フィールドは空にしてくたさい。
	ソースエンポーネント内で、MapForce はこのフィーノドの値を使用して、列名を読み込み、インスタンステキストファイルのエンテンソをプレビューします。
	新しいアイルを選択するコは「入力ファイル」をクリックします。
出力ファイル	MapForce がデーダを書き込むファイルを指定します。このフィールドはターゲオ・コーポー ネントにとり重要です。
	新しいファイルを選択するコよ「出カファイル」をクリックします。
MFD ファイルロニ相対的なすべてのファイル パズを保存する	このオプションが有効化されると、MapForceは、コンポーネト設定ダイアログボックス に表示された、MapForce Design (.mfd)ファイルの場所に相対したファイルのを 保存します。この設定は、テキストコンポーネントにより使用された入力と出力ファイルに 影響を与えます。以下も参照: <u>コンポーネントにで相対いてを使用する</u> 。
<i>入力 / 出力 エンコー</i> ギ	出カインスタンスファイルの次の設定を指定することを許可します
	 エンコード名 バイトオーダー バイトオーダーマーク(BOM) 文字が含まれるか否か。
	デフォルトでは、「新しいコンポーネントのためのデフォルトのエンコード」オプション内で 定義されたエンコードを持つ新しいコンポーネント。「ツール オプション」からこのオプ ション(こアクセスし、タブを生成します。
充填文字	データの長さがフィールドで定義されたものより先短い場合、このオプションで指定された文字により残りのスペースが充填されます。カスタムフィールドにより独自の充填文字を定義することができます。
	カスタムフィーリドに入力された充填文字と同じ文字が入力データに含まれている場合、 入力データにある文字も充填文字として切り取られます!
レコードの区切り文字があるものとして処 理	このオプションは、ソースフラホファイルからCR/LFなどの区切り記号を含まれ、ゲーダを 読み込むためこ役に立ちます。ませよ、区切り文字を含まれ、ターゲホフラホ FLF ファ イルを作成する際にも役に立ちます。
	下のレコードの区切り文字があるものとして処理するオプションセクションを参照してくたさい。
空のフィールドは存在しないものとして扱う	このオプションが有効化されると、ソースファイル内の空のフィールドは対応する空のアイテム (要素おけば属性)をターゲットファイル内に作成しません。
	出力がXML ファイルと仮定し、このオプションが無効化されている場合、空の値を持つ 空のフィールドが出力に作成されます(このサンプルでは、要素 Last、Title、と Email):

	33 ● <person> 34 <first>Gener 35 <last></last> 36 <title></title> 37 <email></email> 38 39 ● 40 <first>1100 41 <last>897 38 39 ● 40 <first>1100 41 <last>897 38 ● 39 41 <last>60 38 ● 41 38 41 <last>60 39 41 42 43</last></last></last></first></last></first></first></person>	al outgassing pollutants /First> .ast> < 空のスイーノいドは出力に作成されません al outgassing pollutants .First> .ast>
CSV / 固定	コンポーネトの型をCSVま	たまFLF(固定値フィールド)に変更します。
ILEI-IV	ダイアログボックスの下の部分で を表示します。 必要であれば、以下のようこ、 とかできます。 フィールドの追加 フィールドの崩除 << >> フィールドの利除 << >> フィールドの名前を変更するに 1)、新しい値を入力します。 フィールド1 string 8 Vernon## Frank### Loby#### Joe##### Susi#### Fred####	Gは、入力ます」は出力として選択されたファイルの20の行 、ファイルの構造を作成、ます」は既存の構造を変更)するこ 最後のレコードの後に新規のフィールドを作成します。 現在選択されている記録の前に、新規のフィールドをすく1 します。 現在選択されているフィールドを削除します。 現在選択されているフィールド左にコープ移動します。 は、 ヘッダーをクリックして (このサンプルでは、フィールド

フィールドのデータ型を変更するコよ、ドロップダウンリストから必要な値を選択してくたさ い。MapForce は、入力データとフィールドフォーマナトが一致してい場合赤で イライト できるように、データ型をチェックします。
フィールド1
de cimal 📃
8
Vernon##
Frank###
Lo by####
Joe#####
Susi####
Fred####
 フィールド内の文字のサイズを設定できるように、上から3番目にフィールドのサイズを入力 します。

"レコードの区切り文字があるものとして処理"オプションを理解する

このオプションを理解するけっかいは、 <マイドキュメント> ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥Tutorial ¥フォルダー内のAltova-FLF.txtファイルを開きます。 ファイルは、 CR/LF などの区切り文字を使用したい、 71文字の長さのレコードから構成されています。 このファイルからデータを読み込む場合、最初にこのファイルをレコードに分割する必要があります。 すなけた、 下に表示されるようにフィールドが 1文字である複数のフィールドを作成し、 レコードの区切り文字があるものとして処理を無効化します。 順序を追す上例は、 例 固定長のデキストファイルをデータベースにマンピングするを参照してください。

固定長フィールド設定 充填文字					
■レコードの区切り文字	があるものとして処理				
☑ 空のフィールドを存在し	しないものとして 扱う				
フィールド1	フィールド2	フィールド3	フィールド4	<mark>フィールド5</mark> 🔺	
string 🗾	string 🗾	string 🗾	string 💌	string	
8	10	3	25	25	
Vernon##	Callaby###	582	v.callaby@nanonull.com###	Office Manager### 😑	
Frank###	Further###	471	f.further@nanonull.com###	Accounts Receivab 🗍	
Loby####	Matise####	963	l.matise@nanonull.com####	Accounting Manage	
Joe#####	Firstbread	621	j.firstbread@nanonull.com	Marketing Manager 📖	
Susi####	Sanna#####	753	s.sanna@nanonull.com#####	Art Director######	
Fred####	Landis####	951	f.landis@nanonull.com####	Program Manager##	
Michelle	Butler####	654	m.landis@nanonull.com####	Software Engineer#	
Ted#####	Little####	852	t.little@nanonull.com####	Software Engineer#	
Ann#####	Way#######	951	a.way@nanonull.com#######	Technical Writer###	
Liz#####	Gardner###	753	l.gardner@nanonull.com###	SoftwareEngineer#	
Dou1####	Smith#####	224	nemith@nononull.com######	Software Engineer#	
		111		•	
フィールドを追加(A)	フィールドを挿入(<u>N</u>) フィールドを挿入(<u>N</u>) フィールドを挿入(<u>N</u>)	(ールドを削除(<u>R</u>) <<			

このファイルから同じ構造を使用する目的のファイルにデータを書き込む場合、"レコードの区切り文字があるものとして処理 "を有効化すると、新規のレコードが72文字目から新たに表示されます。

1	Vernon##Callaby###582v.callaby@nanonull.com###Office Manager######################
2	Frank###Further###471f.further@nanonull.com###Accounts Receivable######
3	Loby####Matise####9631.matise@nanonull.com#####Accounting Manager#######
4	Joe#####Firstbread621j.firstbread@nanonull.comMarketing Manager Europe#
5	Susi####Sanna#####753s.sanna@nanonull.com#####Art Director###############
6	Fred####Landis####951f.landis@nanonull.com####Program Manager############
7	MichelleButler####654m.landis@nanonull.com####\$oftware Engineer########
8	Ted#####Little####852t.little@nanonull.com####\$oftware Engineer########
9	Ann#####Way######951a.way@nanonull.com#######Technical Writer########
10	Liz#####Gardner###7531.gardner@nanonull.com###Software Engineer########
11	Paul####Smith#####334p.smith@nanonull.com#####Software Engineer########
12	Alex####Martin####778a.martin@nanonull.com####IT Manager##################
13	George##Hammer####223g.hammer@nanonull.com####Web Developer####################################
14	Jessica#Bander####241j.band@nanonull.com######Support Engineer#########
15	Lui#####King######3451.king@nanonull.com######Support Engineer#########
16	Steve###Meier#####114s.meier@nanonull.com#####Office Manager######################
17	Theo#####Bone######331t.bone@nanonull.com######Accounts Receivable######
18	Max#####Nafta#####122m.nafta@nanonull.com#####PR & Marketing Manager US
19	ValentinBass######716v.bass@nanonull.com######IT Manager#########################
20	Carl####Franken###147c.franken@nanonull.com###Support Engineer###################################
21	Mark####Redgreen##152m.redgreen@nanonull.com##Support Engineer###################################
22	

「レコードの区切り文字があるものとして処理 」が有効化された場合のマンピングの結果

"レコードの区切り文字があるものとして処理 " か無効化されている場合、マンピングされたデータが、充填文字とともに、1 つの長い文字
列として表示されます。



"レコードの区切り文字があるものとして処理 "が無効化された場合のマッピンク結果

6.4 バイナリファイル

MapForce はBLOB (バイナリラージオブジェケト)データをバイナリファイルからマングに読み取り、バイナリデータ(生)の内部構造を 変更することなく消費する機能を搭載しています。例えば、バイナリデータをデータベースBLOB フィールド、XML ファイル内の xs:base64Binaryのフィールドに保存、おけよ、Web サービスに送信することができます*。

*Web サービス呼び出しはMapForce Enterprise Edition のみでサポートされています。

(データベース内のBLOB フィールド、XML ファイル内の型 xs:base64Binary のフィールド、Web サービスなど)一部のノースから イ ナリデータを読み取るマンピングを作成し、ディスタン イナリファイルを書き込むこともできます。

以下は、イナリファイルの読み取り、おとお、イナリファイルへの書き込み機能の利点を示したサンプルシナリオです。

- ベース-64 データとてエンコードされたッドイナを XML ファイルから抽出しディスクに(例えば PDF ファイルとしつ保存します。
- ディスクトに保管されたイメージファイルを処理してベース-64 エンコードされたパイナリエンテンソとしてWebサービスに送信します。
- BLOB コンテンンをデータベーステーブルから抽出し、ディスクにイメージファイル(データベーステーブル内の各行に対して1つのイメージファイル)として保存します。
- ディスクからイメージファイルを読み取り、BLOBデータフィールドとしてデータベーステーブルに保存します。
- メモ バイナリファイルからデータをマピングする、おけよ、バイナリファイルにデータをマピングするけよ、変換言語とてBUILT-INが 要けなります。MapForce内でマピングをプレビュー(し存在する場合は出力ファイルを保存)、おけま個別にライセンス供与されている)MapForce Serverを使用して異なるコンピューターおけよプラルフォームで実行することを選択できます。実行可能な C#, C++, おけよしava プログラムをパイナリファイルを読み取るまけけお イナルファイルに書き込むマピングから生成することは サポートされていません。

バイナリファイルをマピングに追加する方法

XML テキスト、ませまJSON ファイル同様にMapForce 内でバイナリファイル 関連するエンポーネントは存在しません。代わりに、上記のような目的を達成するためにMapForce ビルドイン関数を使用することができます:

- <u>read-binary-file</u>
- <u>write-binary-file</u>

これらの関数をライブラリウィドウで見つけることができます(具体的には[lang] ライブラノで見つけることができます)。マピング内でどちらかの関数を使用するには、ライブラリウィドウからマピングドボラッグ、おけは、マピングの空のエリアをダブルクリックし、関数名の入力を開始して選択します。詳細に関しては、マピングに関数を追加するを参照してください。

read-binary-file

この関数は型 xs:base64Binary のBLOB (バイナリラージオブジェクト) として指定されたファイルのエンテンンを返します。データ型が [base64Binary] と呼ばれる場合でも内部の表記はBLOB です。関数の結果を型 xs:base64Binary のXML ノードにマップする 場合のみ、ベース64 エンコードになります。関数の結果を xs:hexBinary、データベースプロブ、ませまプロトコール デファー構造内のバ イナリフィールドにマップすることもできます。

	f_8 read-binary-file					
Ç	>filepath	contont [
ļ	>must-exist	content				

バイナリファイルをマピングに読み込むにはfilepath 引数に対しての入力としていた提供します。filepath が相対的な場合 MapForce はマピングと同じディレクトリ内でファイルを探します。must-exist 引数が任意です。ファイルを開くことしてできず、この、ラメー ターはtrue、してなり、マンピングはエラーをスローします。ファイルを開くことができなく、この、ウメーターがfalseの場合、からの・イナリか返されます。

write-binary-file

この関数は、イナリニンテンルを指定されたファイル、マニ書き込み、書き込まれたファイルの、マを返します。、イイナリファイルが唯一の希望する 出力の場合 単純型出力 コンポーネントに関数の結果を接続します。出力がマンピング内で使用される都度この関数はファイルに書き込む ため、間に他の処理を使用することなく、関数の結果を直接ターゲントコンポーネントに接続することが奨励されます。

f ₈ write-binary-file					
	filenath				
¢ content					

バイナリファイルを書き込むはすfilepath 引数に対しての入力としていを提供します。filepath が相対的な場合 MapForce はマピングと同じディレオ・リ内でファイルを生成します。The content 引数は実際の、イナリエレテンソ(例えば、データベース内のBLOB フィールド)に接続されている必要があります。

MapForce 内でマピングをプレビューすると、ディスクィンアイルを直接書き込む代わりに関数は一時的なファイルをデフォルトで生成します。

一時的なファイルをディスタニ保存するコよ、最初に出力 タブをクリックし適用可能な場合 生成された出力を保存 🗓 まけま全ての 生成済みの出力 🕼 ツール いーボタンをクリックします。

MapForce が出力を一時的なファイルではなくファイナルファイルに直接書き込むように構成するコは[ツール|オプション メニューコマ・ド] を選択し [全般]をクックして [最終出力ファイルに直接書き込む] オブションを選択します。 このオプションは同じ名前を持つ既存のファイ ルを上書きすることに注意してくたさい。

この関数はファイルカディスクに保存されていない場合(マノピングをプレビューし [最終出力ファイルに直接書き込む] オプションカ無効化されている場合)でも常に最終(一時的ではな)、ファイル名を返します。

マピングが自身の出力ファイルを読み取ることはサポートされていないてという意してくたさい。

例 サンプル・バイナル読み取りと例、バイナノファイルの書き込みを参照してくたさい。

6.4.1 例: バイナリファイルの読み取り

下て説明されるマッピングはイメージファイルからデータを読み取りデータベーステーブルに書き込みます。ターゲットデータベースはSQLiteです。pictureデータベースフィールドのデータ型はBLOBであることに注意してくたさい。



ファイルカシ、イナレンテンンを抽出するためにread-binary-file 関数が使用されました。このサンプルでは最初の引数 filepath が 関数により提供されています。ノンの相対的なため MapForce はマンピングとして同じディレクトリ内のイメージファイルを検索します。

マピングはターゲナデータベースに次のフィールドを作成します

id - このサンプルではid がデータベースコンポーネントがマンピングにとい提供されるとりに構成されています。データベースにとい生成されるように構成されています。詳細に関してはデーダをテーブルに挿入するを参照してくたさい。

😢 Database Table Actions - products				
SQL statement to execute before first record ● None				
O DELETE all records	also delete all records in all child table	s		
O <u>C</u> ustom SQL:				
Actions to execute for each rec	ord			
All input data are compared to the DB table data, using the operators Select the "NULL equal" check box to treat NULL values as equal. If all comparisons are true, then the specific action is executed.				
Action on record	NULL equal Insert All 📃 💌	j		
💵 id	DB-generated 💌	I		
1 title	mapped value 💌	J		
description	mapped value 💌	J		
picture	mapped value 💌	J		

• title - この値は単純型入力コンポーネトにより提供されています。デザイン時の実行値は("product1") に設定されており マンピング内でプレビューできるよう なっています。詳細に関しては、マンピングレン ラメーターを与えるを参照してくたさい。

🙁 Create	Input X
<u>N</u> ame:	title
Datatype:	string ~
	✓ Input is required
Design-tin	ne Execution
Specif	fy value
<u>V</u> alue:	product1
	OK Cancel

• picture - このフィールドは read-binary-file 関数の直接の出力を受け取ります。

ターゲナコンポーネントはデータベースであるため、マンピングのプレビューは疑似 SQL スクレプトを生成しますがデータベースに変更を送信しません。実際のスクレプトをデータベースに対して実行するコゴ 出力 | SQL-スクリプトの実行」を選択します。

6.4.2 例: バイナリファイルの書き込み

下て説明されているマピングはBLOB 値をSQLite データベースから読み込み ディスクにイメージファイルを書き込みます。データベースに は以下の列(フィールド)を持つproducts と呼ばれるテーブルが存在します:

- id (整数、レコードの一意の永続的なシリアル番号です)
- title (*テキス*ト、製品のタイトル)
- description (テキスト、製品の説明)
- picture (*ブロブ、*製品のイメージ)

このサンプルのスコープのためにid とpicture フィールドは関連性があります。



このマッピングのゴールは products テーブルからすべてのピクチャを抽出しディスクにファイルとして書き込みます。上記のとおり writebinary-file 関数が目的を達成するために使用されています。最初の引数 filepath はファイルレ ひを各イメージのために受け取りま す。ファイルの上書きを回避するために、以よ一意である必要があります。ですから用語 [image] を持つ各レコードの一意のデータベースid を連結し、 [.png] ファイル拡張子を追加することに以生成されています (全てのピクチャイは PNG の書式を持っていることが想定されま す)。 パマスマッピングと同じディレクトリに保存されるように、マスは相対的です。

2番目の引数 content はデータベース内に保存されている イナリコンテンルを受け取ります。

count 関数はすべての生成されたファイルのカウトを返し、この数値を文字列 [書き込まれたファイル] ど連結します。これはマンピングによ り生成されたファイル数に関するレポートを提供します。このサンプルでは、データベースは2つのレコードのみを含んでおり、マンピングの出力はこ れを反映しています:

Preview 1 of 3 Preview of simple output: result	~
1 2 file(s) written.	A
	-
Mapping DB Query Output	

前記の通り、マピングがプレビューの実行で作動する場合この関数は一時的なファイルを生成します。各ファイルをプレビューする場合、矢印を使用して興味のあるレコードをナビゲートし、以下で開くボタンをクリックし、イメージエディターを選択します。

• •	Content 2 of 3		SIGNARMAN, I DU	ExampleFiles\EN\MapForce\Functions\write-binary-file\image1.png	\sim
File size: 3	3727 bytes				
Open	With				
Mapping	DB Query	Output			

全てのファイルを保存するコよ生成された出力を保存 🗊 おっよ全ての生成された出力を保存 🗐 ツール デを適用できる箇所で

7 関数

MapForce 内では必要に応じてデータを変換するために以下の関数のカテゴレを使用することができます。

- MapForce ビルドイン関数 MapForce 内でこれらの関数は定義済みで、これらを文字列、数値、日付、および他のデー タ型を扱う広範囲の処理を行い、マピング内で使用することができます。グループ分け、集計、自動付番、および他の多種のタス 少を行うさめに使用することができます。使用することのできるすべてのビルドイン関数への参照に関しては、関数ライブラルの参照 を参照してください。
- ノード関数とデフォルト これらはマッピングコンポーネント上の1つまけは複数の子孫ノードにカスタム処理ロジックを作成し適用 する更に特化された関数です。マッピング構造のデータに到達する前まけは回達した直後にデータを処理できるようはなります。
 詳細に関してはデラオルトとノード関数を参照してくたさい。
- ユーザー定義関数 (UDF) これらはMapForce 内で使用が可能なネイティブなエンポーネト型とバイン関数のベースとする、自身で作成が可能な MapForce 関数です。ユーザー定義関数を参照してください。
- カスタム関数 これらはXSLT ライブラ人 XQuery ライブラノモジュール Java .class ファイル .NET .dll ファイルなどの 外部ノースからインポートし MapForce に適用することが可能な関数です。MapForce 内で再利用可能にするコよカスタム 関数は文字列まけは整数)単純型のデータを返し、単純型の ラメーターを取る必要がおりま。詳細に関してはカスタムXSLT 関数のインポート、カスタム XQuery 1.0 関数のインポート とカスタム Java と.NET ライブラルのインポートを参照してくたさい。
- メモ (構成を必要とすることなく) 関数のカスタム外部ライブラ を直接、封は MapForce にと認識されている MFF (MapForce 関数 ファイル) を構成することにというポートすることができます。後者のアプローチを使用する場合 Java グラスと.NET アセンブリリコ加え C++ ライブラ をつっポートすることができます。.mff ファイルによいつポートされたライブラルは Java、C# とC++ ライブ <u>ラ を手動で参照する方法</u>で説明されている必要条件を満たす必要かあります。

を実行するコよ	以下を参照してください
MapForce 関数、おけた変数をマメングに追加する方法	 マレングに関数を追加する マレングに定数を追加する 夏数の検索 関数の型と詳細を確認する 関数引数の追加、おけば削除
MapForce 内で自身の関数を作成し、同じマナング封ま地の マナングで後に使用する方法	● <u>ユーザー定義関数</u>
カスタムXSLT 関数をMapForce にんポートする方法	• <u>カスタム XSLT 関数のインポート</u>
カスタム.NETとJava ライブラをMapForceに追加する方法	• <u>カスタム Java、C# とC++ 関数ライブラルの追加</u>
MapForce 関数ファイル(.mff)を使用して適応することより スタムJava、C# とC++ ライブラを MapForce にんポート します。このアプローチは上記のアプローチとは異なり手動の構成を 必要としますがC++ ライブラルのインポートを追加でサポートしてい ます。	 Java、C# おばC++ ライブラルご手動的に参照する 方法

以下のロードマップを使用して特定のタスクは関連した関数にアクセスしてくたさい

7.1 使用方法

7.1.1 マッピングにビルトイン関数を追加する

MapForce はま下記の通り、マピングに追加することのできるすくに使用可能な多数のビルドイン関数か含まれています。使用することのできるすべてのビルドイン関数への参照に関しては、関数ライブラルの参照を参照してくたさい。

以下を前提条件として下で説明されるアプローチと同様ユーザー定義関数(UDF)をマピングに追加することができます:

- 同じマピング内でUDF か既に作成されている おけよ
- UDF を含むマピングをローカルませまグローマ シリライブラ として既にインポートしている。

マッピング内で関数を使用する

- 1. 変換言語を選択します。ライブラリウィズウ内の必要な関数をクリックし、マッピングエリアイズラッグします。
- 2. 名前別に関数をフィルターするコよウィドウの下の部分内にあるテキストボックス内に関数名を入力します。ライブラノウィドウの下の部分内にあるテキストボックス内で関数名の入力を開始します。

ライブラ	У	-	џ	×
2 00	re			^
	aggregate functions			
	avg			
	count			
	max		_	
	max-string		_	
	min		_	
	min-string		_	
	string-join		_	
	sum		_	
-	conversion functions			
	boolean		_	
	format-date		_	
	format-dateTime		_	
	format-number		_	
	format-time		_	
	number		_	
	parse-date		_	
parse-dateTime				
	parse-number		_	
	parse-time		_	
	string		_	×
関数の	の検索			•
	ライブラリを追加/削除		_	
2 51	ブラリ 🛅 プロジェクト			

代わりに、定数を次のように追加することができます

1. マノピングの空白のエリアをダブルクトックし、関数名を開始します。入力されるテキストによりフィルターされるライブラリウィドウと同じ 関数を持つエンドドックスが表示されます。各関数に関する詳細を表示するとこれを確認するコよ、リスト内から関数を選択してく ださい。

ſ	
	con
	core.concat core.contains lang.convert-to-utc lang.millisecond-from-datetime lang.millisecond-from-duration lang.second-from-datetime lang.second-from-duration

2. 必要とされる関数を選択し、マンピングに追加するためにEnterを押します。関数を選択することなく、コンボボックスを閉じるこは、Escapeを押し、ボックス外のエリアをクトックします。

7.1.2 マッピングに定数を追加する

定数を使用することことりマッピングにカスタムのテキスト、おさば数値を与えることができます。定数の値は、その名前が示すように、マッピングのライフタイムの期間中、変化せず同じままの状態でいます。

マッピングに定数を追加する方法:

- 1. 以下のつを行います:
 - a. 「挿入」メニューから、「定数」をクルクします。
 - b. マピングを右クトックし、コンテキストメニューから「定数を挿入する」を選択します。

😢 定数を挿入		×
1		● 文字列(S) ○ 数値(N) ○その他(O)
	ОК	キャンセル

2. 定数の値を入力し、データ型(「文字列」、「数値」、「その他」)を選択し、「OK」をクリックします。

代わりに、定数を次のように追加することができます

- 1. 空のマピングエル内をダブルクトックします。
- 2. 以下のつを行います:
 - a. 文字列の定数を追加するには、二重引用符の後に定数を入力します。終わりの二重引用符は任意です。



b. 数値の定数を追加するコよ数値を入力します。

3. 「Enter」を押します。

7.1.3 関数の検索

ライブラリウィドウ内で関数を検索するコよライブラリウィドウの下の部分内にあるテキストボックス内で関数名の入力を開始します。

ライブラリ		×
🔁 core		<u> </u>
_ ⊖aggre gate	functions	
count	result = count(r	
		_
		- I
count	×	•
51:	ブラリを追加/削除	
置 ライブラリ (┣ プロジェクト	

デンォルトでは、MapForce は関数名、または説明テキストにより検索することができます。検索から関数の説明を除外するには、下向きの 矢印をクックし、「関数の説明を含む」オプションを無効化します。

ライブラリ		×
🔁 core		
aggregate functions		
count	result = count(r	
Concernance functions		
	$ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$	
first-items	results = first-ite	
last-items	results = last-ite	
replicate-item	results = replicat	
replicate-sequence	results = replicat	
skip-first-items	results = skip-fir	
		Υ.
count	×	R
ライブラリを追加/削除		
置ライブラリ 🛅 プロジェ	:クト	

検索をキャンセルするコは、「Esc」キーを押す、おけば 🗶 」をクリックします。

ライブラリウイドウ内で使用することのできる関数は、現在選択さえている変換言語によりにとなります。次を参照してくたさい、変換言語の選択。

現在アクティブなマッピング内での関数のすべての発生を検索するコよ、ライブラリウイドウ内の関数名を右クリックし、「全ての呼び出しを 検索する」をコンテキストメニューから選択します。検索結果 がシッセージ ウイドウ内に表示されます。

7.1.4 関数の型と詳細を確認する

入力、または出力の引数のデータ型を確認する

- 1. 「ヒントの表示」 1. 「ヒントの表示」 1. 「ヒントの表示」 1. 「ヒントの表示」 1. 「ヒントの表示」 1. 「トントの表示」 1. 「「トントの表示」 1. 「トントの表示」 1. 「トントの人」 1. 「トントの人」 1. 「トントの表示」 1. 「トントの人」 1. 「トントーの人」 1. 「トントの人」 1. 「トントの人」 1. 「トントの人」 1. 「トントの人」 1. 「トントーの人」 1. 「トントの人」 1. 「トントーの人」 1. 「トーー人」 1. 「トーー人」 1. 「トーー人」 1. 「トー人」 - 2. 関数の引数の部分にマウスを移動させ、ポイントします。



関数の説明を確認する

- 2. 関数のマケスを移動します(マピングエア上のライブラルペン内でこれらの機能を使用することができます)。



7.1.5 関数引数の追加、または、削除

コンテキストにより必要とされる通り必要な数量の、ラメーターを追加できるという意味でMapForce ビルトイン関数の一部は拡張可能です。このような関数のよい例は全ての文字列を連結を必要とする concat です。

(このような振る舞いをサポートする関数のための)関数の引数の追加、または削除方法:

・ パラメーターの横のパラメーターの追加(□) おさはパラメーターの削除(図)を使用して、パラメーターを追加、おさば削除します。



● シンボル上に接続をドロップすると、パラメーターが自動的に追加され、接続されます。

7.2 デフォルトとノード関数

MapForce かデータを読み取り、おけまデータを書き込む場合、ソース、おけま目的ファイル、おけまデータベースに空、おけま null フィールド か存在する場合がおります。このようよ場合を取り扱うっかに、MapForce には多数の内蔵の関数、if-else 条件、不足している、おけま null データを置き換える、おけま不足しているフィールドが検出されると例外をスローする他のメカニズムが搭載されています。

更に、複数のアイテムのよめのデフォルトの値を同時に設定する必要かある場合かあります(例えば、XML要素の全ての子)。代わりに、 空の値をテキスト(例えば、 "n/a") で置き換えるシンプルな関数を作成し、複数のアイテムにの関数を適用する必要かある可能性かありま す。通常の環境では、これを行うために、マレング上で同じ関数を複数回コピーして張り付ける必要かあります。これを行うと、マレングを乱 雑にするおけではなく、理解し難くします。簡単な代替とし、このチャプターの主題であるデフォルトとノード関数を使用することができます。

メモ マピングのターゲト 言語がBUILT-IN の場合、デフォルトとノード関数はサポトされています。このようなマピングを生成された C#、C++、Java プログラムコードから、おけま生成された XSLT/XQuery から実行する場合、変換はサポトされません。サー バー側では、MapForce Server Advanced Edition を使用してこのようなマピングを実行することができます。

用語 "ノード関数" は、関数をノードレベル(XMLノード、おけはCSV、JSON、EDI、おけはデータベースフィールド)で適用することができることを意味します。 ノード関数を単一のアイテム、おけは複数のアイテムコ 度に適用することができます。 同様に、用語 "default" は、単一のアイテム、おけは複数のアイテムのためにノードレベルに適用するデンオルトの値を意味します。

同じ処理ロジックを構造の中で子孫アイテムニ複数回適用する場合、デフォルトとード関数には特に役に立ちます。例

- 空、おけまnull 値が発生すると、他の値と置換え、全ての子孫アイテムに対して再帰的に行うことができます。
- 特定の値が発生すると、(例えば、 "N/A")、他の値と(おけ空の文字列と) 置き換えることができます。全ての子孫アイテムに対して再帰的に行うことができます。
- データベーステーブルから読み取る際に、データベース null 値を空の文字列、おけカスタムのテキスト(数値のフィー) いの場合は、
 0 の値とと置き換えることができます。
- ソースデータベースから来る全ての値のために末尾の空白を削除することができます。
- ターゲトファイル、おとはデータベースに書き込まれる全ての値にカスタムプレフィックス、おとなサフィックスを追加します。
- 特定の値が発生する都度、null 値を作成します。
- メモ "null"と"空の"値を区別することは重要です。null値は"nothing"(値が不足している状態)を意味し、空の値は空の文字 列(")を意味します。MapForce はノード関数を含む両方のケースに対応する多種の方法を提供しています。値は通常から の文字列(")です。MapForce は、ノード関数とデフォルトを含む複数の方法を与えます。

7.2.1 デフォルトとノード関数の作成方法

マルング上のまますべてのアイテム(ノード)のためコノード関数、おさまデフォルトを作成することができます。この過程をルールと呼ぶことこしましょう。

デフォルト、おイコメード関数を作成するオムリニ、デフォルト、おイコメード関数を定義する箇所でアイテム(ノード、おイコスィールド)を決定します。これは子孫を持たされ)、「リーブ、アイテム、おイコン子孫を持たされり、「リーブ、アイテム、おイコン子孫を持たされ アイテムであることができます。後者の場合、関数、おイコンデフォルトを全ての子孫アイテムにも適用することができます。条件を満たすアイテムを型、おイコン名前別に正規表現を使用して検索することができます。

必要条件:

- デフォルト、おイコンード関数をターゲトコンポーネントの入力側、おイコンパーネントの出力側に作成することができます。 作成する側を確立するイコン、次を参照してください、入力と出力サイドの選択。
- デフォルトとノード関数は、ソースとターゲット間の接続型は、"ソースを優先型" おけよターゲットを優先型"です。"全てユピー" 接続はサポートされません。具体的には、ノード関数とデフォルトは"全てユピー接続"に対して適用することができません。"全てユピー" 接続を持つノードは、単純な値を持つノード関数とデフォルトに対してのみ適用することができます。 例えば、単純型のコンテンソと

属性を持つ要素など。このため、関数を定義、おけは子孫を持つノード上でデフォルトを設定する場合、ソースとターゲト間の接続型はで、全てコピー、あってしばいません。接続型を確認、おけは変更する場合、接続を右クルクし、コンテキストメニューからターゲー優先(標準マッピング)を選択します。詳細に関しては、次を参照してください、接続の重類。

デフォルトの作成、おさまレートアイテム上のノード関数はファイルノードのためにサポートされません。このノードを使用すると、ファイルの作成と読み取りを動的に行うことができます。次を参照してください、複数の入力おさま出力ファイルを動的に処理。

デフォルト、またはノード関数の作成方法:

 アイテム(ノード)を右クトック、デフォルトとノード関数 | 入力デフォルトとノード関数 をコンテキストメニューから選択します。 (まけま、場合によっては、出力デフォルトとノード関数)。マッピング ペインはグリッドを上に表示します。まけま この場合、ユネ クタを右クトックすると、MapForce はそのサイドのナックコンード関数を表示します。マッピングペインは、上の部分に表示されま す。例:

コンボーネント 圖 <u>Text file</u> 内のアイテム D-Rows の出力側で定義されたノード関数とデフォルト:		
📝 祖先からルールを継承する		
適用 データ型 デフォルトの値/関数の説明		def
		f*
		×
	P	

ルールを定義する箇所に親が存在する場合、親に反対するルール(ノード関数、おゴはデフォルト)が存在する場合があります。このようなルールを継承する場合、祖先からルールを継承 チェックボックスを選択します。継承の詳細に関しては、次を参照してくださ、 デフォルトとノード関数のしくみ。

- 2. 以下の内のつを行ってくたさい
 - a. デフォルトを追加するコは、デフォルトを追加(mi)をクリックします。
 - b. 関数を追加するコよ、関数を追加(👫)をクルクします。

これにより新し、リレーンが作成されます(マッピングペインの上のブリンド内の行でこのレールのために条件を選択することができます)。ルールを以下のように構成します

適用	現在のアイテム、おうは全ての子孫アイテムに、深さて関係なく、おうは直接の子のみに ルールが適用されるかを決定します。
	ステップ1 で選択したアイテムに子孫か存在したは、場合、 "現在のアイテム" か唯一の 選択となります。
データ型	省略記号 ボタン・・・・・ をクトックし、ダイアログボックスから データ型を選択します。ルール (デンオルト、おイオノード関数) はこのデータ型(おイス、生成されたデータ型)を持つアイ テムのみい、適用されます。詳細に関しては、ノード関数とデンオルトを条件付きで適用す るを参照してくたさい。
	ステップ1 で選択されたアイテムに子孫が存在しては、場合、アイテムのデータ型は選択 された項目のみに適用されます。
デフォ ルトの値 / 関数 <i>の</i> 説明	デフォルトの(10) 型を定義する場合、選択されたアイテムのために設定するデフォルトの値(適用できる場合、は全ての子孫)をここに入力します。デフォルトとして空の文字列を設定する場合、このフィールドを空のままししておいてくたさい。

関数(「**)を定義する場合、このフィールドは情報のためのアメニ使用されます。この フィールドは関数の概要を表示します。次のステップ内の関数を定義することができま す。

3. 関数を定義する場合、マピングエアは関数の入力("raw_value"の下に示されると出力("result")を表示を表示するため、変更されます。このマピングエアは、ミニマピングであり、同じ一般的なルールが通常のマピングを定義する時にも適用されます。例えば、関数のボディは以下のようこなります。



上記のノード関数は、空の値を値 "n/a" と置き換えます。このサンプルの詳細に関しては、次を参照してくたさい <u>例: 空のCSV フィール</u> ドの置換。

以下の点に注意してください

- ノード関数内では、ビルトイン関数、編集、if-else 条件などのエのエンテキスト内で意味を成す特定のMapForce エンポーネントがサポートされます。XML、JSON、EDI、おゴはデータベースなどの複雑な構成はサポートされません。インラインのユーザー定義関数、おゴンジョインエンポーネントをノード関数に追加することもサポートされません。
- ノード関数は、最多で「つの入力」、ラメーターを持つ、おけお、ラメーターを持たないでしなことができます。入力、ラメータは常に「raw_value」と呼ばれます。関数のために必要なは場合でも、関数入力エンポーネント("raw_value")を削除しないでくたさい。それ以外の場合、マメニングの実行時にエラーが表示されます。編集の出力も同様です。メニューニマンドを使用して「関数 | 入力の挿入」偶然に削除された入力エンポーネントを復元する必要があります。
- 入力、ラメーター("raw_value") か既存の場合、後に追加される入力、ラメーターはメタデータ、ラメーター ゴンはます。このようなメタデータを使用して現在のノードに関する追加情報を取得し、関数の内部で使用することができます。 ノードメタデータをノード関数(1提供する)を参照してください。
- 一部の場合では、ノード関数を以下の方法で作成すると便利です: ライブラリウイドウから入力、おけは出力 ユネクタ ▷ にノード関数をドラッグする。これは<u>right-trim</u> おけは<u>uppercase</u> などの単純型の関数を使用する場合便利です。

ノード関数を終了する方法:

• マルングペインの左上の戻る = ボタンをクトック、おけば Escape」を押します。

既存のルールの編集と削除も参照してくたさい。

前に定義されたノード関数を確認、または変更する方法:

- 1. メインマ・ピング内で、注目するノードの横の「* アイコン(黒色おけ赤色)をクリックします。このアイコノはノード関数を以前に定 義した箇所に存在します。ノード関数アイコンの意味に関する詳細は次を参照してくたさい、デフォルトとノード関数のしくみ。
- 2. マッピング ペインの上のグリッドから関数を選択し、Edit をクリックします。Edit ボタンの存在しない場合、関数は、現在のアイ テムではなく、祖先によ)定義されている可能性があります(前のステップを参照してください)。

7.2.1.1 入力と出力サイドの選択

MapForce ユノポーネント は常に入力側と出力側が存在し、必要に応じてどちらかのサイドででノード関数とデフォルトの値を定義すること ができます。 更に理解を深めるために、 マノピングの壮組みについて思い出してくたさい、 最初にノースロレポーネントからデータを読み取り(例 えば、 データベース おけはファイル)、 (例えば、 関数、 おけはフィルターを使用して) データを任意で処理し (例えば、 データベース おけはファ イルなどの)ターゲットコレポーネント にデータを書き込みます。 この点を考慮して、 ノード関数を適用し、 異なるステージでデフォルトを設定しま す:

ソールファイル、おけまデータベースからデータか読み取られたすく後(マンピングにより処理される前)。例えば、下のマンピングでは、
 関数、おけまデフォルトは、ソースコンポーネントの出力側で定義されています(ノード関数、おけまデフォルトを示す 5. アイコンが存在することに注目してくたさい):



データがターゲトファイル、おさまデータベースに書き込まれるすく前(そして、全ての中間の処理が終わえた後)。例えば、下のマピングでは、関数、おさまデフォルトがターゲトコンポーネントの入力側で定義されています。



• マッピング処理内の中間のステージで、例えば、マッピングが複雑型の中間関数(XML 構造など)を含む場合、XML 構造に 提供される前、おける構造により返されたすく後にすべての値を削除することができます)。

上記を要約すると、コンポーネント上の入力、おけは出力、のとちらかでノード関数を定義することができます。入力側の関数(おけはデフォルト)は、コンポーネント上の対応するアイテムを入力する前にデータを処理します。他方、出力側で定義される場合、データを対応するアイテ
ムから返された後すくに処理します。関数が定義されている箇所でアイテムが子アイテムを持つ場合、デフォルトの関数を全ての子アイテムに対して適用することもできます。

7.2.2 既存のルールの編集と削除

デフォルト、おイコンード関数のためコレールがマピングに含まれている場合、注目されているノードの横に「キアイコン(黒色、おイオ赤色)と共に表示されています。以下に表示されているように、ルールを確認、変更、おイオ消除することができます。

確認、または編集のためにデフォルト、またはノード関数を開く方法:

1. メインマ・ピング内で、注目されているノードの横の「* アイエン(黒色、おけは赤色)をクトックします。このアイエンは、デフォルト、 おけまレード関数のためのレールが弦然定義されている個所に存在するノードを表しています。例えば、下のマ・ピング内で、ソー スエンポーネントのArticles アイテム、および、ターゲ・トエンポーネントのRows アイテムとでルールは定義されます。このデモマッ ピングは以下の、次で見つけることができます: 〈マイドキュメント〉¥Altova¥MapForce2021¥MapForceExamples¥ \OrderInUSD.mfd。



2. J* アイコンをクリックした後、このノード上で定義されている各 デフォルト、おイオノード関数のための行か含まれている上のグリッドを観察してくたさい。対応する行を選択し、必要に応じて変更を加えます。



3. 関数の場合、関数の実行 (マピング)を変更する場合、Edit ボタンをクリックします。Edit ボタンが存在しない場合、関数 は以前の祖先に定義されている可能性が高いです。この場合、ルーンが定義されている個所のアイテム上の fx アイエムカシノッ かされていることを確認してくたさい(下の)+を参照してくたさい)。

▶ アイコン はレールが定義されている個所を示しており、定義されている個所ではありません。アイコンの意味の詳細に関しては、デフォルトとノード関数のしくみを参照してくたさい。

ルールを削除する方法:

• 上にあるグリッドからルールを選択して、「関数 / デフォルトの削除」 × ボタンをクリックします。

メインマッピングを完了する方法:

• 「マッピング」ペインの左上の「戻る」 セ ボシをクリックして、「Escape」を押します。

7.2.3 デフォルトとノード関数のしくみ

<u>デフォルトとノード関数の作成方法</u>で説明されているとおり、ノード関数、おけはデフォルトをマピング上のまぼ全てのアイテム(ノード)のため に作成することができます。この過程をノレーノの定義と呼ぶことして、また。柔軟性のためコレールはは以下の重要な特徴があります。

- 継承、子孫を持つアイテム上のルールを定義する場合、このオプションを無効化しない限り、デフォルトで、ルールは子孫に継承されます。関数を定義する箇所でアイテムがネルされている子アイテムを持つ場合、直接子アイテムのみにコレールを適用する、または全ての子孫アイテムコレールを適用するかを決定することができます。
- データ型のフィルタリング、MapForceは各アイテムのデータ型に応じてルールを条件に従い適用することができます。これにより、 例えば、文字列の型の全てのアイテムのための特定のデフォルトの値(おさば関数)を適用、おさば数値型の全てのアイテムのため に異なるデフォルト(おさば関数)を適用することができます。

上記の振る舞いは特定の影響を与えます。具体的コよ ルールの定義と、ルールの適用を区別して理解する必要があります。一部のアイテ ムのためコルールを定義しても、そのルールがアイテムコ影響を与えるとは限りません。例えば、型 "文字列"のアイテム上でルールを定義し、 ルールの条件としてデータ型 "数値" を選択すると、ルールは適用されません。ルールの条件 (データ型と継承)が満たされた場合アイテム、ま たはその子孫にコレールが適用されます。

アイコン	詳細
fж	このアイエノよ(黒色)はこのアイテムのナーカンプレールが定義され、全ての子孫に影響を与える可能性があることを示しています。アイコンをクリックして、ルールを変更、おけ、胡鳴します。
f ₈	このアイコンは、(赤茶色)はアイテムが祖先レベルで定義されているルールは見込みがある、適用可能である)ことを示しています。すなわち、このアイテムに適用(おさは、影響)することができるルールは存在します。
fя	このアイコノよ(太字、赤茶色)は、このアイテムのナムリコレールが定義されている、おここのアイテムコレールが適用される ことを示しています。祖先からレールを継承するアイテムの場合この詳細があてはまり、同時に、アイテムのナムリコレールが定 義されていることを指します。
X e	このアイコノは、このアイテムに適用することのできるルールは意図的にブロックされていることを指します。 ルールを適用しない 箇所で特定のアイテムのためにこの操作を行うことができます。
f _×	このアイコノよ(グレーアナ)は、このアイテムに適用されるアイテムは無効化されています。例えば、このアイコンマンピングは接続されていたい、アイテムのために表示されます。

定義し適用するルールは関して理解するはよ、MapForceマピングのための視覚的なビートに注意してくたさい

通常、複数のノード関数、おけまデフォルトがつ、および、同じアイテムのためこ存在する場合、次のルールに注意してくたさい

アイテムに適用することのできるノード関数とデフォルトの数に関わらず、マンピング上の単一のアイテムのために、MapForce はつのソード関数、および、1つのデフォルトのみを適用します。

通常、以下のようコレールを解釈することかできます

- 同じアイテムのためは「複数のルール的存在する場合、MapForce はそのアイテムに近いルールを適用します。例えば、3回ノード 関数を定義したと仮定します: Company と呼ばれるルート XML ノードでは、 Department と呼ばれる子ノード、および、 孫 Employee。この場合、MapForce は近しため、 Employee アイテム上で定義されている Employee アイテム 関数 を適用します。関数が存在しない場合、直接の祖先 Department の関数を検索します。 Department のために関数が存 在しない場合、この場合 Company である ルートノードを更に検索します。継承は任意です。継承を無効化するは、 祖先 からルールを継承 チェックボックスを無効化します。このチェックボックスがやりアされていると、アイテムはは、プロック済みのルールベ メアイエム検索にされます。
- 同じアイテムご複数のルールが存在する場合、MapForce はマングペイン上の上のグリンドから最初のルールを適用します。このグリンドのルールの順序を変更するはは、ルールをクリング、グリンド内でルールをクリング、グリンド内の新しい箇所にボラングアンドドロップします。現在のアイテムのために定義されている場合のみグリンドのルールをドラングできることに注意してくたさい、継承されたシルールの位置は変更することができず、継承を有効化、おけば無効化することはできます。

仕組みを説明するこすと、次の、次で使用することのできるマピングを使用します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\MissingFields.mfd.



MissingFields.mfd

上記のとおり、このマンピングはレースXML ファイルからデータを読み取りターゲホテキストファイル(固定長フィーリド)に書き込みます。ソースXML ファイル内では、要素 Article には異なる型の次の子要素が存在します、"integer"、"string"、および "decimal"。かそれぞれの子要素は注意です (minOccurs="0")。このため、これらの要素がソースXML 内で存在したい場合、デフォルトの値を与える必要がある可能性があります。それ以外の場合、ターゲナ CSV ファイル内で空のフィールドが存在する場合があります。例

2	T-Shirt	25 2.3	Available in all sizes
3 4	Pants Jacket	57.5	Limited stock

下記にレールを使用して不足しているデータを扱う異なる方法、および、マノピング結果をルールかどのように影響するかについて説明されています。特定のアイテムに対して複数のルールが存在する場合、どのルールを管理するかについて理解することができます。

サンプル1:全ての文字列アイテムのためにデフォルトを提供する方法

マピング MissingFields.mfd 内で、次の条件が存在すると仮定します: Article の子カ型 "string"であり、不足している場合、デフォノルの値として "n/a" を使用します。

この条件を満たすためコよ、以下のステップに従います。

- 1. Article アイテムを右クトパし、デフォルトとノード関数 | 出力デフォルトとノード関数 をコンテキストメニューから選択します。
- 2. デフォルトを追加(1)をクルクします。
- 3. デフォルトの値の下に、"n/a"を入力し、「Enter」を押します。

・コンポーネント		
▼ 祖先からルールを継承する		
適用 データ型 デフォルトの値/関数の説明	<u>^</u>	÷
全ての子孫アイテム ▼ string … III デフォルトの値: n/a	f,	+ ×
	>	K I
	-	
G Articles		
日〇ファイル: MissingFields.xml <u>ファイル(文</u> \square 日〇ファイル(デフォルト) <u>ファイ</u> \square 日〇 Rows \square, ∞		
Article ArticleType [1∞] Field1 string [01]		
I> \bigcirc Number xs:integer [01] \blacksquare <		
Construction victoria [0.1] Construction victoria [0.1]		

上のマルングでは、ルールの条件を以下のように設定することができます: Article の全ての子孫アイテムで、データ型が "string"で、ソ ースXML 要素が不足している場合、デンオルトの値として "n/a" を使用します。このサンプルでは、型 "string" の2つのアイテムであ る Name と Discription が存在し、ルールは両方に適用されます。

上記のとおり、 ルールは定義される箇所で、アイテムの横に J* アイエムを表示されます。 ルールは定義される箇所で、アイテムの横に J* アイ エム表示されます。 この時点でマメピングをプレビューすると、出力内で不足して、 ケン文字列は、 "n/a" と置き換えられて、 ることこ気が付き ます:

	T-Shirt	25	Available in all sizes
2	n/a	2.3	n/a
3	Pants		Limited stock
4	Jacket	57.5	n/a

サンプル2: データ型をベースにデフォルトを条件付きで提供する方法

文字列アイテムのナメのデフォルトに加え、数値型のアイテムにデフォルトの値 0 を提供すると仮定します。この条件を満たすナメリゴよ、以下のステップに従います:

1. Article アイテムをクトックします。

2. デフォルトを追加(1)をクリックして、次の条件を持つ2番目のルールを追加します:

コンボーネント 倒 Articles 内のアイテム D Article の出力側で定義されたノード関数とデフォルト:		
☑ 祖先からルールを継承する		
適用 データ型 デフォルトの値 / 関数の説明		
全ての子孫アイテム ▼ string … 500 デフォルトの値: n/a		f.*
全ての子孫アイテム ▼ numeric … 四 デフォルトの値: 0		~
	-	
Image: Articles Image: Articles Image: Articles		
D Article Statistics of Standback 11 27/1/272 C		
= -O Number ssinteger [0.1]		
C Since Price vs demail 0.11 f. 1 1 1 1 1 1 1 1 1 1		
C Description xsstring [0.1]		

上のマピングでは、ルールの条件は以下のように設定することができます

- Article の全ての子孫アイテムのためにデータ型が「string"で、ソースXML 要素が不足している場合、デフォルトの値 "n/a"を使用します。
- Article の全ての子孫アイテムのためにデータ型が数値で、ソースXML 要素が不足している場合、デフォルトの値 "0" を使用します。

結果的には、概要は以下のようてなります

```
0 T-Shirt 25 Available in all sizes
2 n/a 2.3 n/a
3 Pants 0 Limited stock
4 Jacket 57.5 n/a
```

メモ "integer" と"decimal" データ型の両方を含むため、データ型 "numeric" は実際は型のカテゴリです。このため、ルールは Number とSinglePrice 要素の両方に適用されます。データ型とて "decimal" を選択すると、XML スキーマ型階層で は、型 "integer" は型 "decimal" からが派生するため、ルールは Number とSinglePrice の両方に適用されます。 ("XML Schema Part 2: Datatypes Second Edition"、<u>https://www.w3.org/TR/xmlschema-2</u>の S を参照し てくたさい。データ型 として "integer"を選択する場合、ルールは Number のみに適用されます。

サンプル3:特定のアイテムのためにレールをブロックする方法

前のアイテム同様全ての文字列と数値アイテムニデンォルトを適用すると仮定します。しかしなから、SinglePrice アイテムニデンォルトを適用しないとする場合以下を行います。

この条件を満たすためコは、アイテム Single Price をクトックして、チェックボックス祖先からルールを継承をクレアします。

Ð	コンボーネント 🗐 <u>Articles</u> 内のアイテム 🗅 <u>SinglePrice</u> の出力側で定義されたノード関数とデフォルト:		
	祖先からルールを継承する		
遁	用 データ型 デフォルトの値 / 関数の説明		dei
			f ₈ +
			×
		-	
1			
	間 Articles 回门 ファイル MissingFields xml ファイル文字列 string ロロファイル (デフォルト) ファイル文字列 string		
I	Image: Second second		
[$ = \mathbf{f}_{\mathbf{X}} $		
, I	\frown O Nome xstring [0.1] \downarrow_R \models \models Field string [0.1] \downarrow_R \models \models Field string [0.1] \downarrow_R		
1	C) SinglePrice xs:decimal [0.1] K Field4 string [01]		
1	C Description xsstring [01]		

上のマッピングでは、アイテム SinglePrice は親 Article からルールを継承しません。このため、 "ブロック済みのルール" アイコンメ か表示されます。

この結果、対応するフィールドは結果内で空白のまま表示されます

0	T-Shirt	25	Available in all sizes
2	n/a	2.3	n/a
3	Pants		Limited stock
4	Jacket	57.5	n/a

サンプル4:特定のアイテムのための継承されたルールをオーバーライドする方法

全ての文字列に数値を提供すると仮定します。しかしながら、アイテムSinglePriceのみ、9999を提供するとします。

この条件を満たすために、以下のステップに従ってくたさい

- 1. アイテム Single Price をクリックします。
- 2. デフォルトを追加(感)をクリックし、9999のデフォルトの値を入力します。
- 3. 任意で、祖先からルールを継承 チェックボックスを選択します。

€] ⊐ンポ-	ネント 🝓 <u>Articles</u> 内の	アイテム ▷ <u>SinglePrice</u> の出	力側で定義されたノード関数とデフォルト:		
🔽 祖先からル	ールを継承する				
適用	データ型		デフォルトの値 / 関数の説明	*	
現在のアイ	テム 💌 decimal	🜆 デフォルトの値:	9999		f.+
	string	🜆 デフォルトの値:	n/a		× <
	numeric	🜆 デフォルトの値:	0		\sim
				*	
😫 Articles			Text file		
ロロファ	イル: MissingFields.xml	ファイル/文字列 string	日日ファイル (デフォルト) ファイル/文字列 string		
in A ()⊟ ⊂i	cles restriction of xs:any	Type [01]			
	Number xs:integer [01]	j. fi	= Field string [0.1]		
⊳ ()	Name xs:string [01]	f	Field3 string [01]		
	SinglePrice xs: decimal [0	1] f	R Field4 string [01]		
	Description xs:string [0	I. I.			

メモ 継承されたししつの背景は黄色です。

上のマピングでは、アイテムSinglePrice にて起用することのできる3つのレールが存在するとます:2つの継承されナーレールと、1つの直接的なルールが存在します。この場合、アイテム上で直接定義されているルールが優先されます。継承されナールは無視されるため、概要は以下のよう」さいます。

0	T-Shirt	25	Available in all sizes
2	n/a	2.3	n/a
3	Pants	9999	Limited stock
4	Jacket	57.5	n/a

サンプル5: 適用するルールを手動で選択する方法

前のサンプルを更に展開し、アイテム SinglePrice のためにもう1 つアイテムを定義します。8888 がデフォルトの値です。上記のとおし、現在のアイテムに直接定義されているルールは優先されます。しかしなから、(継承されたルールに加え) 2 つのルールが現在のアイテムに存在します。問題 は つのデフォルト 8888 まけは 9999 のどちらが適用されるかです。

🕣 コンボーネント 🗐 <u>Articles</u> 内のアイテム 🗅 <u>SinglePrice</u> の出力側で定義されたノード関数とデフォルト:		
▼ 祖先からルールを継承する		
現在のアイテム ▼ decimal		6193 f +
現在のアイテム 🔽 decimal 🖉 デフォルトの値: 9999	=	Jx
string 🖽 デフォルトの値: n/a		~
numeric 5日 デフォルトの値・0	~	
A Articler		
国内 Andres 国内 Andres 国内 アイル (デフォルト) ファイル (アフォルト) (アート) (P-1)		
C Articles restriction of xs:anyType [01]		
↓ □ () Article ArticleType [1∞] fx ↓ → → Field1 string [01] ↓		
B Image: Sinteger [0.1] Image: Sinteger [0.1] Image: Sinteger [0.1]		
O Description xsistring (01)		

上のマンピングのように同じアイテムに対して複数のルールが存在する場合、優先されるルールをグルドの上に手動でドラッグします。一番上 に表示されているルールは優先されます。このため、SinglePriceのデンオルトの値は、グルドの上に表示されている8888です:

0 T-Shirt 25 Available in all sizes 2 n/a 2.3 n/a 3 Pants 8888 Limited stock 4 Jacket 57.5 n/a

7.2.4 ノード関数とデフォルトを条件付きで適用する

ノード関数、おはオデフォルトの(しサカゆび ルー・リ)を作成すると、子孫を持つアイテム上、おけば リーフ」 アイテム上で定義することができま す。重要な点は、アイテム:子孫が存在すると、ルーンのためにデータ型を指定しなければないません。これは、このデータ型の子孫のみがルー ルによ影響を受けることを意味します。この概念を更によく理解するために、次のデモマンピングを開いてくたさい、 <マイドキュメント >\Altova\MapForce2021\MapForceExamples\\OrderInUSD.mfd。ターゲルコンポーネント内の「行」アイテムの横の「* アイコンをクトックしてくたさい。



OrderInUSD.mfd

上のマピングは XML ファイルからデータを読み取り、ターゲオ CSV ファイルに書き込みます。ターゲオ CSV コンポーネト内では Rows アイテムの入力側でルールが定義されています。 Rows アイテムコは 型 string とinteger の子が存在します。 ルールのデ ータ型は string であるため、ルールはこの型のアイテムのみ影響します。 すなオム、「**アイコンは型 string のアイテムのためにつか存 在しますが、型 integer のためコン存在しません。 メデアイコンは、一部のアイテムのためココルールがブロックされていることを意味し、この ファケトは現時点では無効です。 デフォルトとノード関数のしくみに関する詳細は、を参照してくたさい。

ビント: 上のマンピング内で各アイテムの横にデータ型を表示するコは「データ型の表示」 ビ ツール いやお をオイン切り替え「注 釈の表示」 ジール いやお をオオン切り替えます。「表示」メニュー内でこれらの切り替えコマンドを使用することができま す。

ルールのデータ型を変更するコま、マノピングの上のグリンド内の「省略記号」ボタン 200 をクリックします。これにより、必要とする型を選択することのできるダイアログボックスか開かれます。

関数

😢 フィルターノード関	数とデフォルト		×
このダイアログボックス 加フィルターオプション ができます.	により、条件付きで) に従い、関数または	アイテムのデータ デフォルトを適用	型と追 引すること
この型のアイテムに関 string numeric decimal	腰、または、 デフォル	トをこの型のアイ	(74
integer date dateTime time gDay gMonth gMonthDay gYear gYearMonth duration boolean binary QName			
	ノード名		~
1史糸×1家(四);	 大文字と小文² 正規表現(R) 	字を区別(C	
	OK	中心	ッセル

「フィルターノード関数をデフォルト」ダイアログボックス

例えばデータ型とてintegerを選択する場合、(型integerの唯一のアイテムであるけっか)ルールはアイテム Amountのみに適用 されます。このアケトを示すけっかこことポーネトは再度描かれます。 Amount 以外の他のアイテムは、型 string であるけっかルールに と見影響を受けていたいてとご注意してください。

		DrderInUSD
		□] ファイル: (テフォルト) <u>ファイル/文字列</u>
Н	⊳ f _×	
Н	× K	= Company string [01]
Ч	× K	= Article string [01]
Ч	► f _x	SinglePrice string [01]
Ч	•	= Amount integer [01]
Н	► f _×	Price string [01]

ソールのデータ型をノード関数内で意味が対す場合のみ変更します。デモOrderInUSD.mfdマパング内で、ノード関数は2つ

の文字列を結合する 数値ではなく、string データを処理するようにデザインされています。この結果、このマピングはノード関数の データ型をinteger に変更すると、失敗します。

型の継承

MapForce はXML データオナロオよく、データ型およびノード関数とデフォルトをフィルター」ダイアログボックス上での選択のためは使用 できるデータ型とも作業することができ、XML スキーマ型に正確に対応する必要はおりません。一部は型のカテゴルである場合もおり、これ は、型の広範囲の選択に一致することを意味します。例えば、型 string は、normalizedString、token、NCName、 NMTOKEN、IDREF、ENTITY などから派生した。string 多種の他のデータ型に一致します。同様に型 decimal は、派生した型 integer、long、short などに一致します。

型の階層構造はXML スキーマW 3C 勧告に従います。階層構造を説明するダイアグラムは関しては、 S in "XML スキーマパート 2: データ型第2エディション"、 https://www.w3.org/TR/xmlschema-2 を参照してくたさい。

高度なフィルタレング

任意で、更に高度な条件をベースコンたりレールを適用することもできます

- 特定の型 AND であり、ノード名がカスタムのテキストの一部に一致する場合、ルールを適用します。
- 特定の型 AND であり、型名がカスタムのテキストの一部に一致する場合、ルールを適用します。

例えば、下の設定は、ノード型がstring ANDであり、ノード名が「Title」の場合のみりレールが適用されます。右のイメージは、ノード「Title」のみが一致し、このために「*アイエンが取得されるなど」このようなルールが没に立つ場合のサンプル構成を示しています:

 マイルターノード関係 このダイアログボックス(加フィルターオプション) ができます。 この型のアイテムに関い string numeric decimal integer date dateTime time gDay gMonth gMonthDay gYear gYearMonth duration boolean binary 	数とデフォルト × により、条件付きでアイテムのデータ型と追 に従い、関数またはデフォルトを適用すること 数、または、デフォルトをこの型のアイテム	People File: People.txt File/String strin Company string [01] Company string [01] Company string [01] First string [01] Last string [01] Fitle string [01] Shares string [01] LeaveTotal string [01] LeaveTotal string [01] LeaveTotal string [01] Company string [01] First string [01] Company string [01] First string [01] Fi	
QName 「フィルター(F): 検索対象(M):	ノード名 ✓ Title □大文字と小文字を区別(C) □正規表現(R) OK キャンセル		

おけよ、適するする複数のノード名おけは型名に一致する正規表現を使用することができます。例えば、下の設定は、名前が"total"で終わるすべての型 string アイテム ご適用されます。正規表現が使用されていることご注意してください。正規表現 .+total 文字通り、 以下を意味します: テキスト "total"に続く1回まけは複数回の文字の発生を一致させる。また、検索で大文字と小文字を区別することができ、 "SOME TOTAL" と "Some Total"の双方が一致することを意味します。 右のイメージは (具体的には、3つの "Total" アイテムが 同様 コレールに適し 5. アイコンを取得するなど このようなレールが役に立つかもしれないサンプルの構成を示しています:

♡ フィルターノード関数とデフォルト ×	People
このダイアログボックスにより、条件付きでアイテムのデータ型と追 加フィルターオブションに従い、関数またはデフォルトを適用すること ができます。	\Box File: People.txt File/String string string
この型のアイテムに関数、または、デフォルトをこの型のアイテム string numeric decimal integer date dateTime time gDay gMonth gMonthDay gYear gYearMonth duration boolean	Department string [01] First string [01] East string [01] Title string [01] EMail string [01] EMail string [01] EleaveTotal string [01] UsedTotal string [01] EleftTotal string [01]
QName QName ⑦ フィルター(F): ノード名 検索対象(M): .+total ① 大文字と小文字を区別(C ⑦ 正規表現(R) OK キャンセル	

正規表現大アレケトは XML スキーマのものと同じです。正規表現を参照してくたさい。以下の点に注意してくたさい

- アノカーへと\$ は暗黙であり、「検索対象」ボックス内に入力する必要はありません。
- 「大文字と小文字を区別」チェックボックスにより大文字と小文字は区別されます。ですから、フラグはサポートされません。
- ノードフィルタレノブのオーガン複数の行を一致させることは意味をなしません。ですからm フラグはサポートされません。

7.2.5 ノードメタデータをノード関数に提供する

現在のノードに関する情報をベースして何かを行うされて、ノード関数が必要な場合があります(「ノードメタデータ」とこの情報を呼ぶこと 「しましょう」。例えば、次のロジックを持つノード関数が必要と仮定します:ノード名に「Total」とう単語が含まれている場合、ノードの値 「ゴルのシンボルが追加されます。それ以外の場合、ノードの値がそのまま返されます。

サンプルでも記述されていますが、「ノード名」はノードメタデータのサンプルです。一般的には、「メタデータ」はデータ自身を説明する何かを 意味します。すなわち、こればデータに関するデータ」を意味します。「ノードメタデータ」により、数値のデータ型や他の場合、ノード名、値の 長さ、まけよ精度なと関数が適用されるノードに関する他の情報を理解することができます。

次のテーブルは、ノード関数内で使用することのできるすべてのメタデータをリストしています。下にリストされるメタデータは、特定の種類のノードにオヨカの決意味があります(例えば、XML、オヨコキデータベースフィーリド)。この結果、現在のノードと互換性の内メタデータを使用しようとすると、MapForceは警告を表示します。

node_name	現在のソードの名前を提供します。このメタデータをすべてのソードに適用することができます。 XML の場合、現在の要素おさは属性の名前です。 CSV の場合、これはCSV フィーリドの 名前です。 データベースの場合、テーブル列の名前です。
node_annotation	「注釈の表示」 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
node_minLength	ノードのデータ型のminLength ファセナの植を提供します。適切な型を持つXMLとテキストノードに適用することができます。
node_maxLength	ノードのデータ型のmaxLength ファセナの植を提供します。適切な型を持つXMLとテキストノードに適用することができます。
node_totalDigits	ノードのデータ型のtotalDigits ファセルの値を提供します。適切な型を持つXMLノードに適用することができます。
node_fractionDigits	ノードのデータ型のfractionDigits ファセトの値を提供します。 適切な型を持つXML ノードに適用することができます。
node_length	ノードのデータ型の長さを提供します。適切な型を持つフィールドに適用することができます。
node_precision	ノードのデータ型の精度を提供します。適切な型を持つフィールドに適用することができます。
node_scale	ノードのデータ型のスケールを提供します。適切な型を持つフィールドに適用することができます。

ノード関数にメタデータを提供する方法:

1. 新規のノードの作成を開始(デンカルトとノード関数の作成方法を参照)、おけよ既存のノードを編集(既存のレールの編集と削 除を参照)のためは開きます。例えば、下で示されている関数は、ノード値を持つ文字列 "\$" を連結して、外部のマメピングの結 果を返します。この関数のマメピングを次の いな使用して見つけることができます: くマイドキュメント >\Altova\MapForce2021\MapForceExamples\OrderInUSD.mfd。関数のマメピングを開くには、ターゲナイコンポ ーネナのRows アイテムの横の 「* アイコンをクリック、グリッド上のEdit」 ポタンをクリックします。

Ð	Node functions and defaults defined on input side of item [> Rows in component I OrderInUSD:		
🗹 In	herit rules from ancestors		
100 C	Apply to	Data ty	Default value / Function description
f.t	All descendant items 💌	string $f_{\mathbf{x}}$ Node function:	concat("\$", raw_value)
~			
\sim			
	J		
-	æ∐ raw_value raw_value		concat
	Add Node Specifics	Dval D	uez

2. 次の、ずれがま行してくたさい

- •「ノードの指定の追加」をクリックします。
- マピング内の空のエリアを右クトックし、「入力の挿入」をコンテキストメニューから選択します。
- •「入力の挿入」 🖬 ツール デーボタンをクリックします。
- •「関数」メニューから「入力の挿入」をクトックします。

😢 Insert Input with Node Specifics	\times
Nodes can supply various static properties to node functions via metadata parameters. Choose a <u>m</u> etadata parameter to insert:	
node_name node_annotation node_minLength node_maxLength node_totalDigits node_fractionDigits node_length node_precision node_scale	
Description: Provides the static (displayed) name of the node. Supported by all nodes.	<u>^</u>
When metadata is not supported by a node: Return empty sequence from input Do not apply the node function 	·
OK Cancel	

- 3. ダイアログボックスから必要とされるメタデータ(例えば "static_node_name")を選択します。
- メモ 挿入するメタデータ、ラメータを選択すると、MapForce は、マメピング上のノード関数が既に適用する条件を満たしている個所で現在展開されているノードを分析し、メタデータ、ラメータがごれらのノードによりサポートされるかを決定します。サポートされていない場合、ダイアログボックスは「選択されナメタデータ、ラメータはこの関数のスコープ内の既存のノードによりサポートされていない場合、ダイアログボックスは「選択されナメタデータ、ラメータはこの関数のスコープ内の既存のノードによりサポートされていない場合、ダイアログボックスは「選択されナメタデータ、ウメータはこの関数のスコープ内の既存のノードによりサポートされていない場合、「加え似した警告を表示します。メモルを保存しユーザーエクスペリエンスを構造するナムウェデンオルトでは、深くネストされた構造はフルレスキャンされません。ノード関数を適用する箇所でコンポーネントがごのような深くネストされた構造を持つ場合、 MapForce が認識できるようにマンピング上で関連するノードを展開することができます。この場合、新規のメタデータ、ウメータを追加する際に、MapForce は展開されナノードを考慮し、警告が表示される可能性があります。ノード関数を適用するはおお、ため、存在する必要がりますが、接続されていない、アイテムの展開は関連性がありません。
 - 4. メタデータが現在のノードにサポートされていない場合、関数の振る舞いを次のように決定することができます。
 - a. ノード関数を適用する場合、チェックボックス「入力から空のシーケンスを返す」を選択し、メタデータ・デメータに空のシーケンスを返すようにすることができます。空のシーケンスは空の文字列と区別される必要かあります。substitute-missing

おけまexists などのシーケンス関数、おけ地のエンポーネントを処理するために使用する必要が通常あります。警告: 空のシーケンスが扱われる必要があります。それ以外の場合、ノード関数は植を全く返され、場合があります。

- b. このメタデータがシードにサポートされなくノード関数を適用しない場合、チェックボックス「ノード関数を適用しない」を選択 します。
- 5. 「OK」をクリックします。デフォートのraw_value に加え、関数のマッピングに新規の入力・デメータが追加されました。新規のパ ラメータの出力のエネクタをこのメタデータ通常は関数の入力コネクタを必要とするターゲットアイテムは接続します。

7.2.6 例: 空の CSV フィールドの置換

このサンプルは、CSV ファイルからデータを読み取り他のCSV ファイルにデータを書き込むMapForce マメングの作成方法を説明して します。カスタムの値("n/a")を持つノース CSV ファイルからすべての空のフィールドを置き換えることがこのサンプルの目的です。ソース CSV ファイルは以下のようこなると仮定します:

```
H,111,332.1,22537.7,,Container ship,,,
D,111,A-1579-227,10,3,400,Microtome,,
D,111,B-152-427,7,6,1200,Miscellaneous,,
H,222,978.4,7563.1,,Air freight,,,
D,222,ZZ-AW56-1,10,5,10000,Gas Chromatograph,,
```

希望するマピングの出力は以下のようてなります。

H,111,332.1,22537.7,n/a,Container ship,n/a,n/a,n/a D,111,A-1579-227,10,3,400,Microtome,n/a,n/a D,111,B-152-427,7,6,1200,Miscellaneous,n/a,n/a H,222,978.4,7563.1,n/a,Air freight,n/a,n/a,n/a D,222,ZZ-AW56-1,10,5,10000,Gas Chromatograph,n/a,n/a

次の 次でこのサンプル内で作成されたマンピングを見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\ReplaceEmptyFields.mfd。マンピングのナックのノースCSV ファイルは、Orders.csv と呼ばれ、同じファイル内にあります。ターゲオ CSV ファイルは MapForce により生成されます。

マルングの目的を達成するために、存在するそれそれの空の値を "n/a" と置き換える単一のノード 関数を作成します。 下に示されるように、この関数は1度のみ定義されますか複数の子孫 CSV フィールドに適用されます。

ステップ 1: ソース CSV ファイルをマピングへ追加する

ソースCSV ファイルをマピングに以下のように追加することができます:

- 1. 「挿入」メニューから、「テキストファイル」をクルクします。
- 2. (MapForce Enterprise Edition のみ) オプション「標準 CSV(カンマ区切りまたは FLF(固定長)のファイルに単純な 処理を使用 フィールド」を選択して、「継続」をクリックします。
- 3. 「入力 ファイル」をクトックして、次のファイルを選択します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Orders.csv。

チェックボックス「空のフィールドを存在しないものとして扱う」が選択されている場合、クリアしてくたさい。チェックされている場合、このチェックボックスは、空の値を抑圧し、ノード関数が作動しません。詳細に関しては、次を参照してくたさい、CSV オプションの設定。

- 4. 「OK」をクリックします。
- 5. "Orders" 」、対するエンポーネトを変更するようこプロンプトされると、希望するオプションをクリックします(例えば、コンポーネント名を変更しない)。

MapForce のCSV コポーネト詳細、関しては次を参照してけさい CSV とテキストファイル

ステップ 2: ターゲナ CSV ファイルをマピングへ追加する

ターゲオ CSV ファイルをマピングに以下のように追加することができます:

- 1. 「入力」メニューから「テキストファイル」をクルクします。
- 2. (MapForce Enterprise Edition のみ) オプション「標準 CSV(カンマ区切りまたは FLF(固定長)のファイルに単純な 処理を使用 フィールド」を選択して、「継続」をクトックします。
- 3. ターゲナ ファイルはノースと同じ数のフィールドを有する必要がみます。このため、9つのフィールドを追加するために「フィールドの追加」ボタンを複数回クトックしてくたさい。
- 4. 「OK」をクルクします。

ステップ 3: マピング 接続の描写

この時点では、マピングロは次の2つのエレポーネトか管まれています: ソースCSV ファイルとターゲト。ソースエレポーネト上の Rows アイテムの横の出力コネクタ ^D クトックし、ターゲトコレポーネント上のRows アイテムの入力コネクタ ^D はデラックします。これを 行うと、設定に応じて MapForce は自動的に全ての子孫アイテムを接続し、しかゆる、 "全てコピー"接続を作成する可能性があります。 一致する子要素を自動接続する マール・オブションが有効化されている場合のみ上記が行われます。前記のとお人ノード関数

一致する子要素を目動接続するトー・リールトーオンタンか有効化されている場合のみ上記が行われます。前記のとおり、ノート関数 ば、全てコピー、接続の子孫に適用されません。このため、「全てコピー、接続は最初に通常の接続に変更される必要がみます。これを行う には、ソースのField1 から、ターゲトのField1 に接続します。プロンプトされると、接続の置換をクリックし、全てコピー接続を解決する をクリックします。

```
一致する子要素を自動接続する目まオプションが有効化されていない場合、ソースとターゲオ出力間の接続を以下のように作成します
```

- 1. ソース内のRows アイテムをターゲナ内のRows アイテムと接続します。
- 2. 接続を右クトックし、一致する子要素を接続するをコンテキストメニューから選択します。
- 3. 全てコピー接続を作成するチェックボックスをクリアします。
- 4. 「OK」をクリックします。

マピングは、以下のようなります



ステップ 4: ノード 関数の作成

データがシースを離れてすぐに、ませまターゲトに書き込まれる前に、ノード関数を作成することができます。このサンプルの目的のため、ター ゲトコンポーネントの入力側でノード関数を作成します。これは、基本的には、「ターゲナトにデータが書き込まれる前」を意味します。詳細に 関しては、次を参照してくたさい、入力と出力サイドの選択

ターゲナコンポーネナト上のRows アイテムを右クリックし、「デフォルトとノード関数 | 入力デフォルトとノード関数」をコンテキストメニュー から選択します。「マッピング」ペインの上に空のブリンドが表示されます。

コンボーネント 国 <u>Text file</u> 内のアイテム Rowsの出力側で定義されたノード関数とデフォルト:		
📝 祖先からルールを継承する		
適用 データ型 デフォルトの値 / 関数の説明		def
		f*
		×
	~	

次に、グリバの左の関数を追加 👫 ボタンをクリックします。マノピングは関数の入力 ("raw_value") 出力 ("result") を表示します。

- コンポーネント 国 <u>Text file</u> 内のアイテム ▷ <u>Rows</u> の出力側	定義されたノード関数とデフォルト:		
▼ 祖先からルールを継承する			
適用 データ型	デフォルトの値 / 関数の説明		तल
全ての子孫アイテム ▼ string 🛛 🛄 ∫ ノード関数	出力は接続されていません		f,*
			×
		-	
1			
2 raw_value		😥 result	
raw_value[2		Presult	

前記のとおり、関数の目的は、空の値を文字列 "n/a" に置き換えることです。これを達成するために、マルングに追加コンポーネントを追加します:

- MapForce ビルトイン関数 empty。引数とて提供された値が空の場合、この関数はtrue を返します。それ以外の場合は false を返します。ライブラリウィドウから関数をマナピングドボラッグ、おけはマナピングをダブルクトックして、"empty"を入力しま す。マナピング1、関数を追加するた参照してくたさい。
- テキストの値 "n/a"。値を追加するコよ、マピング上の空のエリアをダブルクトックし、2重引用符に囲まれた "n/a"を入力します。 次も参照してください、マンピングニ定数を追加する。
- If Else 条件。マピングに追加するコは If Else 条件(:)ツールレーボタをクリックします。このようコンポーネントの詳細に関しては、次を参照してくたさい、例:条件付で値を返す。

これらのエレポーネトを使用して、関数をデザインをします



上記のデザインは、以下の用に作動します:最初に、外側のマピングからの入力値は、raw_value 入力を使用して関数を入力します。行の値は、empty 関数に入力として提供されます。If-Else コンポーネント はempty 関数により返されたブール関数の結果を評価します(true おけはfalse)。結果がtrue の場合、定数 "n/a" は関数の結果にないます。結果がfalse の場合、関数の行入力の値は、関数の結果にないます。関数のresult ("n/a" おけはraw_value) が外側のマピングから返されます。

Exit をクリックして、おさよ Escape を押し) 関数の編集のエリアを終了します。



上記のマピング内で、以下の点に注意してくたさい

- ウイドウの上の部分のテキストは関数が定義された箇所を示します。これは同じコンポーネトのために複数のノード関数が定義されて、る状況では特に役にたちます。
- グリンド内の適用 オプションは全ての子孫 アイテムに設定されています。このサンプルでは、これは期待される振る舞いです。すなわち、Rowsの全ての子孫アイテムが条件を満たす場合影響を受けます。関数は親アイテムコ度のみ定義されていますが、マンピング上で確認できるように、ターゲトコンポーネントの左側(入力)は複数の 5% アイコンを表示します。
- データ型 オプションは "string "に設定されています。このサンプルでは、テキストデータを作業しているため、これは期待される振る 舞いでありまた、デフォルトの振る舞いでもあります。
- 編集 ボタイこより、必要に応じて関数の定義に戻り変更することもできます。このボタンカ表示されていない場合、「* アイコンを 最初にクリックしてください。

ステップ 5: マピングの実行

MapForce 内でマピングの結果をプレビューするコよ 出力 ダブをクトックします。検証エラーが発生すると、メッセージ ウイドウ内に表示 されます。成功するとCSV 結果が出力ペイン内に表示されます。次を参照してくたさい、マノピングの検証。

MapForce Server Advanced Edition を使用して、次の方法でサーバーマシン上でこのようなマピングを実行することもできます。

- MapForce Server Advanced Edition スタイプロンを使用する場合、マピングを実行可能ファイルコンパルして、ター ゲトマンノコピーしてくたさい。次を参照してくたさい、MapForce Server 実行可能ファイルコマノピングをコンパルする
- FlowForce Server とMapForce Server Advanced Edition にライセンスが供与されている場合、FlowForce Server に直接マビングをデプロイレ、予定通り、お台オンデマイドとしてジョブが作動するように構成します。次を参照してくたさい、マビングをFlowForce Server にデプロイする。

7.3 ユーザー定義関数

ユーザー定義関数(UDF)は一度定義されるカスタム関数で、同じマンピング、おけは複数のマンピング内で複数回使用することができます。 ユーザー定義関数自身はかさなマンピングと類似しています。通常、一つおけは複数の入力パラメーター、データを処理するけざの中間パラメ ーター、および、と呼び出し元にデータを返すけざの出力によい構成されています。呼び出し元はメインマンピング、おけは他のユーザー定義関 数であることができます。

ヒント: 複数の出力を返すユーザー定義関数を作成することは可能です。関数が「インライン」と定義されているとこの機能はサポートされます。 <u>インラインと正規ユーザー定義関数</u>を参照してくたさい。

再使用の可能性とお別に、マメピングの部分を小さ、ロンポーネントに、シケージ、および、実装の詳細を無視し、結果、マメピングを簡単に 理解するオンタンコーザー定義関数は役に立ちます。



通常、文字列、数値、日付、および、他のデータを内蔵のMapForce 関数を拡張しカスタム化して処理するためにユーザー定義関数を 作成します。例えば、特定の方法で結合、おけよ分割する、おけは高度な計算を実行、時刻を操作、再利用できるようにマッピングの一部 をパッケージするなどの場合が挙じたれます。ユーザー定義関数の他の一般的な使用方法は、MapForce によりサポートされる他のファイ ル、データベース、おけはデータストア内のフィーリビを検索することです。

以下のマピングは、文字列を2つの異なる文字列に分割する、ユーザー定義関数のサンプルです。このユーザー定義関数は次のデモマピングの一部です: <マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥ContactsFromPO.mfd。パラメーター (例えば、 "Helen Smith") としての名前を取り、

ビルイノ関数 substring-before とsubstring-after を適用し、2つの結果する値 ("Helen" と"Smith") を返します。



ContactsFromPO.mfd

上記のとおり、ユーザー定義関数をメインのマッピング、おコはユーザー定義関数から呼び出すことができます。 すなわち、ユーザー定義関数 を必要であれば、以下に示されるように、 ネスト することができます。



例えば、下のマンングはユーザー定義関数(「LookupPerson」)を呼び出し、XML ファイル内の個人の名前を検索します。 「LookupPerson」エンポーネントのヘッダーをダブルクトックすると、マンピングウイドウ内の定義が開かれ、この関数は次のユーザー定義関数を呼び出していることに気が付くはずです: "EqualAnd" と "Person2Details"。このマンピングはデモとして使用することができ、次のパスで見つけることができます: <マイドキュメント

>\Altova\MapForce2021\MapForceExamples\PersonListByBranchOffice.mfd.



PersonListByBranchOffice.mfd

ユーザー定義関数を再帰的に呼び出すことができます(すなわち、ユーザー定義関数が自身を呼び出します)。これは、ユーザー定義関数 を(インラインではなく)関数正規関数として定義する必要があることを意味します。<u>インラインと正規ユーザー定義関数</u>を参照してくたさい。



再帰的なユーザー定義関数により、N か事前に既知ではなく、N 個の子の深さを持つデータの構想を反復するなど、多種の高度なマピングの必要条件を解決することができます。例 再帰的な検索を参照してください。

ユーザー定義関数の作成後、作成した同じマピング内に保存されます。しかしたから、他のマピングにインポートし、そこから呼び出すこともできます。詳細に関しては、ユーザー定義関数の呼び出しとインポートを参照してくたさい。

7.3.1 ユーザー定義関数の作成

ユーザー定義関数の作成を始めから作成、おけオマパング上に既存のエレポーネトの選択から作成することができます。

ユーザー定義関数を最初から作成する方法:

1. 「関数」メニューカム「ユーザー定義関数の作成」をクリックします。おけよ 代わりに、「ユーザー定義関数の作成」(ご) ツール ディジンをクリックします。

ユーザー定義関数を作成 ×
設定 関数名(F): LookupPerson ライブラリ名(L): user
説明 構文(S): LookupPerson(OfficeName, First, Last)
詳細(D): Altova_Hierarchial.xml 内の社員に関する詳細な情報を検索します。
実装 ✓ インラインを使用(I) 「インラインを使用」により、MapForce はこの関数が使用される場所なら どこでも関数のコンテンツを取得する様にします。これにより生成されるコ ードは長くなりますが、スピードは多少速くなり、1つの関数に複数の出力 を持たせることができます。 この関数を再帰的に呼び出したい場合は、「インラインを使用」を無効に してください。複数の戻り値を持たせたい場合は、例えば複数の要素を 持つ XML 構造を使用することができます。
OK キャンセル

2. 必要なフィールドに清報を挿入します(下のテーブルを参照してくたさい)。

関数名	必須のフィーリド。作成するユーザー定義関数のためこ名前を入力します。有効な文字:	
	 英数字文字 (a-z、A-Z、0-9) 下線(_) ハイフン/点線(-) コロン(:) 	
ライブラリ名	必須のフィールド。関数が属するへきライブラバを入力します。関数は <u>ライブラリウィドウ</u> 内でこの ライブラバイで表示されます。ライブラ応指定しない場合、関数は "user" とう名前のデフォル トのライブラル内に配置されます。	
構文	任意のフィールド。関数の構文を的確に詳細するテキストを入力します。(例えば、期待される ハラメーター)。このテキストは、ライブラリウイドウ内の関数の横に表示され、関数の実行にコよ影響を与えません。	

詳細	任意のフィールド。関数の任意のテキストの詳細を入力してください。このテキストは、ライブライウィ ンドウ内で、おさま地のコンテキスト内でカーノルを動かすと表示されます。
インラインされた使用	関数がインライン関数として作成される場合、このチェックボックスを選択してくたさい。正規関数を 作成する場合は、チェックを解除してくたさい。詳細に関しては、 <u>インラインと正規ユーザー定義</u> 関数を参照してくたさい。

3. 「OK」をクリックします。上記で指定されているライブラノ名の下のライブラノウイドウ内で関数がすくに表示されます。列:

ライブラリ	🔹 🗘	×
🔁 core		\wedge
_ ⊟ math functions		
floor	result	
<u>∎</u> ⊖string functions		
translate	result	
LookupPerson	Looku	
I		~
loo	×	•
ライブラリを追加/削除		
塗 ライブラリ 🛅 プロジェクト		

おっ、マピングウイドウが有度描き直され、新規の関数を作成することができます(これは "関数のマピング"として参照されるスタイアロンマピングです)。関数は出力を必要とするため、関数のマピングにはデフォルトで出力コンポーネントか含まれてします。

user.LookupPerson	ERAJE
	*
result result	

左上の角の「メインマッピングに戻る」(
いいの)、おなくこり、関数のマッピングからメインマッピングに戻ることができます。関数のマッピングを開くためには、ライブラリウィドウ内の関数をダブルクルクします。詳細に関しては、ユーザー定義関数の 呼び出しとインポートとユーザー定義関数のデビゲートを参照してくたさい。

- 4. 関数の定義により必要とされるすべてのエレポーネントを関数のマンピングに追加します。標準のマンピングと同じ方法でこれを行うことができます。例えば、入力ませま出カノ・ラメーターを追加するまは、次の、「ずれかを実行してください」
 - メニューコマド「関数 | 入力の挿入、お」」 「関数 | 出力の挿入」 をそれぞれ実行します。
 - マピング エ 「たち クリックし、「入力の挿入」 おけは「出力の挿入」 をコンテキスト メニューから選択します。
 - •「入力の挿入」(🖬) おさば「出力の挿入」(🖬)ツールレーボタンをクリックします。

少なくとも、関数はデータが接続される1つの出力コンポーネントを必要とします。入力パラメーターは関しては、関数は、ゼロ、おけおつ以上の入力を持つことができます。入力おけは出力パラメーターは、単純型(文字列、おけは整数など)、おけは複合型(構成)であることができます。単純型、および、複合型のパラメーターは関する詳細は、ユーザー定義関数内のパラメーターを参照してください。

ユーザー定義関数が作成されましたが、おき使用されていません。関数をマピング内で使用するコよ、ライブラリウィドウから関数をメ インマッピング エリアイ ボラッグします。 ユーザー定義関数の呼び出しとインポートを参照してください。

ユーザー定義関数を既存のコンポーネントから作成する方法:

- 1. マウスを使用して正方形を作成し、マンピング上で複数のエンポーネントを選択します。「Ctrl」キーを押したまま、それそれをクリックして複数のエンポーネントを選択することかできます。
- 2. 「関数」メニューから「選択からユーザー定義関数を作成する」をクルクします。代われに「選択からユーザー定義関数を 作成する」(¹⁰⁰) ツール デーオシンをクルクします。
- 3. 上記のステップ 2-4 を繰り返します。

7.3.2 ユーザー定義関数内のパラメーター

ユーザー定義関数を作成する場合、(必要な場合) どの入力/ ラメーターが取られるか、および、出力が返されるかを指定する必要があります。入力/ ラメーターが必要とされない場合がありますが、出力/ ラメーターはすべての場合に必要とされます(すなオオム、関数は常に何かを返す必要があります)。例えば、下の関数には入力は存在しませんが、呼出元にテキスト「hello」を返す1 つの出力が存在します:



関数パラメーターは単純型(文字列、おけ整数などの)おけ複合型の構成です。例えば、下に表示されるユーザー定義関数 "FindArticle" では、2つの入力としつの出力パラメーターが存在します。

- POArtNr は単純型 "string" の入力パラメーターです。
- Amount は単純型 "integer" の入力パラメーターです。
- CompletePO は複合型 XML の出力パテメーターです。



LookupArticle.mfd

この上のマピングはデモ上て次のパマで見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\LookupArticle.mfd。

パラメーターの追加

入力または出力パラメーターを追加する方法:

- 1. ユーザー定義関数 マピングを作成、おけお既存のものを開きます(ユーザー定義関数の作成を参照してください。) おけよ (ユーザー定義関数の編集を参照してください)。
- 2. 次の、ずれかを実行してくたさい
 - メニューコマンド「関数 | 入力の挿入」まけは「関数 | 出力の挿入」を実行します。
 - 入力の挿入 👬 おけは 出力の挿入 🖬 ツール いちゃんたりかんます。

🙄 入力の作成			×
名前(N): ShortPO			
ェ 〇単純型(integer,	string など)(S)		
データ型 (D):	string ~		
◉ 複合型(ソリー構	造)(C)		
構造(U):	:2019¥MapForceExamples¥ShortPOlxsd	選択 <mark>(H)</mark>	編集
ルート <mark>(</mark> R): [ShortPO	選択 <mark>(0)</mark>	
E	☑構造ファイルパスをMFDファイルから相対的)に保存 <mark>(V)</mark>	
☑入力は必須(1)			
☑ 入力はシーケンス(E)		
		ОК	キャンセル

3. 上のダイアログボックスで、入力おさは出カノ・ラメーターが文字列、おさは整数などの)単純型(XMLおさよ、データベース構成などの)複合型の構成であるかを選択してくたさい。複合型のXML型である、ラメーターを作成するためには、「構造」の横の 「選択」をクリックし、必要とされた構成を説明する XML スキーマを参照してくたさい。

関数のマッピングはXML スキーマを既に含む場合、構成として選択のために使用することができます。それ以外の場合、ハラメーターの構成 を適用する新規のスキーマを選択することができます。 MapForce のエディンコンによりサポートされる場合、データベース まけよ地の複合 型の構成も同様に扱われます。

XML構成を使用すると、XMLスキーマか許可する場合、構成のためコレト要素を選択することができます。ルト要素を指定するため はよ、「Root」の横のを「選択」し、開かれるダイアログボックスからルト要素を選択します。

選択された場合、チェックボックス「MFD ファイルから相対的に構成ファイル・マを保存」はファイルを保存する際、構成ファイルの絶対 パマを現在のマッピングに対して相対的ないマ変更します。詳細に関しては、
<u>コンポーネントとで相対いてを使用する</u>を参照してくたさい。

「入力は必須」と「入力はシーケンス」チェックボックスは次のセグションで説明されています。

必須のデメーター

ユーザー定義関数内で、デメーターを必須にするコよ「入力は必須」チェックボックスを選択します。パデメーターカ泌須の場合、入力が接続されていたい場合検証エラーカ発生します。

ハラメーターを任意にするコは「入力は必須」チェックボックスのチェックを解除します。メインマンとつでは、任意のパラメーターコは若干 異なる外観(入力コネクタ(小さな三角形)の点線の境界線で表示)が存在します。

ハラメーターの「default」入力に接続することにより、デフォルトの、ラメーターの値を指定する必要があります。例

	∺ ∐ input	De result
C= Come default value1	a dafa ult in a ut	be a coult
Some default value in	- derauit input	result

デフォルトの値はその他の値が存在しては、場合のみ適用されます。関数が呼び出される際に、任意の、ラメーターが価を受け取ると、この値は、デフォルトの値に対し優先されます。

シーケンスノラメーター

関数の ラメーターカ単一の値(デフォルトの振る舞い)、おけよシーケンスとして扱われるように任意で指定することができます。パラメータの入 力値を単一の値と異なり、シーケンスとして扱うには「入力はシーケンス」チェックボックスを選択します。このチェックボックスは役に立ち、ユ ーザー定義関数が型「正規」に設定されている場合のみ有効化されます。 ハンラインと正規ユーザー定義関数を参照してくたさい、それ以 外の場合、チェックボックスは無効化されます。

シーケンスはゼス、おさはゼロ以上の値の範囲です。ユーザー定義関数が入力データをシーケンスとして期待する場合、そのシーケンス内で 値の集計を行うさま、パラメーターの入力値をシーケンスとして扱う必要がある場合があります(例えば、avg, min, max などの関数を呼び 出す場合などか挙げられます)。例えば、デモマノビングを開いてください、くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\InputIsSequence.mfd。このマピングでは、「data」フィルターはユーザー 定義関数「Calculate」に接続されています。フィルターの出力は、アイテムのシーケンスで、関数の入力パラメーターイギンーケンスに設定さ れています。



InputIsSequence.mfd

内部では、「Calculate」 関数はすべてのシーケンス値が集計されます(下で示されるようにmin、max とavg を実行し入力シーケンス上の関数を集計します)。



一般的には、シーケンス、おさはキシーケンスに関わらず、入力データはどのように関数が呼び出されるかを決定します。

- sequence / デメーターに入力データが接続されている場合、ユーザー定義関数1度のみ呼び出され、完全なシーケンスかユー ザー定義関数内についてされます。
- non-sequence パラメーターに入力データが接続されている場合、ユーザー定義関数はシーケンス内の各単数のアイテムのためにそれぞれ呼び出されます。

空のシーケンスを非シーケンスパラメーターは接続すると、パラメーターは関数が呼び出されないままの結果を取得します。

これは、ソース構成が任意のアイテムを持つ場合、おゴコンハルターの条件が一致するアイテムを返され、場合に発生します。これを回避する には、シーケンスが空になられ、ことを保証するために、関数を入力する前に<u>substitute-missing</u> 関数を使用し、パラメーターをシーケンス に設定し、関数内の空のシーケンスのために処理を追加します。

「出力はシーケンス」チェックボックスは、出力、デメータのために必要とされています。関数が複数の値のシーケンスを出力エンポーネントに、パマレ、出力エンポーネントがシーケンスに設定されていない場合、関数は、シーケンス内の最初のアイテムのみを返します。

パラメーターの順序

ユーザー定義関数が複数の入力ませ出力、ウメーターを持つ場合、この関数の呼出元に表示される、ウメーターの順序を変更することが できます。例えば、下の関数には3つの、ウメーター(input1、input2、およびinput3)が存在します。



関数のマパング内の上から開始するハラメーターの順序は、この関数の送信元か表示される順序を指定します

ustom_function		
≥ ₩i input1	🍽 result 🖓	
>> input2		
¦> ₩≣ input3		

以下の点に注意してくたさい

- 入力と出力、ラメーターは上から下に位置が並べ替えられます。このため、、ラメーターinput3 を関数のマパングに移動すると、 この関数の、ラメーターは最初になります。
- 2つの デメーターか同じ垂直の位置にある場合、左側の デメーターか優先されます。
- 2つの らメーターか同じ位置にある場合、内部コンポーネント ID か自動的に使用されます。

7.3.3 インラインと正規ユーザー定義関数

ユーザー定義関数には2つの種類が存在します: インラインと正規関数。関数を作成する際に、関数がインライン、ませま正規関するである かを指定することができます。 ユーザー定義関数の作成を参照してくたさい。 インラインと正規関数は、コード生成、再帰性、複数の、ラメー ダを持つことができる能力などで異なる振る舞いをします。

インライン関数	正規関数
インライン関数は、生成されたコード内で発生する箇所ですべての インスタンス内から抽出することができ、コードは長くなりますが、若 干速くなります。	各ユーザー定義関数コンポーネントは、入力が デメーター コ なされる個所で、関数の呼び出しのためニコードを生成し、出力は植を返す関数 (コンポーネント)
インライン関数は、大幅に生成されたプログラムコードを増加することができます。ユーザー定義関数コードは、関数が呼び出されるす べての場所で実際に挿入されます。このため、正規表現とは異な り、コードのサイズが増加します。	ランタイムでは、すべての入力/ ラメーターカ最初に評価され、関数 が入力データの各発生のナムカニ呼び出されます。
インライン関数は、複数の出力を持つことができ、このため、複数の 値を返します。	正規関数は つの出力のみを持つことができます。複数の値を返す には、複数の値を発信者につなすることを許可する出力を複合型 (例えば、XML 構成に宣言することができます。

インライン関数	正規関数
インライン関数を再帰的に呼び出すことはできません。	正規関数を再帰的に呼び出すことは可能です。
インライン関数は ラメーター上で優先コンテキスト 優先コンテキスト を設定することをサポートしません。	正規関数は、パラメーター上で優先コンテキストの設定をサポートします。

ユーザー定義関数を「ヘライン」から正規につい替える、おけまその逆の場合、マッピングエンテキストに影響を与える場合かあり、これ はマッピングが異なる結果を生成する可能性を引き出します。

マピング上で、 インライン ユーザー定義関数は、 点線の囲いにと表示されています。 例えば、 下のマピング内の中間エンポーネトはインライン ユーザー定義関数です。



正規関数は、実線の囲いにお表示されています。例えば、下のマンピングの中間コンポーネトは正規のユーザー定義関数です。



7.3.4 ユーザー定義関数のナビゲート

マピングにユーザー定義関数が含まれている場合、各ユーザー定義関数とメインマピングの間を下に示されているように簡単にナビゲートすることができます。

ユーザー定義関数を表示、または編集のために開く方法:

• マピング上のユーザー定義関数のタイトルドーをダブルクトックします。

custom_function	
l≥ ⊭≣ input1	🍽 result 🖗
⊳ ₩ input2	
¦> ₩∥ input3	

• ライブラリウィドウの特定のユーザー定義関数をダブルクリックします。

ライブラリウイドウの名前をダブルクトックして関数を編集することができます。現在アクティブよドキュメント内の関数をこの方法で開くことができます。他のマッピング内で作成されたユーザー定義関数をダブルクトックすると、新規のウイドウ内にそのマッピングを開きます。

メモ 複数のマピングにインポートされたユーザー定義関数を編集ませる
期除する場合、この変更によりすべてのインポートされるマピングか影響を受けます。

メインマッピングに戻る方法:

• メインマッピングに戻るコよ、マッピングウィボウの左角の「メインマッピングに戻る」(50) ボタンをクリックします。

更に、ユーザー定義関数を含む多種のMapForce タブをナビゲートするとその履歴が保存されます。使用したタブ間を前後に移動するこは、「戻る」(シンと「次へ」(ジンツールドーボタンをクリックします。これらのボタンコン対応するキーボードのショートカナトは「Alt+Left」と「Alt+Right」です。

7.3.5 ユーザー定義関数の編集

ユーザー定義関数を編集する方法:

- 1. ユーザー定義関数を含むマピングを開きます。
- 2. マパング上のユーザー定義関数のタイトルバーをダブルクリックします。コンテンンを表示するかかに、必要に応じてコンポーネントを追加、編集、おけ、期除することのできるマパングウイドウか変更されます。
- 3. (名前おは詳細などの関数のプロ、ティを変更するために、以下の一つを行ってくたさい
 - マピングの空白の部分を右クリックし、コンテキストメニューから、関数の設定」を選択します。
 - ●「ユーザー定義関数設定」(□)ツール デーオタンをクトックします。

ライブラリウィンドウの名前をダブルクトックして関数を編集することができます。現在アクティブムドキュメント内の関数をこの方法で開くことができます。他のマンピング内で作成されたユーザー定義関数をダブルクトックすると、新規のウィンドウ内にそのマンピングを開きます。

メモ 複数のマピングにインポートされたユーザー定義関数を編集おけば削除する場合、この変更によりすべてのインポートされるマピングが影響を受けます。

7.3.6 ユーザー定義関数の削除

マピング上のユーザー定義関数のタイトルィーをダブルクトックしてしてユーザー定義関数を削除します。



1. マピングウィボウの右角の「削除」 🎎 ボタンをクリックします。

2. 現在開かれているマピング内で関数が使用されている場合、ダイアログボックスか表示されます。

MapForce		×
?	このユーザー定義関数の代わりに、split-name (ライブラリ:user)が 以下の場所に存在します: - メインマッピング	
	内部コンポーネントのインスタンスを全て置換えますか?	
	はい(Y) いいえ(N) キャンセル	

関するを削除し、関数のコンポーネントと共に呼び出されているすべてのインスタンスを置き換える場合、「しよい」をクリックします。こ れにより、関数が削除される場合でも、メインマンピングを有効に保ってとができます。削除された関数が他の外部のマンピングで使 用されている場合、これらは無効しています。

関数とその内部コンポーネトを永久に削除する場合は、「いいえ」をクリックします(この場合、関数が使用されるすべてのマッピングは無効」ないます)。

7.3.7 ユーザー定義関数の呼び出しとインポート

ユーザー定義関数の作成後、作成した同じマピングから呼び出す、おけは他のMapForce マピングから呼び出すことができます。

ユーザー定義関数を同じマッピングから呼び出す方法:

1. ライブラリウィボウ内で関数を検索します。関数の作成時に指定したライブラルの下に関数が表示されます。デフォルトのユーザ ー」ライブラを作成した場合、「ユーザー」ライブラル内で関数を検索してくたさい。名前別に関数を素早く検索するコよ、ライブラ リウィボウ内に名前を入力します。

ライブラリ	🔹 ф	×
🔁 core		\wedge
_ → math functions		
floor	result	
<u>∎</u> =string functions		
translate	result	
💼 user 📼 LookupPerson	Looku	
		v
loo	×	•
ライブラリを追加/削除		
塗うイブラリ プロジェクト		

2. ライブラリウイボウからメインマンビングに関数をドラッグします。必要とされるすべての、ラメーターを接続することができます。関数 の結果は出力、ラメーター(ませよ、適用可能な場合複数の、ラメーター)によい適用されます。

ustom_function		
≥ + input1	🍽 result 🖗	
⊳ ⊮∥ input2		
P∾≣ input3		

ユーザー定義関数を他のマッピングからインポート する方法:

1. ライブラリウィボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウィンボウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラノとして関数をインポートする場合 グローノ シルライブラリのインポートの横の「追加」 ボタンをクトックします。
- メモ ライブラをローカルでインポートする場合、ライブラリファイルの ひをマルングに対して相対的に設定することが可能です。グローバ ルニインポートされたライブラノでは、インポートされたライブラノへの ひは常に絶対的ない ひです。
 - 3. ユーザー定義関数が含まれるマッピングファイル(.mfd)を参照し、「開く」をクリックします。新規のライブラル追加されたことを通知するメッセージボックスが表示され、新規のライブラルライブラリウイズウに表示されます。

MapForce X
A new library has been added. See Messages window for details.
If you don't see newly added functions in the Libraries window, please make sure that the correct language is selected in the Output menu.
Don't show this message again.
ОК

ライブラリケンドウからマッピングバボラッグして、現在のマッピング内のインポート済みの関数を使用することができます。<u>関数をマッピングに追加する</u>も参照してくたさい。

ライブラリケイドウゼンルトイン関数と現在のマメピングファイル内のユーザー定義関数を表示します。上に示されるようご他の、mfd ファイルを ライブラルとて現在のマメピングにインポートすると他のインポート済みのファイルからのユーザー定義関数も表示されます。複数の*.mfd ファ イルますコカスタムライブラノ全体で同じライブラノ名を使用すると、ライブラリケイドウ内で同じライブラノ名の下に表示される使用可能なすべ てのノースからの関数が表示されます(カスタム Java と.NET ライブラノのインポートを参照してくたさい)。

関数ライブラル表示と整理に関する詳細は関数ライブラル管理を参照してくたさい。

7.3.8 マッピング間で UDF をコピーし張り付ける用法

以下のようこマピング間でユーザー定義関数を簡単にコピーして貼り付けることができます

1. ライブラノウイドウのベースのライブラリの追加/削除をクリックします。ライブラルの管理ウイドウか開かれます。
- 2. ライブラリウイボウ内の空のエリアを右クリックしオプロン「全ての開かれているドキュメントを表示」を選択します。
- 3. ソースと目的マピングを両方開きます。例えば、下のイメージではノースはBuildHierarchyRecursive.mfd で、目的の マピングはNew.mfd です。
- メモ ソースとターゲットマンピングが既にディスタに保存されていることを確認してくたさい。これにより正確ないなが保証されます。相対的ないなのなど一と長い行行を参照してくたさい。
 - 4. ソースマッピングファイルからのユーザー定義関数を右クリックし、コンテキストメニューからコピーを選択します。(おさは「Ctrl+C」 を押します)。

Manage Libraries ×				
📮 🛃 BuildHierarchyRee	cursi	ve.mfd	*	
🕀 User-Defined Functio	ns			
📟 📲 user.BuildFile	Hier	archy		
Own Library Import.		Locate Function in Libr	arie	s window
📮 🗑 New.mfd	u	- ·		61.1 M
User-Defined Functi	ሙ	Cu <u>t</u>		Ctrl+X
Own Library Import.	₿ <mark>₽</mark>	<u>C</u> opy		Ctrl+C
Υ Global Library In	×	<u>D</u> elete		Delete
	~	Show All Open Docum	ents	;
4		Show <u>F</u> ile Paths		

5. ターゲナトマンプファイルの「ユーザー定義関数」エトリを右クリックし、コンテキストメニューから「張り付け」を選択します。

関数ライブラノの管理も参照してくたさい。

7.3.9 例: ルックアップと連結

ユーザー定義関数の一般的な使用方法を説明したよくつかのデモマングがMapForceで使用することができます。 PersonListByBranchOffice.mfd ファイルの1 つはくマイドキュメント> Altova MapForce2021 MapForceExamples ¥ フォルダー内で見つけることができます。



PersonListByBranchOffice.mfd

マピングコお以下の必要条件がみます

- XML ファイルからデータを抽出し、ターゲオ XML ファイルに書き込みます。データは氏名など従業員の詳細から構成されています。
- 個別のXML ファイル(電話、電子メールアドレス、職位)内で個別の従業員に関するデーダを検索します。
- ターゲトに書き込む前に希望する方法でデータを処理します。具体的には、電話番号、電子メール、職位が単一の文字列で表示されており、ターゲト XML の詳細 要素に書き込まれています。
- XML 要素、この場合、特定の部署からの従業員に関する情報のみを抽出します。マピングの発信者はオフィス名をコマイ・ラインで、ウメーターとして指定できる必要があります。例えば、マピングがMapForce Server により実行されている場合など。

上記の必要条件を実装するエレポーネトを検証してみましょう

- マメピング("OfficeName")の入力、ウメーターはシンプルな入力コンポーネントです。デフォルトの値("Nanonull, Inc.")は 定数により提供されます。この値はマメピングの呼出元が、ウメーターの値を提供しては、場合使用されます。マメピングロン・ウメータ <u>一を提供する</u>を参照してください。
- 特定のオフィスコ属する従業員をフィルターするために、マピングはフィルターコンポーネト("Office")を使用します。基本的には、フィルターは、ラメーターにとり提供されているオフィス名がノース XML ファイル内のオフィス名と等価かどうかをチェックします。等価の場合、フィルターはノース Office アイテムからデーダをターゲナコンポーネントにつ やします。フィルターは関する詳細はフィルターと条件を参照してくたさい。.
- 2番目のノースXML ファイルからの情報を検索するコよ、マンピングはユーザー定義関数 "LookupPerson" を呼び出します。この関数のロジッグは以下で詳細に説明されています。
- デーダを処理するために、「LookupPerson" 関数は内部で適切な方法で各従業員に関する情報を抽出し連結する他の関数 を呼び出します。これらすべての操作は巻数の自身のマメピング内にあり、メインマメピング内では表示されません。これは一般的な カプセリかの例です。、「LookupPerson" 関数は Details 要素をターゲオ XML 内で作成します。

ルッケアップの実行

ルックアップ機能は機能について以下で説明されている"LookupPerson" 関数には提供されています。



"LookupPerson" function

上記の通り、関数にはデータを抽出することのできるソースXML ファイルが含まれています。次に、ルックアップ値を提供する3つの入力パラメ ーターが存在します: Office_Name、First_Name、およびLast_Name。捨てべの入力パラメーターは必須として設定されます(す なけち、チェックボックズ「入力が必要とされています」がプロパティダイアログボックス内で選択されています)。

"EqualAnd 関数" は現在のものに囲まれている個別のユーザー定義関数です。この機能はブール値を返します。上記のシーケンス内での この関数の呼び出しは以下のブールのコジックを説明しています:



関数の値(TRUE おけまFALSE)は新規のアイテムが処理される都度フィルターイン やされます。フィルターが値 TRUE を得ると、ルック アップオペレーションは成功し、従業員の詳細が抽出され外部のマメピングに返されます。それ以外の場合、コンテキスト内の次のアイテムが 検査され、ループが完了するまで継続されます。

さしよの "EqualAnd" 関数の発生では、コネクタ b は丸枠で囲まれてお上 これはこの レメーターが優先コンテキスト とて設定されて いることを示しています。 優先コンテキストはマピングの実行を最適化する任意の機能です。 具体的には、特定のオフィスの個人情報が入 カパラメータ a により提供され、最初に処理されます。 優先コンテキストとして ジメーターを設定するには、右クルクして、コンテキストメニュー から 優先 を選択します。 詳細に関しては <u>優先コンテキスト</u>を参照してくたさい。

実行の連結

The "Person2Details" 関数は"LookupPerson" 関数にネストされている他の関数です。この関数は単一の値を返します。以下で説明されているとおり、パテメーターとして受け取するつの値とつのテキスト定数を連結します:



"Person2Details" function

concat 関数はビルトイン関数です。必要とする数量の、ラメーターを取ることのできるMapForce ビルトイン関数です。<u>関数の引数の</u> 追加または削除を参照してくたさい。

マピングの実行

MapForce 内のマピングの実行を北ビューするコは「出力」ダブをクリンクします。マピングはデフォルトの入力、ウメーター (「Nanonull, Inc.」)を使用して実行され、この結果このオフィスからのみ従業員の情報を抽出します。他のオフィスからデータを抽出するこ は、入力、ウメーターに背う俗されている定数を「Nanonull, Inc.」から「Nanonull Partners, Inc.」に変更しマピングを再度実行します。

ライセンス供与済みのMapForce Server を使用することができる場合、Linux、macOS、お台はWindows マシン上のコマイトライン 上でマピングを実行することができます。最初に、メニューコマイドファイル| MapForce Server 実行可能ファイルコンパイル」を 使用してMapForce Server 実行ファイル(.mfx) にマピングをコンパイルします。<u>MapForce Server 実行可能ファイルにマピングを</u> <u>コンペイルする</u>を参照してください。次に、.mfx ファイルをサーィーマンイニピーし、以下のコマイドでMapForce Server を実行します。 名前が付けられたノデメーター -p=OfficeName は入力パテメーターを提供します:

mapforceserver run PersonListByBranchOffice.mfx -p=OfficeName:"Nanonull, Inc."

灹

- mapforceserver はMapForce Server 実行可能ファイルのなで、使用中のオペレーティングシステムに適用可能です。
- 適用できる場合 .mfx ファイルン ひを変更します。おけよ .mfx を実行可能ファイルと同じフォルダー ことします。

7.3.10 例:再帰的な検索

このサンプルは、再帰的なユーザー定義関数の助けを使用してノースXML ファイル内でデーダを検索するマピングを表しています。マピングファイルは次の、なで見つけることができます: くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\RecursiveDirectoryFilter.mfd。



RecursiveDirectoryFilter.mfd

ソース XML ファイルコよ 下のコードリストに示されているとおり、ファイルとディレクトリニ関する情報が含まれています(リスティングではデータの) 一部が省略されていることに注意してくたさい):

```
<?xml version="1.0" encoding="UTF-8"?>
<directory name="Examples">
<file name="ExampleSite">
<file name="blocks.sps" size="7473"/>
<file name="block_file.xml" size="992"/>
<directory name="output">
<file name="block_file.xml" size="992"/>
<directory name="output">
<file name="block_file.css" size="3174"/>
<file name="images">
<file name="images">
<file name="blank.gif" size="88"/>
<file name="block_file.gif" size="13179"/>
</directory>
</directory>
</directory>
```

```
ソースXML ファイル
```

ソースとターゲト XML ファイルは同じスキーマ Directory.xsd を使用します。ファイルシステム上で、デルノトリはファイルおけま地のデル ケーを含むことができ、これはスキーマ内で反映されます。重要な点は、スキーマが「ディレクトリ」要素が再帰的であることを指定していることです(ライン <xs:element ref="directory"/> を参照してください。。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"</pre>
attributeFormDefault="unqualified">
   <xs:element name="directory">
      <rs:complexType>
         <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="file">
               <re><xs:complexType>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="size" type="xs:unsignedLong"/>
               </ms:complexType>
            </ms:element>
            <xs:element ref="directory"/>
         </xs:choice>
         <xs:attribute name="name"/>
      </ms:complexType>
   </r></r></r>
</xs:schema>
```

Directory.xsd

マピングのビジネスの必要条件は、特定の拡張子を持つファイルのみをフィルターします。すべてのディレクトリのネストされた構造は、保存される必要があります。例えば、拡張子が、.xml、の場合、(ソース XML ファイルが以前にリストされているため)期待される出力は以下に類似します:

```
<?xml version="1.0" encoding="UTF-8"?>
<directory name="Examples">
<directory name="ExampleSite">
<file name="block_file.xml" size="992"/>
```

```
<directory name="output">
        <directory name="images"/>
        </directory>
        </directory>
</directory></directory></directory></directory></directory></directory></directory></directory></directory></directory></directory>
```

期待される XML 出力

2番目に、マピングの発信者は、ハラメーターとしてファイル拡張子を提供する必要があります。デフォルトで、発信者が、ラメーターの値を提供しない場合、マピングは拡張子を持つ.xml ファイルをフィルターアナトします。

上記の必要条件を対処するために、マンピングにはテキスト定数を介してデフォルトのファイルの拡張子を提供する単純型入力パラメーター、 "SearchFor"、か含まれています。このパラメーターは任意です(「入力は必須」チェックボックスはプロパティダイアログボックス内で選択されていません):

🙁 Edit In	put	×
<u>N</u> ame:	SearchFor	
<u>D</u> atatype:	string	~
	Input is required	
Design-ti	me Execution	
Spec	ify value	
<u>V</u> alue:		
	OK Cancel	

入力、ウメーターは関する詳細は、マノピングレン・ウメーターを与えるを参照してくたさい。

次に、マンングロココーザー定義関数、"FilterDirectory" か含まれています。この関数は再帰的で、自身への呼び出しも含んでいます。 再帰的な要素の「ディレクトリ」に接続されているすか、この関数はノースXML インスタンス内のネストされた「ディレクトリ」要素を必要な だができます。再帰的な呼び出しをサポートするすかに、この関数は、インライン関数でしおく、正規関数として作成されていま す(「インラインの使用」オプションは、関数のプロ、ティ内で選択されていません)。関数のプロ、ティを確認するこは、マノヒング内の空のエリ アを右クトックし、コンテキストメニューカら「関数の設定」を選択します。。ユーザー定義関数の編集を参照してくたさい。



上記で示されるように、関数は入力として2つの、ラメーターを取ります:

- 1. 検索される XML 構成を定義する複合型パラメーター、Directory、(この) ラメーターは「haystack」です)。
- 2. 検索するファイル拡張子を指定する文字列/ ウメーター、SearchFor、(この) ウメーターは「needle」です)。

マピング上の入力ませま出力パラメーターのタイトルバーを右クトックして、設定を確認します。

関数には、MapForce ビルトイン関数 contains が接続されているフィルターコンポーネトか含まれています。 contains 関数は 検索の値がシース構成内の "name" 属性 (ファイル名) に一致する場合、 true を返します。 true の値は フィルターに現在のアイテム を出力にコピーするようこ命令します。 それ以外の場合、 スキップされます。 フィルターに関する詳細は、 <u>フィルターと条件</u>を参照してくたさい。

関数のdirectory パラメーター(入力と出力の双方) 同様、マメピングの、ソースとターゲオファイルは同じスキーマ Directory.xsd を 持ちます。 MapForce は、これらの型の割り当ては整合性があると検知し、入力パラメーターと関数間の接続の型は「全てユピーする」 で す。 <u>全てユピー接続</u>を参照してください。

マピングの実行

MapForce 内で、マピングの実行をプレビューするコは「出力」ダブをクリックします。デフォルトの入力、ラメーター(".xml")を使用して マピングは実行され、この結果、この検索結果に一致する結果のみが抽出されます。異なる検索の条件を提供するコは、入力、ラメーター へ接続されている定数を".xml" から ".sps" に変更し、マピングを再度実行します。

ライセンスが提供されたMapForce Server を使用できる場合、Linux、macOS、おけよWindows マシン上のコマイドラインでマピ ングを実行することができます。最初に、メニューコマイ「ファイル| MapForce Server 拡張ファイルにコンパルする」を使用して、 MapForce Server 拡張ファイル(.mfx) にマピングをコンパルします。おこ、<u>MapForce Server 拡張ファイルにマピングをコンパ</u> <u>ルする</u>を参照してください。.mfx ファイルをサーバーマンノニピーし、下のコマイドを使用して MapForce Server を実行します。名前付 いたれたい デメーター -p=SearchFor は入力値を提供します:

mapforceserver run RecursiveDirectoryFilter.mfx -p=SearchFor:".sps"

灹

- mapforceserver は使用中のオペレーティングシステムご適用することのできるMapForce Server 実行可能への文です。
- .mfx ファイルへの なを適用できるように変更、おけま実行可能と同じフォルダー内に.mfx をエピーします。

7.4 カスタム XSLT 1.0 または 2.0 関数のインポート

XSLT 1.0、XSLT 2.0 とXSLT3.0 関数ライブラを MapForce 内で使用することのできる単純型を返すカスタム関数として拡張する ことができます。

簡単なデータ型(例えば、文字列)を返すカスタム関数のみかりポートされます。

XSLT ファイルから関数をインポートする

1. ライブラリウィボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウィンドウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベリビ) グローノ ジレライブラノとして関数をインポートする場合 グローノ シレライブラリのインポートの横の「追加」 ボタンをクリックします。
- メモ ライブラをローカルでインポートする場合、ライブラリファイルの ひをマピング に対して相対的に設定することが可能です。 グローバルにインポートされたライブラリでは、 インポートされたライブラレヘの いは常に絶対的ない いです。
 - 3. 関数か含まれる.xsl ファイルを参照し、「開く」をクリックします。メッセージボックスか表示され、新規のライブラル追加されている ことを通知します。

・ クポートされたXSLT ファイルは、ライブラリウィドウ内のライブラル表示され、全ての名前を持つテンプレートを関数として、ライブラ 名の 下に表示されます。 インポート 済みのライブラ が表示されていない場合、 XSLT が 変換言語として選択されていることを確認してくたさい。 関数ライブラルの管理も参照してくたさい。

以下の点に注意してくたさい

- MapForce にインポーカ許可されている場合、関数はXSLT ファイル内でXSLT 仕様に準拠している名前が付けられたテンプレートとして宣言されている必要があります。XSLT 2.0 ドキュメント内で発生する関数を<xs1:function name="MyFunction">の書式でインポーすることもできます。インポートされたXSLT ファイル内で、他のXSLT ファイルをインポートまけは含むことができ、これらのXSLT ファイルと関数もインポートされることができます。
- インポートされたカスタム関数のマップすることのできる入力ニネクタはテンプレート呼び出し内で使用される、ラメーターの数により異ないます。任意の、ラメーターもサポートされています。
- 名前空間はサポートされています。

- MapForce に既にんポートされているXSLT ファイルの更新を行う場合、変更は自動的に検知され、MapForce がファイルの再ロードを促します。
- 名前付きのテンプレートに書き込む場合、テンプレート内で使用されている XPath ステート メイカ が正し 名前空間を持つことを 確認してくたさい、マメピングの名前空間に バイドする名前空間を確認する コよ 生成された XSLT コードをプレビューしてください。

XPath 2.0 内のデータ型

XMLドキュメイカらXMLスキーマカ参照されており、スキーマーマオレてドキュメイカ妥当である場合、演算にて目的のデータ型へ暗示的に変換されないデータ型を明示的に構築ませまキャストする必要があります。

Altova XSLT 2.0 エンジンで使用される XPath 2.0 データモデルでは XML ドキュメト から原子化された全てのノード値に xs:untypedAtomic データ型が割り当てられます。xs:untypedAtomic 型は 以下にあるようご暗示的な型変換に使用されます:

例えば

- xdt:untypedAtomic 値か加算演算子におxs:double に暗黙的に変換されるため式 xs:untypedAtomic("1") + 1 はの値を結果とます。
- 四則演算ではオペランドが暗黙的にxs:double へ変換されます。
- 値の比較を行う場合、比較の前にオペライがxs:string へ変換されます。

次も参照してくたさい

<u>例: カスタムXSLT 関数の追加</u> 例: ノードの値の集計 XSLT 1.0 エンシの実装 XSLT 2.0 エンシの実装

7.4.1 例: カスタム XSLT 関数の追加

この別は、カスタムXSLT 1.0 関数をMapForce にんパートする方法を説明しています。以下に示される単純なサンプルロ必要なファイルは <u>くマイドキュメント > #Altova MapForce2021 MapForceExamples ¥</u> ディレクトリコ収められています。

- Name-splitter.xslt.このXSLT ファイルは "string" の ラメーターを伴う "tokenize" と ら名前付きテンプレートを定義しています。テンプレートは入力文字列と作業し、それそれの大文字の文字を空白文字で区切ります。
- Name-splitter.xml (処理されるノースXML インスタンスファイル)
- Customers.xsd (ソースXML スキーマ)
- CompletePO.xsd (ターゲナ XML スキーマ)

カスタムXSLT 関数の追加方法:

1. ライブラリウイボウのベースのライブラリの追加/削除ボタンをクトックします。ライブラル管理ウイボウが開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラ として関数をインポート する場合 グローノ シレライブラリのインポートの横の「追加」 ボタンをクトックします。
- メモ ライブラをローカルでインポートする場合、ライブラリファイルの ひをマピング こ対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラレでは、 インポートされたライブラレヘの いは常に絶対的ない いです。
 - 3. 関数とて振る舞う名前の付けられたテンプレートを含む.xsl おけは.xslt ファイルを参照します。この場合は Namesplitter.xslt です。「開く」をクリックします。新規のライブラルが追加されたことを通知するペッセージボックスか表示され、ライブラノ ウイドウ内に名前の付けられたテンプレートとして定義された関数と共に XSLT ファイル名が表示されます。(この場合は tokenize 関数を持つ Name-splitter)。

Libraries		×
E core		~
🔁 Name-splitt	ter	
tokenize	e result = tokenize([string])	_
記 xlsx 記 xpath2 記 xslt		~
Search for func	tion	•
	Add/Remove Libraries	
🗮 Libraries 🥤	Project	

XSLT 関数をMapForce から使用する方法:

1. tokenize 関数をマピングウィンドウボラッグして、以下に示されるようにアイテムをマップします。



2. XSLT タブをクリックして生成された XSLT コードを確認します。



- メモ マピング内で名前の付けられたテンプレートが使用されると、名前の付けられたテンプレートを含むXSLT ファイルは生成された XSLT コード (xsl:include href...)内に含まれます。 そしてコマド xsl:call-template を使用して呼び出されます。
 - 3. 「出力」ダブをクリックして、マノピングの結果を確認します。

MapForce からカスタム XSLT ライブラリを削除する方法:

- 1. ライブラリウィボウのベースのライブラリの追加/削除をクリックします。ライブラリの管理ウィボウか開かれます。
- 2. 削除するライブラルの横のライブラリの削除 🔀 をクルックします。

7.4.2 例: ノードの値の集計

このセクションではXML インスタンズドキュメント内にある複数のノードを処理し、その結果をターゲホアイテムにある単一のアイテムへマピン グする方法について記述します。この例で使用されるファイルは以下のとおりで、具体的には、マピングの目的は、ソース XML ファイル内の 全ての製品の値段を計算し、出力 XML ファイルに単一の値として書き込むことです。この例で使用されるファイルは以下のとおりで、くマイ ドキュメント > ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥Tutorial ¥フォルダーコ収められています。

- Summing-nodes.mfd マメピングファイル
- input.xml ソースXML ファイル
- input.xsd ソースXML スキーマ
- output.xsd ターゲナ XML スキーマ
- Summing-nodes.xslt カスタムXSLT スタイルシートには個別のノードを集計する名前が付けられたテンプレートかきまれています。

以下の2 つの方法により、集計関数を使用することができます:

- <u>sum</u> 関数を使用します。このMapForce ビルトイン関数はライブラリウィドウ内で使用することができます。
- カスタムXSLT スタイルシートをMapForce にインポートします。

ソリューション 1: "sum" 集計関数の使用

sum 関数をマピング内で使用するコよ 関数をライブラリウイボウからマピング「ボラッグします。ライブラリウイボウ内で使用することの できる関数は、選択されたXSLT 言語・デジョン(XSLT 1 お出はXSLT 2)により異なります。 次に、以下に表示されるようこマピング 接続を作成します。



コアライブラル集計関数に関する情報に関してはcore 集計関数を参照してくたさい。

ソリューション 2: カスタム XSLT スタイルシートの使用

上記の通じのサンプルの目的はノースXML ファイル内の製品のPrice フィールドを集計することです。この場合製品 A とB です。

```
<?rml version="1.0" encoding="UTF-8"?>
<Input xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="input.xsd">
<Products>
<Products>
<Name>ProductA</Name>
<Amount>10</Amount>
<Price>5</Price>
</Product>
```

```
_____
```

698

関数

```
<Product>

<Name>ProductB</Name>

<Amount>5</Amount>

<Price>20</Price>

</Product>

</Products>

</Input>
```

下のコードリスティングは、名前をつたけなテンプレート "Total" と単一パラメーター string を使用するカスタム XSLT スタイルシートを 表示しています。テンプレートは、XML 入力ファイルを介して作動し、XPath 式 / Product / Price によ取得された全ての値を合計 します。

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>

<xsl:template match="*">

<xsl:template match="*">

<xsl:template match="*">

<xsl:call-template name="Total">

<xsl:call-template name="Total">

</xsl:call-template name="string" select="."/>

</xsl:call-template>

</xsl:template name="Total">

<xsl:template name="Total">

<xsl:template name="Total">

<xsl:template name="Total">

<xsl:template name="Total">

<xsl:template name="string"/>

<xsl:template name="string"/>

<xsl:template>

</xsl:template>
```

メモ XSLT 2.0 内のノードを合計するコよスタイルシートの宣言をversion="2.0" に変更してけさい。

XSLT スタイルシートをMapForce にインポートする前にXSLT 1.0を変換言語とて選択してくたさい。カスタム関数をインポートする準備が整いました。以下を行ってくたさい

1. ライブラリウィドウのベースのライブラリの追加/削除 ポタンをクリックします。ライブラルの管理ウィドウか開かれます。

Manage Libraries		×
📮 🛃 New Design1		*
User-Defined Functions		
Own Library Imports Add		
Global Library Imports Add		
		-
	Þ	

- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラノとして関数をインポートする場合 グローノ シレライブラリのインポートの横の「追加」 ボタンをクトックします。

- メモ ライブラ をローカルでインポートする場合、ライブラ ファイルの ゆをマピング に対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラ ノでは、 インポートされたライブラ レヘの いは常に絶対的ない ひです。
 - くマイドキュメント> ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥Tutorial Summing nodes.xslt を参照して「開く」をクリックします。新規のライブラル「追加されたことを通知するメッセージボックスが表示され、新規のライブラル「ライブラリウイドウに表示されます。

Libraries		x
📰 core		\sim
E Summing-nodes		
Total	result = Total(string)	
a∰ xsit		~
Search for function		-
Add/Remo	ove Libraries	
🗮 Libraries		

4. ライブラからTotal 関数をマピングバボラッグし、下に表示されるとおりマピング接続を描きます。



マピングの結果をルビューするコよ「出力」ダブをクリックします。2つのPrice フィールドの合計がTotal フィールドに表示されます。

xml version="1.0" encoding="UTF-8"?
<output <="" td="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></output>
<pre>xsi:noNamespaceSchemaLocation="output.xsd"></pre>
<total>25</total>
<product></product>
<name>ProductA</name>
<amount>10</amount>
<price>5</price>
<product></product>
<name>ProductB</name>
<amount>5</amount>
<price>20</price>

7.5 カスタム XQuery 1.0 関数のインポート

XQuery がマピングの変換言語として選択されている場合、MapForce は、XQuery のために使用することのできるビルトイン関数ラ イブラ をライブラリンイドウ内に表示します。必要であればこのリストをXQuery 関数を使用して、カスタムXQuery 1.0 ライブラノモジュ ーリを MapForce にインポートして拡張することができます

MapForce にんパートするためコよ XQuery ファイルは以下の条件を満たす必要があります:

- XQuery 仕様に従う有効なライブラノモジュールである必要があります。すなわち、module namespace cprefix>= (namespace name]などのモジュール宣言と共に開始する必要があります。
- インポートされたライブラノモジュール内で宣言された全ての関数は、動的なデータ型を返す必要かあます(例えば、xs:string、xs:boolean、xs:integer など)。関数パラメーターも動的な型を有する必要かあます。

XQuery ライブラリモジュールをインポートする

1. ライブラリウイドウのベースのライブラリの追加/削除 ボタンをクリックします。ライブラルの管理ウイドウか開かれます。

Manage Libraries	x	
📮 🛃 New Design1	-	
User-Defined Functions		
Own Library Imports Add		
Global Library Imports Add		
	-	
٠		

- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラノとして関数をインポートする場合 グローノ シレライブラリのインポートの横の「追加」 ボタンをクレックします。
- メモ ライブラをローカルでインポートする場合、ライブラリファイルのパをマッピングに対して相対的に設定することが可能です。グローバルにインポートされたライブラノでは、インポートされたライブラレヘの、なは常に絶対的ないなです。
 - 3. .xq およ.xquery ライブラノファイルを参照し「開く」をクトックします。

インポートされたライブラルモジュールは、ライブラリウィドウに表示され、特定の関数をマッピングエリアイドラッグし、他の関数コンポーネントと同様に使用することができます。
<u>関数をマッピングに追加する</u>も参照してください。

インポート済みのXQuery ライブラノモジュールは表示されていない場合、XQuery が変換言語として選択されていることを確認してくたさい。

次も参照:

<u>XQuery エジの実装</u>

7.5.1 例: カスタム XQuery 関数のインポート

このサンプルは、デモXQuery ライブラノモジュールをMapForce にインポートし、マルングから関数を呼び出す方法について説明しています。このサンプルのデモモジュールは20%としての小数で示される税金を計算する関数により構成されています。生産のシナリオについては、 XQuery モジュールはは複数の関数が含まれている可能性があります。

XQuery モジュール内で宣言されているすべての関数は、アトミックな型を返し、彼らの、ラメーターもアトミックデータ型である必要があります。それ以外の場合、モジュールはMapForce にインポートすることはできません。

デモXQuery モジュールファイルをMapForce がインストールされているコンピューターのドキュメトフォルダー(されして相対的ないなで見つ いることができます: <マイドキュメント>WiltovaWapForce2021WapForceExamplesYutorialWoodule.xq.

```
xquery version "1.0";
module namespace demo="http://www.altova.com/mapforce/demo";
declare function demo:calculatetax($val as xs:decimal) as xs:decimal {
    $val*0.2
};
```

module.xq

MapForce にXQuery モジュールファイルをインポートした後、demo:calculatetax 関数をマソピングから呼び出せるようしています。 XQuery 関数を使用して税金を計算することは、デモの目的です。同じ結果をMapForce ビルドイン関数を使用して達成することもできます。

demo: calculatetax 関数を呼び出すデモマングは、次の、次で見つけることができます… くマイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\CalculateTax_XQuery.mfd。このマンングを最初に開くと、 MapForce はXQuery 内で使用することのできないりつおけ、複数のエンポーネントが含まれていることを通知する警告を表示します。こ の警告は通常で、マンピングがまだインポートされていないカスタムXQuery ライブラから関数をマンビングが参照すると発生します。この警 告を削除し、マンピングを実行するけよ、以下の通り、不足するXQuery モジュールをMapForce にインポートします。

XQuery モジュールを MapForce にインポート する方法:

1. ライブラリウイボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウイボウか開かれます。



- 2. 以下の内のつを行います:
 - •(現在のマピング内のスコープのみで)ローカルライブラルとて関数をインポートする場合現在のマピング名の下の「追加」

おりをクリックします。

- (プログラムレベルで) グローノ ジレライブラリとして関数をインポートする場合 グローノ ジレライブラリのインポートの横の「追加」 ポタンをクトックします。
- メモ ライブラ をローカルでインポートする場合、ライブラ ファイルの ワをマピング に対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラ ノでは、 インポートされたライブラ レヘの 、 なば常に絶対的ない なです。
 - 3. <マイドキュメント> ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥Tutorial ¥module.xq を参照して「開く」をクリックします。 シッセージボックスか表示され、新規のライブラルが追加されていることを通知します。

MapForce X	
A new library has been added. See Messages window for details.	
If you don't see newly added functions in the Libraries window, please make sure that the correct language is selected in the Output menu.	
Don't show this message again.	
ОК	

インポートされたライブラルとdemo: calculatetax 関数はライブラリウイドウで確認することができます。

Librari	es		×
≣	ore		\sim
🖾 m	odule		
	demo:calculatetax	result = demo:calculatetax(val)	
₹÷ ×r	path2		~
Search for function		•	
	Add/R	lemove Libraries	
ELit	braries 📄 Project		

おっ、マピングは検証され、警告なしで実行されます示されている demo: calculatetax 関数はインポートされた XQuery モジュール から来ており、下のイメージで表示されています。おっ、他のビルトィン関数同様マピングに追加することができます。マパングに関数を追加 するを参照して下さい。



CalculateTax_XQuery.mfd

マピングの説明

CalculateTax_XQuery.mfd マピングは、商品を保管するXML ファイルを入力としてとます。各商品には単一の価格が小数として表示されています。例:

```
<Articles>
   <Article>
      <Number>1</Number>
      <Name>T-Shirt</Name>
      <SinglePrice>25</SinglePrice>
  </Article>
   <Article>
      <Number>2</Number>
      <Name>Socks</Name>
      <SinglePrice>2.30</SinglePrice>
   </Article>
   <Article>
      <Number>3</Number>
      <Name>Pants</Name>
      <SinglePrice>34</SinglePrice>
  </Article>
   <<u>Article></u>
      <Number>4</Number>
      <Name>Jacket</Name>
      <SinglePrice>57.50</SinglePrice>
   </Article>
</Articles>
```

Articles.xml

XML ファイルとして同じスキーマーこく XML ファイルをマッピングは作成します。このために、ソースとターゲルのコンポーネントは、マッピング上 て同じ構成を持ちます。マッピングの接続が提案するとおり、ほとんどすべての要素が率直な方法でターゲルカらソースにマップされます。例、 ソース内の各 Article に対して、ターゲル内にArticle が存在します。全てのアイテムは逐語的に、SinglePrice を除いてノース XML にコピーされます。SinglePrice の値はつの関数の助けを借りて計算されます。

- demo: calculatetax XQuery 関数は 基のSinglePrice を入力とてと税金を計算します。
- MapForce ビルトインadd 関数は税金と基のSinglePrice 価格を足して最終の価格が返されます。

重要な点は入力パラメーター型に対応し、XQuery関数の型を返すSinglePrice アイテムのデータ型がis xs:decimal であることです。

下記の出力 ダブをクリックすると、マッピングにより生成された出力が表示されます。ソース XML. に比べ洛価格は20%の価格情報が適用されていることに注意してくたさい。

1	xml version="1.0" encoding="UTF-8"?
2	<pre><Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-</pre></pre>
3	<pre></pre>
4	<number>1</number>
5	<name>T-Shirt</name>
6	<singleprice>30</singleprice>
7	<pre>- </pre>
8	Article>
9	<number>2</number>
10	<name>Socks</name>
11	<singleprice>2.76</singleprice>
12	<pre>- </pre>
13	Article>
14	<number>3</number>
15	<name>Pants</name>
16	<singleprice>40.8</singleprice>
17	<pre>- </pre>
18	Article>
19	<number>4</number>
20	<name>Jacket</name>
21	<pre><singleprice>69</singleprice></pre>
22	<pre>- </pre>
23	L

7.6 カスタム Java と .NET ライブラリのインポート

コンパルされたJava クラスファイルや.NET アセンブファイル、NET 4.0 アセンブを含む)はライブラルペインに追加して、MapForce にあるその他の関数と同じようご使用することができます。これらJava や.NET 関数の出力は、出力ペインで確認することができるしお、関数は生成されたコードでも利用することができます。

灹

- カスタム Java おは.NET 関数をイパーするは、Java クラス(.class) おは.NET.dll アセンブノファイルをエンパル する必要があます。NET アセンブノではない、Java .jar ファイルおは.dll ファイルのイパートはサポートされていません。
- エンパルされた Java ケラス(.class) ファイルは、マピング言語が Java に設定されているサポートされています。 Java Runtime Environment 7 おけお以降が使用するマシノニインストールされている必要があります。 特定の型とシンバーがサポートされています(以下を参照してください、<u>Java 関数サポート</u>)。
- .NET アセンブリファイルは、マンピング言語がC#に設定されているとサポートされています。パラメーターと戻り値の型としてシステムアセンブルら基本データ型 NET アセンブルは、C#以外の.NET 言語で書かれている可能性があります。(例えば、C++.NET おけはVB.NET)。のみを使用することを考慮すると、(以下を参照してくたさい).NET 関数サポート)。
- マピングがインポートされた Java .class おけは.NET DLL アセンブを使用する場合、マピング言語をC++ に設定することは、サポートされていません。
- ネイティブのC++ DLL 関数をからインポートすることは、制限されており、特別なアプローチが必要です。詳細に関しては、以下を参照してくたさい、Java、C#とC++ライブラルに手動で参照する。
- マピングがインポートされた Java .class おけは.NET DLL アセンブを使用する場合、マピング言語をXSLT に設定する ことは、サポートされていません。(アダプターの役割をするカスタム XSLT 関数が書き込まれる必要があります)。
- MapForce マピングから呼び出される全ての関数が"へき等"である必要があります、これは、同じ入力、ラメーターを使用して 関数が呼び出される都度、同じ値が返されることを意味します)。 MapForce により呼び出される正確な順序および回数は定 義されていません。
- Java の場合、ヘポートされたクラスファイルと、シケージは、CLASSPATH 変数に追加される必要はおりません。これは、ビルトイン実行エンジンが、生成された、Java コードと同様、Java エンジンのクラス・スマオーはAnt の自動的にヘパポートされた、パシケージに追加されるからです。しかしながら、ヘパポートされたクラスファイルと、シケージの拡存関係は、自動的には処理されません。ですから、ヘノポートされた、Java クラスファイルましま、シケージが他のクラスファイルはな存する場合、CLASSPATH環境変数の全ての拡存・パッケージの親ディレクトリが追加されていることを確認してくたさい。

Java 関数サポート

トップレベルのクラス静的メンバークラス そして非静的メンバークラスがサポートされます::

- new <classname>(<arg1>, <arg2>, ...)
- <object>.new <member-class>(<arg1>, <arg2>, ...)

メンバー関数と静的関数がサポートされます:

- <function>(<arg1>, <arg2>, ...)
- <object>.<method>(<arg1>, ...)

XML スキーマと Java のデータ型間におおサポートされている変換を以下に示します:

スキーマ型	Java 型
xs:string	String
xs:byte	byte
xs:short	short

スキーマ型	Java 型
xs:int	int
xs:long	long
xs:boolean	boolean
xs:float	float
xs:double	double
xs:decimal	java.math.BigDecimal
xs:integer	java.math.BigInteger

両方向への接続を行うことができます。その他のJava データ型はサポートされていません。配列型はサポートされていません。配列を使った引数や戻り値は無視されます。

Object型はコンストラクター呼び出しませまシンドの戻り植としてサポートされています。これらの型はその他の Java メンドへマノピングすることもできます。MapForce にてオブジェクトを操作することはできません。

.NET 関数サポート

トップレベルのクラス静的メンバークラス そして非静的メンバークラスがサポートされます:

• new <classname>(<arg1>, <arg2>, ...)

メンバー関数と静的関数がサポートされます:

- <function>(<arg1>, <arg2>, ...)
- <object>.<method>(<arg1>, ...)

以下に示されるXML スキーマ型と.NET/C# 型との間の接続がサポーされます:

スキーマ型	.NET 型	C# 型
xs:string	System.String	string
xs:byte	System.SByte	sbyte
xs:short	System.Int16	short
xs:int	System.Int32	int
xs:long	System.Int64	long
xs:unsignedByte	System.Byte	byte
xs:unsignedShort	System.UInt16	ushort
xs:unsignedInt	System.UInt32	uint
xs:unsignedLong	System.UInt64	ulong
xs:boolean	System.Boolean	bool

スキーマ型	.NET 型	C# 型
xs:float	System.Single	float
xs:double	System.Double	double
xs:decimal	System.Decimal	decimal

両方向への接続を行うことができます。その他の.NET/C#型はサポートされていません。配列型はサポートされていません。配列を使って引 数や戻り値は無視されます。

Object型はエンストラクター呼び出しおさはメンドの戻り植としてサポートされています。これらの型はその他の、NET メンドへマンピングすることもできます。MapForce にてオブジェクトを操作することはできません。

データの型の問題と解決方法

カスタムライブラル内の関数が整数型を期待する場合、型 "Number" の定数を関数の引数は接続すると以下に類似した型不一致のエラ ーを引き起こす場合があります:

No match for MyCustomClassLibrary, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null.MyCustomClassLibrary.Converter.AddValues(MyCustomClassLibrary.Convert er, xs:decimal, xs:decimal). Check argument types.

この問題は型 "Number" の定数に固有の問題です。このエラーを引き起こすマピングの例は以下に示されています。このマピングでは、型 "Number" の2つの定数は(型 "Integer" の)関数の引数は接続されています。



この問題にはいてかの解決方法があります。

1. 変数を「数値」から「その他」に変更します、定数コンポーネントのタイトルレーをダブルクトックしてこれを行うことができます。)

😢 定数を編集		×
100		 ● 文字列(S) ○ 数値(N) ○ その他(O)
	ОК	キャンセル

2. 定数の代わりに関数により期待されるデータ型の値を提供する例えば、XML ファイルなどのソースエンポーネトを使用します。



7.6.1 例: カスタム Java クラスのインポート

この例は、カスタム Java クラスファイルを MapForce にんパート する方法を説明しています。

メモ このサンプは完了するコは Java SE 7 Runtime Environment おうお以降が必須です。

変換言語を Java に設定します(次を参照してください)

1. ライブラノウイドウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウイドウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマピング名の下の「追加」 ボタンをクリックします。
 - (プログラムレベルで) グローノ ジレライブラノとして関数をインポートする場合 グローノ シルライブラリのインポートの横の「追加」 ボタンをクトックします。
- メモ ライブラをローカルでインポートする場合、ライブラファイルの ひをマピング こ対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラノでは、 インポートされたライブラノトの 、 、 以ば常に絶対的な」 なです。
 - 3. 以下のファイルを参照します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Java\Format\Format.class。 メッセージが表示され、新規のラ イブラル追加されていることを通知します。 インポートされたライブラルコケイブラレケイドウで確認することができます。

Libraries	x
🔁 core	\sim
🔁 db	
📴 edifact	
🔁 Format.Format	
Format.Format	
Format.Format.FormatNumber	
語 lang 語 xbri 語 xlsx	~
Search for function	•
Add/Remove Libraries	
🗮 Libraries	

インポートされた新規のライブラルライブラリウイドウニ表示されていない場合、変換言語がJava に設定されていることを確認してくたさい。関数をマッピングに追加するココライブラリウイドウからマッピングにす。関数をマッピングに追加するも参照してくたさい。

MapForce内でマパングの出力をプレビューする方法:

- 次のマ・ピングを開きます: <マイドキュメント
 >\Altova\MapForce2021\MapForceExamples\Java\FormatNumber.mfd。これは上記のJava .class ライ ブラ を へ ポート する完全なマ・ピングです。
- 2. 出力ボタンをクリックして、マッピングの結果を確認します。

1	Start date,End date,Region,Amount	
2	2008-01-01,2008-01-31,CA,"110.400,00"	
з	2008-01-01,2008-01-31,MA,"75.300,00"	
4	2008-02-01,2008-02-29,CA,"114.300,00"	
5	2008-02-01,2008-02-29,MA,"65.200,00"	
6	2008-03-01,2008-03-31,CA,"134.200,00"	
7	2008-03-01,2008-03-31,MA,"86.100,00"	
8	2008-04-01,2008-04-30,CA,"107.300,00"	
9	2008-04-01,2008-04-30,MA,"112.100,00"	
10	2008-05-01,2008-05-31,CA,"114.400,00"	
11	2008-05-01,2008-05-31,MA,"93.800,00"	

Java 内でマッピングを実行する

- 1. 「ファイル」メニューカジョードの生成 | Java」をクトックします。
- 2. コードが生成されるターゲナディレクトリを選択し、「OK」をクトックします。
- 3. 生成されたライブラを Java プロジェクトにインポートし Java アプリケーションをビルドします 例: Java コードの生成と実行を参照してくたさい。

7.6.2 例: カスタム .NET DLL アセンブリのインポート

この列は、C# で作成された.NET DLL アセンブを MapForce にインポートする方法を説明しています。このサンプルのノースコードは 以下の 次で見つけることができます: <マイドキュメント> Altova MapForce2021 MapForceExamples W#Format。 MapForceにインポートされる.dll アセンブルは... \bin\Debug ディレクトリ内にあります。 Visual Studio 内でノリューション.sln ファイル を開き最新の.NET Framework バージョンを使用して新規の.dll ファイルをコンパイルすることができます。

メモ このサンプルを実行するコは、NET Framework 2.0 おけお以降がエレビューターにインストールされている必要があります。有効 化されていない場合 コントロールレドル>プログラムと機能 > Windows 機能の有効化または無効化 > .NET Framework <バージョン>から有効化することができます。

.NET アセンブリファイルをインポート する方法:

1. ライブラリウィボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウィボウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラリとして関数をインポートする場合 グローノ シルライブラリのインポートの横の「追加」 ボタンをクリックします。
- メモ ライブラ をローカルでインポートする場合、ライブラ レアイルの ひをマンピング に対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラ ノでは、 インポートされたライブラノへの ひは常に絶対的ない ひです。
 - 3. Format.dll ファイルを…\MapForceExamples\C#\Format\bin\Debug\ ディレオ・」から参照します。メッセージが表示され、新規のライブラル追加されていることを通知します。 インポートされたライブラルはライブラリウィドウで確認することができます。

Libraries	x		
🔁 core	^		
置 db			
2 ⊕ edifact			
E Format, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null			
Format.FormatNumber			
new Format			
E lang E xbrl E xbrl			
	\vee		
Search for function	-		
Add/Remove Libraries			
🗮 Libraries			

インポートされた新規のライブヲルライブヲリウインドウニ表示されていない場合、変換言語がC#に設定されていることを確認してくたさい。 関数をマッピングに追加するココライブヲリウインドウからマッピングゴンドラッグします。
<u>関数をマッピングご追加する</u>も参照してくたさい。

マッピング出力をプレビューする方法:

- 1. ... \MapForceExamples \C# フォルダー内にある Format Number.mfd ファイルを開きます。これは既に上記の.dll ライ ブラ を へポートして るサンプルマナングです。
- 2. 出力ボタンをクリックして、マンピングの結果を確認します。

1	Start date,End date,Region,Amount
2	2008-01-01,2008-01-31,CA,"110.400,00"
з	2008-01-01,2008-01-31,MA,"75.300,00"
4	2008-02-01,2008-02-29,CA,"114.300,00"
5	2008-02-01,2008-02-29,MA,"65.200,00"
6	2008-03-01,2008-03-31,CA,"134.200,00"
7	2008-03-01,2008-03-31,MA,"86.100,00"
8	2008-04-01,2008-04-30,CA,"107.300,00"
9	2008-04-01,2008-04-30,MA,"112.100,00"
10	2008-05-01,2008-05-31,CA,"114.400,00"
11	2008-05-01,2008-05-31,MA,"93.800,00"

カスタム C# アプリケーションからマッピングを実行する方法:

- 1. 「ファイル」メニューカジョードの生成 | C#」をクルクします.
- 2. コードが生成されるターゲナディレクトリを選択し、「OK」をクリックします。
- 3. Visual Studio を使用してアプケーションをビルドし、コンノールアプケーションを実行します(<u>C#コードの生成</u>を参照してくだ さ)。

7.7 Java、C# および C++ ライブラリを手動で参照する

MapForceに直接カスタムライブラをインポートする代替のオプションとしては、MapForceにより識別されているカスタム.mff ファイル (MapForce 関数ファイルを使用してレファレンスを作成することができます。mff ライブラリファイルは、カスタムコード内のクラス定義と MapForceのレンを手動で定義する XML ファイルにと必必須です。カスタム.mff ファイルを作成すると、NET DLL おけよ Java クラス ファイルをインポートすることと同様 MapForceにインポートすることができます。

灹

- MapForce にんポートするために関数を修飾するこよ、戻り値と、ラメーターか単純型である必要があります。各言語のために使用することのできるデータ型のリストに関しては、データ型マレビングを参照してくたさい。
- 関数ライブラをカスタム.mff ファイルからインポートすると、(「出力」、ボタンをクリックして) MapForce 内で直接マンピングをプレビューすることは制限されます。C++ で書かれているライブラルに関しては、MapForce 内のマンピングのプレビューは、サポートされていません。Java とC# の場合は、ライブラルドイティブ言語の型を使用する場合は、プレビューを使用することができますが、ライブラルがAltova 生成されたクラスをインポートする場合は、使用することができません。しかしなから、ライブラルによりターゲートされている特定の言語でコードを生成することができます。カスタム関数は、生成されたコード内で使用することができ、生成されたコードからマンピングを実行することが可能にないます。
- 生成されたマメピングコードにより呼び出される関数の順序は定義されていません。MapForceではキャッシュされた計算結果を 再利用するか、任意の順序で条件式を再評価する必要があります。ユーザー定義関数とカスタム関数ライブラを区別する必要 があります。
- ユーザー定義関数は、マノピング内で視覚的に作成され、マノピング、mfd ファイルと共に保存されるため*、mff ファイルはは保存 することができません。ユーザー定義関数は、マノピング内で視覚的に作成され、マノピング、mfd ファイルと共に保存されるため *、mff ファイルはは保存することができません。詳細に関しては、以下を参照してくたさい、詳細に関してはユーザー定義関数の呼び出しとインポートを参照してくたさい。
- 2010 より以前のMapForce バージョンをアップグレードする場合、カスタム関数内で使用されているデータ型を更新する必要がある可能性があります。データ型マッピングを参照してくたさい。

カスタム.mff ファイルの作成と構成の順字に関する詳細は .mff ファイルの構成方法を参照してくたさい。例に関しては以下を参照してくたさい。

- <u>例: C# を.mff に適用する方法</u>
- <u>例: C++ を.mff に適用する方法</u>
- <u>例: Java を.mff に適用する方法</u>

7.7.1 .mff ファイルの構成

MapForce 関数ファイル(.mff)は、MapForce にカスタムJava、C#、おゴまC++ ライブラルら関数の適用を可能にし、ライブラノ ウイドウ内で表示されるXML フォーマナの構成ファイルです。.mff ファイルは、基本的にカスタムライブラルとMapForceの媒介であり、 (a)カスタム関数にインターフェイスし、おこ、(b)生成されたコード内に実装が存在する箇所を指定するために構成されていなければなりま せん。このトピックは上記を行う方法について説明しています。

以下の点に注意してくたさい

- *mff *mff ライブラリファイルは以下のスキーマファイルに対して妥当である必要が決ます: ... (Program Files MapForceLibraries \mff.xsd。 mff.xsd スキーマは内部的な使用のためだけのたのです。 Altova GmbH は、新 規リレースこて、このファイルフォーマトを変更する権利を有しています。
- 1つのC#、C++、およよJava クラスを.mff ファイルごとこ定義することが可能です。

以下のコードリストはC++のためのサンプル.mffを説明して、ます。

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cpp">
         <setting name="namespace" value="mylib"/>
         <setting name="class" value="Greetings"/>
         <setting name="path" value="C:\Libraries\cpp"/>
         <setting name="include" value="Greetings.h"/>
         <setting name="source" value="Greetings.cpp"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cpp">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

mff ファイルがMapForce で表示される例が以下のイメージで示されています。新たなライブラリ"mylib" が"sayhello" 文字列関数を 含む、アルファベト順に並べ替えられた)ライブラルエトリーに表示されます。

Libraries		×		
🔁 edifact		^		
赶 lang				
🔁 mylib				
sayhello	result = sayhello(ismorning)			
🖭 xbrl		~		
Search for function		•		
Add/Remove Libraries				
🔚 Libraries 🛅 Pro	ject			

ニーズに合うかたちで mff ファイルを適応させるステップを以下に示します:

ライブラノ名の構成方法

以下に示されるmff ファイルのセグメトにてライブラルなが宣言されます。慣例として、MapForce内ではライブラルには全て小文字で記述されますが、大文字も使用することができます。

```
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
```

上のサンプルでは、ライブラリウインドウに表示されるエイトリはmylib と呼ばれます。

言語の実装の構成

<implementations>要素はライブラルがポートする言語を指定する必須要素であり、<mapping>の子として追加される必要があ ります。例:

```
<!-- ... -->

<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="mff.xsd">

<implementations>

<implementation language="cpp">

<setting name="namespace" value="mylib"/>

<setting name="class" value="Greetings"/>

<setting name="path" value="C:\Libraries\cpp"/>

<setting name="include" value="Greetings.h"/>

<setting name="source" value="Greetings.cpp"/>

</implementation>

</implementation>
```

各 <implementation> の設定により、生成されたコードから Java、C++、C# にて定義された関数を呼び出すことができるようしより ます。

.mff ファイルは、1つ以上の言語をターゲナオることができるように作成することができます。この場合、追加言語ごとに追加 <implementation> 要素が含まれる必要があります。各プログラング言語の設定を以下に示します。

Java ライブラリレファレンス

生成されたファイルがGreetings.class ファイルを検索できることは重要です。このためクラスへの参照は Java クラス・マリン追加されていることを確認してくたさい。

C# ライブラリレファレンス

<!--->

```
<implementation language="cs">
        <setting name="namespace" value="MyLibrary" />
        <setting name="class" value="Greetings" />
        <setting name="reference" value="C:
        Libraries\cs\MyLibrary\bin\debug\MyLibrary.dll" />
        </implementation>
<!-- ... -->
```

C# は関してはコード内の名前空間が、mff ファイル内で定義されている名前空間に一致することが重要です。(上のコードリストでの名前空間はMyLibrary です)。クラス名は関しても同様です(上のコードリストでの名前はGreetings です)。第3の設定である、 reference は、生成されたコードベルンクする dll の 文を提供します。

C++ ライブラリレファレンス

C++のために以下に注意してくたさい

- namespace は、Greetings ケラスが定義される名前空間です。mapping 要素内のlibrary 属性に等しくある必要が あます。
- path は ソールファイルを見つけることができる みです。
- マピングのオングのコードが生成されると、ソースファイルは「ファイル | コードを生成 | C++」メニューコマドおよびデルノカリを 選択する際に定義される)デルノカリtargetdir/libraryname にコピーされ、プロジェクトファイルに含まれます

提供するファイルはすべて生成されるアルゴリズムに含まれます。

ンポーネトの追加

MapForce のライブラリウィド・ウのグラフィカルユー・ザーインターフェイスでは、各関数は、関数グループの下にネストされ表示されます。例え ば「文字列関数」など。.mff ファイル内では、関数は<component> 要素に対応します。一方、各 <component> は <group> 要素の下でネストされる必要があます。例

```
<!-- ...->
<group name="string functions">
<component name="sayhello">
<!-- ... -->
</component>
</group>
<!-- ... -->
```

下に表示されるコードは、sayhello と呼ばれるサンプル関数(コンポーネント)を定義しています。

```
<!-- ...->
<component name="sayhello">
<sources>
```

```
716 関数
```

MapForce 内でコンポーネントは以下のように表示されます:



上のコードリスティングでは、<datapoint> は、大まかに、入力おけ出力コネクタとして知られる関数の入力おけ出力パラメーターとして定義することができます。<datapoint> のtype 引数は、パラメーターのデータ型(おけよ、戻り値のデータ型)を指定します。

各関数のためこターゲナデータポイトのみか許可されています。定義することのできるソースデータポイトの数に制限はありません。

各データポイトのデータ型は XML スキーマ型(例えば xs:string、xs:integer など)の1つである必要があります。これらのデ ータ型は、Java、C++ おけまC# ライブラリ内出定義された関数の・デメーターの型のデータ型に対応します。XML スキーマデータ型を 言語型にマッピングするけま、データ型のマッピングを参照してくたさい。

ライブラリケイドウ内では関数は短い、および、長い説明と共に表示されています。短い説明は常に、関数名の右側に表示されますが、長い説明は、マケズを短い説明の上にポイトするとこともして表示されます。

Libraries	×
편 core	~
₫ db	result = sayhello(ismorning)
🔠 edifact	
ang lang	Returns "Good morning" or "Good day", depending on the input parameter.
nylib	
sayhello result = sayhello(isn	norhing)
J	~
Search for function	▼
Add/Remove Libraries	
🗮 Libraries 🛅 Project	

言語の実装の定義

ライブラリケイドウ内でカスタム Java、C# おうまC++ クラスの関数間の接続を作成します。これは、<implementation> 要素を 使用することに以達成することができます。

上記の通り、1つの関数が、サポートされるプログラミング言語のナーガコンロゲン、複数の実装要素を持つ場合かあります。関数は、Java内では、「Hello"と呼ばれ、C++内では、「SayHello"と呼ばれる場合かあります。このナーガ、各プログラング言語のナーガコ個別の関数名を指定する必要があります。3つのプログラング言語のナーガの関数は以下のようコンジます。

```
<!--->
<component name="sayhello">
<!-->
  <implementations>
     <implementation language="cs">
        <function name="HelloFunction"/>
     </implementation>
     <implementation language="java">
        <function name="Hello"/>
     </implementation>
     <implementation language="cpp">
        <function name="SayHello"/>
     </implementation>
  </implementations>
<!--->
</component>
<!--->
```

関数名として提供される値は、Java、C# おけはC++ クラス内のメンドの名前に正確に一致する必要がおます。

7.7.2 .mff ファイルを MapForce にインポートする

カスタム.mff ファイルの作成後 MapForce に以下のようにインポートすることができます:

1. ライブラリウイボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウイボウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベリルご) グローノ ジレライブラリとして関数をインポートする場合 グローノ シリライブラリのインポートの横の「追加」

718 関数

おりをクリックします。

- メモ ライブラ をローカルでインポートする場合、ライブラ レアイルの ひをマピング に対して相対的に設定することが可能です。 グローバ ルニインポートされたライブラレでは、 インポートされたライブラレヘの いは常に絶対的ない いです。
 - 3. カスタム.mff ファイルを参照し「開く」をクトックします。

マピング言語をカスタムライブラルことりターゲットにされた言語に設定すると、インポートされたライブラルライブラリウインドウ内で表示されます。

.mff ファイルを プログラムファイル(おけまプログラムファイル(x86) フォルダー)に相対する... \Altova\MapForce2021\MapForceLibraries フォルダーに保存すると、MapForceの起動時にライブラルは自動的にライブラル ウィドウコロードされます。対応するライブラリファイル(.mff)を削除ませば追加することにより、ライブラルその関数の表示の切り替えを行 うことができます。

7.7.3 データ型のマッピング

次のテーブルはMapForce にJava、C#、および C++ ライブラ を適応する、カスタム.mff ファイルを作成する際に関数戻り値の型と ハウメーターの型としてサポートされるデータ型をリストしています。テーブルは、ネイティブとネイティブではカデータ型をリストしています。Altova date などのネイティブではカデータ型をサポートする場合、カスタム Java とC# ライブラは Altova ライブラルのレファレンスを含む必要があ ります。C++ の場合、Altova ライブラルは常にインポートされる必要があります。Altova ライブラルの生成方法に関しての詳細は、次を参照してくたさい、ユード、ジェネレーター。

XML スキーマ型	Java 型	C# 型	C++ 型
anyAtomicType	String	string	string_type
anySimpleType	String	string	string_type
anyURI	String	string	string_type
base64Binary	byte[]	byte[]	altova::mapforce::blob
boolean	boolean	bool	bool
byte	int	int	int
date	<u>com.altova.types.DateTi</u> <u>me</u>	Altova.Types.DateTime	altova::DateTime
dateTime	<u>com.altova.types.DateTi</u> <u>me</u>	Altova.Types.DateTime	altova::DateTime
dayTimeDuration	<u>com.altova.types.Durati</u> <u>on</u>	Altova.Types.Duration	altova::Duration
decimal	java.math.BigDecimal	decimal	double
double	double	double	double
duration	<u>com.altova.types.Durati</u> <u>on</u>	Altova.Types.Duration	altova::Duration
ENTITIES	String	string	string_type

XML スキーマ型	Java 型	C# 型	C++ 型
ENTITY	String	string	string_type
float	double	double	double
gDay	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	altova::DateTime
gMonth	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	<u>altova::DateTime</u>
gMonthDay	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	altova::DateTime
gYear	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	<u>altova::DateTime</u>
gYearMonth	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	<u>altova::DateTime</u>
hexBinary	byte[]	byte[]	altova::mapforce::blob
ID	String	string	string_type
IDREF	String	string	string_type
IDREFS	String	string	string_type
int	int	int	int
integer	java.math.BigInteger	decimal	int64
language	String	string	string_type
long	long	long	int64
Name	String	string	string_type
NCName	String	string	string_type
negativeInteger	java.math.BigInteger	decimal	int64
NMTOKEN	String	string	string_type
NMTOKENS	String	string	string_type
nonNegativeInteger	java.math.BigInteger	decimal	unsignedint64
nonPositiveInteger	java.math.BigInteger	decimal	int64
normalizedString	String	string	string_type
NOTATION	String	string	string_type
positiveInteger	java.math.BigInteger	decimal	unsignedint64

XML スキーマ型	Java 型	C# 型	C++ 型
QName	javax.xml.namespace.QNa me	Altova.Types.QName	altova::QName
short	int	int	int
string	String	string	string_type
time	<u>com.altova.types.DateTi</u> <u>me</u>	<u>Altova.Types.DateTime</u>	<u>altova::DateTime</u>
token	String	string	string_type
unsignedByte	long	ulong	unsignedint64
unsignedInt	long	ulong	unsignedint64
unsignedLong	java.math.BigInteger	ulong	unsignedint64
unsignedShort	long	ulong	unsignedint64
untypedAtomic	String	string	string_type
yearMonthDuration	<u>com.altova.types.Durati</u> <u>on</u>	Altova.Types.Duration	altova::Duration

7.7.4 例: C# を.mff に適用する方法

このサンプルはサンプルC# ライブラを作成し MapForce ライブラルしてインポート可能な MapForce 関数ファイル(.mff) に適用する 方法について説明しています。

メモ C# ライブラルの.mff 書式への適用はC# ライブラを MapForce にインポートする多くの方法の一つです。簡単な代替法は .NET アセンブを直接インポートすることです。例 カスタム.NET DLL アセンブルクインポートを参照してくたさい。

構成プロセスは以下のステップにとり構成されています

1. Visual Studio 内に新規クラスライブラを作成します。関数は public static とて定義されていることご注意してくたさい。

```
namespace MyLibrary
{
    public class Greetings
    {
        public static string SayHello(bool isMorning)
        {
            if (isMorning)
                return "Good morning!";
                return "Good Day!";
        }
    }
}
```

2. 特別な XML スキーマ型(日付および期間)が必要な場合は Visual Studio プロジェクトから Altova.dll ライブラルコンファレンスを追加します。このライブラ を取得する コよ カスタム関数を使用しない マンピングから C#コードを生成します。 Altova.dll
ファイルはコードが生成されたディレクトリに相対する...\Altova\bin\debug ディレクトリ内にあります。Visual Studio 内の Altova.dll (ユファレンスを追加するは、「プロジェクト」メニューから「レファレンスを追加」をクルクして、Altova.dll ファイルを参照します。おニュードに以下のラインを追加してください、using Altova.Types; 。XML スキーマ型を C#型 にマップする方法に関する情報はデータ型のマノビングを参照してください。

- 3. Visual Studio プロジェクトを構築します。MyLibrary.dll がプロジェクト出力ディレクトリニ生成されます。
- 4. XML エディターを使用して、新規の.mff ファイルを作成し、... Program Files MapForceLibraries Mff.xsd スキー マーン対して検証します。implementation language="cs] 下のすべての参照が正確な C# メンバーと前に作成され ている やを指しているこを確認してくたさい。おこfunction name="SayHello]は C# 内で定義されている通りの関数 名を参照している必要があります。...mff ファイルの構成 を参照してくたさい。

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cs">
         <setting name="namespace" value="MyLibrary" />
         <setting name="class" value="Greetings" />
         <setting name="reference" value="C:
\Libraries\cs\MyLibrary\bin\debug\MyLibrary.dll" />
      </implementation>
   </implementations>
   <group name="string functions">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cs">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

MapForce に適応するカスタムライブラル.mff ファイルの作成が完了しまた。カスタム.mff ファイルを MapForce で使用することができるようてなりまた。.mff ファイルを MapForce にインポートするを参照してくけさい。

7.7.5 例: C++ を.mff に適用する方法

このサンプルはサンプルC++ ライブラを作成し MapForce ライブラルとてインポート可能な MapForce 関数ファイル(.mff) に適用する方法について説明しています。

1. ヘッダー(.h) ファイルをクラスライブラルのナックに作成します。以下のコードリストは Greetings.h と ら名前のサンプルヘッダーファ イルを表しています。

```
#ifndef MYLIBRARY_GREETINGS_H_INCLUDED
#define MYLIBRARY_GREETINGS_H_INCLUDED
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
using namespace altova;
namespace mylib {
    class ALTOVA_DECLSPECIFIER Greetings
    {
        public:
            static string_type SayHello(bool isMorning);
    };
    } // namespace mylib
#endif // MYLIBRARY_GREETINGS_H_INCLUDED
```

関数は静的として宣言されており、名前空間 altova がインポートされています。ALTOVA_DECLSPECIFIERをクラス名の前に書き込んでくたさい。これにより、動的または静的なリンケージの使用に関わらず、次に生成されるコード内でクラスが正確にコンパルにわます。

2. .cpp ヘッダーファイルと同じ名前のファイルを作成します。.cpp ファイルは、.h ファイルと同じデルクトリン存在する必要があます。次のコード リスティングは、前に作成された Greetings.h ファイルを含む、Greetings.cpp と呼ばれるサンプル.cpp ファ イルを示しています:

```
#include "StdAfx.h"
#include "../Altova/Altova.h"
#include "../Altova/AltovaException.h"
#include "Greetings.h"
mamespace mylib {
    string_type Greetings::SayHello(bool isMorning)
    {
        if( isMorning )
            return _T("Good morning!");
        return _T("Good day!");
    }
}
```

StdAfx.h と一部のAltova ライブラ をインポートするライノニ注意してくたさい。これらのライノは変更されてはなりません。ライブ ラトの やは正確です。生成されたコード内では、これらの やはそれぞれのファイルを指します。

Java おけての、シースC++ ファイルをエンパーオる必要はありません。ファイルは生成されたコードにエピーされ、生成されたマレングコードの残りと共にエンパーにわます。

3. XML エディターを使用して、新規、mff ファイルを作成し、... (Program Files) MapForceLibraries) mff.xsd スキーマ に対して検証します。...mff ファイルの構成方法で説明されている通り名前空間、関数名、データ型が内のC++ コードのたのに 対応していることを確認してくたさい。データ型へのサポートに関する詳細はデータ型のマッピングを参照してくたさい。

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="mylib" xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="cpp">
         <setting name="namespace" value="mylib"/>
         <setting name="class" value="Greetings"/>
         <setting name="path" value="C:\Libraries\cpp"/>
         <setting name="include" value="Greetings.h"/>
         <setting name="source" value="Greetings.cpp"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="cpp">
               <function name="SayHello"/>
            </implementation>
         </implementations>
         <description>
            <short>result = sayhello(ismorning)</short>
            <long>Returns "Good morning" or "Good day", depending on the input
parameter.</long>
         </description>
      </component>
   </group>
</mapping>
```

MapForce に適応するカスタムライブラレ.mff ファイルの作成が完了しました。カスタム.mff ファイルをライブラルとて MapForce にインポートする準備が整いました。.<u>mff ライブラルクレポート</u>を参照してくたさい。<u>C++ コードの生成</u>で説明されているとおり、ネイティブの C++ ライブラ を使用するマンピングを実行するコよ、C++ コードを生成し、C++ コード まけ コアプルケーションから実行する必要かあります。

C++ ゴンパイルエラーが発生した場合

以下のラインでエンパラーエラーか発生した場合:

#import "msado15.dll" rename("EOF", "EndOfFile")

msado15.dll ファイルへの参照を含むプロジェクトプロ、ティを変更します(通常はC:\Program Files\Common Files\System\ADO)。

7.7.6 例: Java を .mff に適用する方法

このサンプルはサンプルJava ライブラを作成し MapForce ライブラルしてインポート可能な MapForce 関数ファイル(.mff) に適用する方法について説明しています。

724 関数

メモ Java ライブラルの.mff 書式への適用はJava ライブラをMapForce にんパトする多くの方法の一つです。簡単な代替 法はJava .class ファイルを直接インポトすることです。例 カスタム Java クラスのインポートを参照してくたさい。

構成プロセスは以下のステップにと、構成されています

- 1. (例えば Eclipse などの優先する開発環境で新規のJava プロジェクトを作成します。
- 2. Greetings と呼ばれるクラスから構成される com.mylib と呼ばれる新規の シケージをプロジェクト に追加します。下のコード リスティングでは SayHello 関数は public static とて定義されていること 注意してください。

```
package com.mylib;
public class Greetings {
    public static String SayHello ( boolean isMorning ) {
        if( isMorning )
            return "Good Morning!";
        return "Good Day!";
    }
}
```

3. 任意で、日付、時間、または期間などの特別なスキーマ型をサポートする必要がある場合、com.altova.types パッケージをインポートしてくたさい、このパッケージを取得するはよ、Java コードをカスタム関数を持たないマッピングから生成してくたさい。

import com.altova.types.*;

- 4. ケラスファイルヘカスタムライブラ をコンパ イルし、 Java ケラス ない追加します。
- 5. XML エディターを使用して、新規.mff ファイルを作成し、...\Program Files\MapForceLibraries\mff.xsd スキーマ に対して検証します。implementation language="java" 下のすべての参照が正確な Java メンバーと前に作成さ れている 次を指していることを確認してくたさい。またfunction name= ~ SayHello」は Java 内で定義されている通りの 関数名を参照している必要があります。...mff ファイルの構成 を参照してくたさい。

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping version="9" library="custom"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="mff.xsd">
   <implementations>
      <implementation language="java">
         <setting name="package" value="com.mylib"/>
         <setting name="class" value="Greetings"/>
      </implementation>
   </implementations>
   <group name="greetings">
      <component name="sayhello">
         <sources>
            <datapoint name="ismorning" type="xs:boolean"/>
         </sources>
         <targets>
            <datapoint name="result" type="xs:string"/>
         </targets>
         <implementations>
            <implementation language="java">
               <function name="SayHello"/>
            </implementation>
```

```
</implementations>

<description>

<short>result = sayhello(ismorning)</short>

<long>Returns "Good morning" or "Good day", depending on the input

parameter.</long>

</description>

</description>

</group>

</mapping>
```

MapForce に適応する Java ライブラル.mff ファイルの作成が完了しまた。カスタム.mff ファイルをライブラルとて MapForce にインポートする準備が整いました。.mff ライブラルのヘルポートを参照してくたさい。

7.8 関数ライブラリの管理

MapForce では以下のライブラルの種類をインポートしマッピング内で使用することができます:

- ユーザー定義関数(UDF)を含むマングデザインファイル(*.mfd)。これはビルドイン関数を使用してポーネントをビルディング ブロックとして使用する MapForceを使用して作成されたユーザー定義関数を含むマンピングファイルを具体的に指しています。
 詳細に関しては、ユーザー定義関数の作成を参照してください。

- <u>カスタム Java と.NET ライブラルクンポト</u>内で説明されている通りにインポートするための Java .class ファイルと.NET .dll ライブラル
- メモ (構成を必要とすることなく) 関数のカスタム外部ライブラ を直接、おけま MapForce によ 認識されている MFF (MapForce 関数ファイル)を構成することによりインポートすることができます。後者のアプローチを使用する場合 Java グラスと.NET アセンブリーブ加え C++ ライブラ をインポートすることができます。.mff ファイルによりインポートされたライブラルは Java、C# とC++ ライブ ラ<u>を手動で参照する方法</u>で説明されている必要条件を満たす必要があります。

ライブラの管理ウィンドウ

マンピングファイルにお使用されたすべてのライブラルライブラルの管理ウィドウからビューして管理することができます。これにはUDFとカスタ ムライブラル含まれています。

デフォルトではライブラノの管理ウィンドウは表示されていません。表示するコは以下の一つを行うことができます

- 「表示」メニューから「ライブラリの管理」をクリックします。
- ライブラリウィボウのベースのライブラリの追加/削除をクリックします。

ライブラリの管理	×
🔁 😓 FormatNumber.mfd	-
日 自分のライブラリのインポート 追加	
🕞 🔂 Format.dll 🗙 オプション	
f _x Format, Version=0.0.0.0, Culture=neutral, PublicKey	i
f _x Format, Version=0.0.0.0, Culture=neutral, PublicKey	i
自分のライブラリのインポート	
😢 グローバルライブラリのインポート 追加	

UDFとライブラを現在アクティブなマッピングのナッションのみ、ます」はすべての開かれているマッピングのナッシュ表示することを選択できます。現在開かれているすべてのマッピングのナッションパートされた関数とライブラを表示するコよ、ウインドウ内を右クトックし、コンテキストメニューから「開かれているドキュメントの表示」選択します。

名前の代わりに開かれているマンピングのウを表示すること、コンテキストメニューから「ファイル・マの表示」を選択します。

ライブラノの管理ウイドウ内に表示されているデータはソノー階層に以下のように整理されています

- ・ 現在開かれているマピングドキュメトはホップレベルエトリとして表示されています。各エトリにはつのブランチが存在します: ユーザー定義関数と自身のライブラリインポート。
 - **ユーザー定義関数 ブラン**升までがキュメト内に含まれているユーザー定義関数を表示しています。
 - ・ 自身のライブラリのインポート ブランチは現在のマメピングドキュメトに ローカルでインポートされたライブラ を表示して、ます。「ライブラリ」とう用語は他のマメピングドキュメト(ユーザー定義関数を含む.mfd) おけはXSLT 1,0 XSLT 2.0、XQuery 1.0*、Java*、C#* で作成されたカスタム外部ライブラノ、封けば前述の.mff ファイルを指します。マメピングは他のマメピングドキュメトをライブラノとしてインポートする可能性があるため自身のライブラリのインポート 構造は複数のレベルの深さったいます。
- グロー・バルライブラリのインポートエトリはアプリケーションノブルでグローノ バルニインポートしたカスタムライブラ を含みます。.mfd ファイルの場合、構造は上記の理由から複数のレベルの深さったります。
- *MapForce Professional またはEnterprise エディション内でのみこれらの言語はサポートされています
- メモ XSLT、XQuery、C#、およびJava ライブラルは自身の拡存関係を持つ場合が設ます。このような依存関係はライブラルウインドウニ表示されません。

コンテキストメニューコマンド

オブジェクトを右クトックし、次のコンテキストメニューオプションを選択することにより、ライブラノの管理ウィンドウ内のオブジェクトに対して多種の 操作を行うことができます

بر≥⊏	説明	適用可能範囲
開く	マピングを開きます。	マピング
追加	関数のカスタムライブラ を参照するダイアログボックスを開き ます。	自身のライブラルのインポート
ライブラリウィンドウ内で関数をロケートする	ライブラリウィボウのフォーカスを変更し、関数を選択しま す。	関数
切り取り、コピー、削除	これらの標準 Windows コマドは MapForce ユーザー 定義関数に対してのみ適用が可能です。外部 XSLT ファ イルオゴゴ地のライブラノの種類から関数をコピーして貼り付 いることはできません。	ユ ーザ- 定義関数
貼り付け	以前にクルプボードにコピーされたユーザー定義関数を現在 のライブラルコ貼り付けることができます。	ライブラノ(UDF)
オプション	現在のライブラルのためこオプションを設定おさま変更すること ができるダイアログボックスを開きます。	ライブラノ
開かれているすべてのドキュメントを表示する	このオプタンがオノゴリト替えられていると、ライブラルの管理 ウィドウは現在開かれているすべてのマッピングを開きます。 マッピング間で関数をコピーして貼い付ける場合この機能は 通常役に立ちます。それ以外の場合、現在焦点の当たって いるマッピングのみが表示されます。	 常に
Windows ファイルの表示	このオプションがオノニジル潜えられていると、ライブラルの管理 ウィンドウ内のオブジェクトがフルレ マと共に表示されていま す。それ以外の場合、オブジェクト名のみが表示されていま す。	常に

728 関数

7.8.1 ローカルとグローバルライブラリ

ライブラ を ローカル おさま グローノ ジルこインポート することかできます。 グローノ ジレなインポート はアプリケーションノベルです。 ライブラ がヴロ ーノ ジルこインポート されると、 全てのマッピングからの関数を使用することかできます。

ローカルなイポトはマピングファイルレベルです。例えば、マピング A.mfd で作業中にマピング B.mfd からすべてのユーザー定義関数をインポトすると仮定します。この場合マピング B.mfd は A.mfd 「ユーカルなライブラルとてインポトされると考えられ、 A.mfd 内の B.mfd からも関数を使用することができます。同様に XSLT ファイルから関数を A.mfd にインポトすると、これもローカルなインポトと考えられます。

ライブラル管理ウィンドウからすべてのローカルとグロー・シルのインポートを確認し管理することができます。ライブラをインポートするコよ以下を行います:

1. ライブラリウイボウのベースのライブラリの追加/削除ボタンをクリックします。ライブラルの管理ウイボウか開かれます。



- 2. 以下の内のつを行います:
 - (現在のマッピング内のスコープのみで) ローカルライブラルとて関数をインポートする場合現在のマッピング名の下の「追加」 ボタンをクトックします。
 - (プログラムレベルで) グローノ ジレライブラノとして関数をインポートする場合 グローノ シルライブラリのインポートの横の「追加」 ボタンをクリックします。
- メモ ライブラをローカルでインポートする場合、ライブラリファイルの ひをマピング に対して相対的に設定することが可能です。 グローバルにインポートされたライブラノでは、 インポートされたライブラノへの いば常に絶対的ない いです。

競合する関数名

以下のレベルで同じ名前を持つ関数が発生するシチュエーションに遭遇する可能性があります。

- メインマパング内
- ライブラルローカルでインポートとされている場合
- ライブラルジロー いでんポートとされている場合

このようよンチュエーションに遭遇した場合、曖昧さを回避するためこMapForceは上記の順序で関数を呼び出す試みを行います。すな わち、ローカルでインポートされたライブラル内に同じ名前の関数が存在する場合、マメビング内で直接定義されている関数は前例を取りま す。また、(両方の関数が同じ名前を持つことを想定して)ローカルでインポートされている関数はグロー・シルでインポートされている関数より 前例を優先します。

同じ名前が存在する複数の関数では、上記のレールに従い「優位な」関数が呼び出されます。その他の曖昧な関数名はブロックされます。

7.8.2 相対的なライブラリパス

<u>ローカルとグローマルなライブラ</u>ノで説明されているとおりライブラルバグローマルではなくローカルにインポートされていることを前提として、インポートされたライブラリファイルのパマをマッピングデザインファイル(.mfd)に対して相対的に設定することができます。

ライブラル なを相対的に設定するオプションオジキュメントレベルで ローカルにインポートされたライブラル されしてのみ適用することができます。マノビングがプログラムレベルで グローノ ジレにインポートされる場合、ノジレズ常に絶対的です。

マッピングデザインファイルに対して相対的にライブラリパスを設定する方法:

1. ライブラリウィボウのベースのライブラリの追加/削除をクリックします。ライブラリの管理ウィンボウか開かれます。

ライブラリの管理	×
E 📴 BuildHierarchyRecursive.mfd	-
日 ユーザー定義関数	
user.BuildFileHierarchy	
□ 自分のライブラリのインポート 追加	
🖃 🔂 MyCustomLibrary.xslt 🔀 オプション	
日期数	
f_x MyCustomLibrary.func:date-parseYYYYMMDD	
f_x MyCustomLibrary.func:datetime-addYears	
f_x MyCustomLibrary.func:time-parseHHMMSS	
自分のライブラリのインポート	
😢 グローバルライブラリのインポート 追加	-

2. 注目するライブラルの横の「オプション」をクリックします。(おさはライブラルを右クリックして、コンテキストメニューからを「オプション」 選択します、

😢 Library Import Options					
File:	C:\Libraries\MyCustomLibrary.xslt				
☑ <u>S</u> ave file pa	th relative to MFD file				
		OK Cancel			

- 3. 「MFD ファイルに対してファイル マを相対的に保存する」チェックボックスを選択します。
- メモ チェックボックスが灰色表示されている場合、ライブラルジロー・シルではなくローカルにインポートされていることを確認してくたさい。

チェックボックスが有効化されている場合、「名前を付けて保存」メニューコマトを使用してマッピングファイルを新規のディレクトリニ保存する際にMapForce は参照されるライブラリファイルへのいなを記録し更新します。また、ライブラリファイルがマッピングファイルと同じディレクトリ に存在する場合、ディスク上の新しい場所にディレクトリ全体を移動してもいなの参照が壊されることはありません。ユンポーネント上で相対的 ないなを使用するを参照してくたさい。

「MFD ファイルに対してファイル マを相対的に保存する」、チェックボックスは、マパンクファイルに対して相対的になることを指定し、生成されたコードには影響を与えません。生成されたコード内でどのようにライブラノ参照が処理されるかについては、多種の実行環境内の、文を参照してくたさい。

7.9 正規表現

MapForce マピングのデザイン時、以下のコンテキストで正規表現("regex")を使用することができます。

- <u>match-pattern</u> と<u>tokenize-regexp</u> 関数
- ノード関数が適用されるノードをフィルターするため。詳細に関してはノード関数とデフォルトの適用を参照してくたきい。

XSLT とXquery のため正規表現構文はXML スキーマパト 2: データベースセカンドエディションの付属 F データベースセカンドエディションの付属 F 内で定義されています。

メモ C++、C#、おさま Java コードを生成する際には、正規表現構文の高度な機能が異なる可能性があります。詳細に関しては 各言語の正規表現ドキュメトを確認してくたさい。

用語

tokenize-regexp 関数を基本の分析正規表現用語を検証してみましょう。この関数は正規表現の助けを借りてテキストを文字列の シーケンスに分割します。これを達成するこれ、関数は次の入力パラメーターを取ります

input	関数によ処理される入力文字列。正規表現はこの文字列上で操作されます。
pattern	適用される実際の正規表現ノターシ。
flags	これは正規表現かどのように解釈されるかを決定する追加オプション(フラグ)を定義する任意の、ラメーターです。下 の【フラグ】を確認してください。

下のマルングでは、入力文字列は[Altova MapForce]です。pattern パラメーターは空白文字で、正規表現フラグは使用されていないとはご注意してくたさい。



これは空白文字が発生する都度テキストの分割を引き起こします。マピングの出力:



tokenize-regexp 関数は一致した文字を結果から除外することに注意してください。つまり、このサンプルの空白文字は出力から除外されています。

上のサンプルは基本で、同じ結果を正規表現を使用せず、tokenize 関数を使用して達成することができます。実際的なシナルでは pattern パラメーターは更に複雑な正規表現を含んでいます。正規表現は次のどれかによ、構成されることができます。

- リテラル
- 文字クラス
- 文字範囲

- 否定/ラス
- 水文字
- 限定子

リテラル

リテラルを使用して書かれている通り(文字通りに文字を一致します。例えば、入力文字列がabracadabra の場合、pattern はテラルbr で出力は以下の通りです。

```
<items>
<item>a</item>
<item>acada</item>
<item>a</item>
</items>
```

リテラルbr は入力文字列 abracadabra内に2個の一致を持っています。出力から一致する文字を削除した後、上記の3個の文字列のシーケンスが生成されます。

文字クラス

角かこ([と])内に文字のセナカ含まれており、これが文字クラスを作成します。文字クラス内の文字の1つのみが一致します。例

- パターン [aeiou] は小文字の母音アルファベトに一致します。
- パターン [mj] ust は "must" と "just" に一致します。
- メモ パターメニ含まれる値は大文字と小文字で区別されるとう点に注意してくたさい。小文字の "a" は大文字の "A" にマッチしません。大文字と小文字を区別したい ようにする コま、i フラグを使用してくたさい。以下を参照してくたさい。

文字範囲

[a-z] を使用して2つの文字間の範囲を作成します。一度にマッチする文字は一文字だけである点に注意してくたさい。例えば、パターン [a-z] は ″a″ から ″z″ までの小文字アルファベットに一致します。

否定クラス

角括弧の開始直後にいた記号(^)を使用することで、その文字クラスを否定したこと」ないます。例えば、パターン[^a-z] は改行文字を含む文字列クラスに含まれていない文字に一致します。

任意の文字を一致する方法

改行文字を除く任意の単一文字に一致させるためゴボット(.) メタ文字を使用します。例えば、パマーン... は全ての単一文字に一致します。

限定子

正規表現内で限定子は一致が発生するために許可される文字ませまサブ式の数量を定義します。

?	直前のアイテムのゼロませまは度の発生に一致します。例えば、パターンmo?は"m"と"mo"に 一致します。
+	直前のアイテムの1度ませま複数の発生に一致します。例えば、/ ターン mo+ は "mo"、"moo"、"mooo" などに一致します。

*	直前のアイテムのゼロおける複数の発生に一致します。
{min,max}	<i>min とmax の間の</i> 発生数を一致させます。例えば、パターンmo{1,3} は "mo"、 "moo"、 および "mooo" に一致します。

かこ

かこ(と)は正規表現のいやをグループ分けするために使用されます。(1文字とは異なりませる代替のある以下参照)サブ式に限定式を適用するために使用されます。

代替

垂直バー(パイプ)文字 | は "or" を意味します。 | により区切られたサブ式の一部に一致するための使用することができます。 例えば パマ ーン (horse | make) sense は "horse sense" と "make sense" に一致します。

フラグ

以下に示されるのはオプションの、ラメーターで、正規表現の解釈方法を指定するために使用されます。各プラグは文字に対応します。文字にどの順序でも使用することができ、繰り返して使用することもできます。

S	このフラブが存在する場合、「全てドット」モード内で一致プロセスが操作されます。
	s フラグがセナされていれば、入力文字列に"hello"と"world"が2 つの異なる行にある場合でも、hello*world とう正規表現によりマッチが行われます。
m	このフラブが存在する場合、「複数行」モード内で一致プロセスが操作されます。
	複数行ノードでは、ハット記号 なか全ての行頭(文字列全体の開始位置と改行文字の直後に来る文字)に対してマッチするようこなります。
	更はドル記号 \$ が全ての行末(文字列全体の終了位置と改行文字の直前に来る文字)に対してマッチするようになり ます。
	改行に使用される文字は#x0A となります。
i	COJFがが存在する場合、「大文字と小文字を区別する」モード内で一致プロセスが操作されます。例えば、正規表現 [a-z] とi プラグはすべての文字 a-z とA-Z に一致します。
x	このプラブが存在する場合、空白文字はマッチング処理が行われる前に正規表現から削除されます。空白文字は #x09、#x0A、#x0D および#x20 です。
	メモ 文字クラス内の空白文字に期除されません。例、[#x20]。

7.10 関数ライブラリレファレンス

このレファレシスチャプターでは、ライブラ別に整理された「ライブラ」」ウイドウ内で使用することのできるMapForce内蔵の関数に関して説明されています。

core ライブラルは、C++、C#、Java 言語で以下のうち1つで使用することのできる関数のコレクションです:XQuery、XPath、おけよXSLT。 lang ライブラルは、言 C++、C#、とJava 語内で使用することのできる関数です。他のライブラルは、出力の異なる型に関連した関数を含みます。

ょ

- ライブラリウィボウ内の関数のライブラルはマンビングの変換言語により異ないます。詳細に関しては、下のテーブルを参照してくたさい。
- XQuery、C#、C++、Java、およびBuilt-In はMapForce Professional おはEnterprise Edition を必要とします。
- シーケンスを扱う XPath 2.0 関数の一部は現在使用することができません。

下のテーブルは各言語内でサポートされている関数をライブラ別にリストしています。

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
avg		۲	۲	۲	۲	۲	۲	۲
count	۲	۲	۲	۲	۲	۲	۲	۲
max		۲	۲	۲	۲	۲	۲	۲
max-string		۲	۲	۲	۲	۲	۲	۲
min		۲	۲	۲	۲	۲	۲	۲
min-string		۲	۲	۲	۲	۲	۲	۲
string-join		۲	۲	۲	۲	۲	۲	۲
sum	۲	۲	۲	۲	۲	۲	۲	۲

core | aggregate functions (集計関数)

core | conversion functions (変換関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
boolean	۲	۲	۲	۲	۲	۲	۲	۲
format-date		۲	۲		۲	۲	۲	۲
format-dateTime		۲	۲			۲	۲	۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
format-number	۲	۲	۲		۲	۲	۲	۲
format-time		۲	۲		۲	۲	۲	۲
number	۲	۲	۲	۲	۲	۲	۲	۲
parse-date					۲	۲	۲	۲
parse-dateTime					۲	۲	۲	۲
parse-number					۲	۲	۲	۲
parse-time					۲	۲	۲	۲
string	۲	۲	۲	۲	۲	۲	۲	۲

core | file path functions (ファイル 以関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
get-fileext	۲	۲	۲	۲	۲	۲	۲	۲
get-folder	۲	۲	۲	۲	۲	۲	۲	۲
main-mfd-filepath	۲	۲	۲	۲	۲	۲	۲	۲
mfd-filepath	۲	۲	۲	۲	۲	۲	۲	۲
remove-fileext	۲	۲	۲	۲	۲	۲	۲	۲
remove-folder	۲	۲	۲	۲	۲	۲	۲	۲
replace-fileext	۲	۲	۲	۲	۲	۲	۲	۲
resolve-filepath	۲	۲	۲	۲	۲	۲	۲	۲

core | generator functions (ジェネレーター関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
auto-number	۲	۲	۲	۲	۲	۲	۲	۲

core | logical functions (論理関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
equal	۲	۲	۲	۲	۲	۲	۲	۲
equal-or-greater	۲	۲	۲	۲	۲	۲	۲	۲
equal-or-less	۲	۲	۲	۲	۲	۲	۲	۲
greater	۲	۲	۲	۲	۲	۲	۲	۲
less	۲	۲	۲	۲	۲	۲	۲	۲
logical-and	۲	۲	۲	۲	۲	۲	۲	۲
logical-not	۲	۲	۲	۲	۲	۲	۲	۲
logical-or	۲	۲	۲	۲	۲	۲	۲	۲
not-equal	۲	۲	۲	۲	۲	۲	۲	۲

core | math functions (数学関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
add	۲	۲	۲	۲	۲	۲	۲	۲
ceiling	۲	۲	۲	۲	۲	۲	۲	۲
divide	۲	۲	۲	۲	۲	۲	۲	۲
floor	۲	۲	۲	۲	۲	۲	۲	۲
modulus	۲	۲	۲	۲	۲	۲	۲	۲
multiply	۲	۲	۲	۲	۲	۲	۲	۲
round	۲	۲	۲	۲	۲	۲	۲	۲
round-precision					۲	۲	۲	۲
subtract	۲	۲	۲	۲	۲	۲	۲	۲

core | node functions (ノード関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
is-xsi-nil	۲	۲	۲	۲	۲	۲	۲	۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
local-name	۲	۲	۲	۲	۲	۲	۲	۲
node-name		۲	۲	۲	۲	۲	۲	۲
set-xsi-nil		۲	۲	۲	۲	۲	۲	۲
static-node-annotation	۲	۲	۲	۲	۲	۲	۲	۲
static-node-name	۲	۲	۲	۲	۲	۲	۲	۲
substitute-missing-with-xsi-nil		۲	۲	۲	۲	۲	۲	۲

core | QName functions (QName 関数)

関数	XSL T 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
QName		۲	۲	۲	۲	۲	۲	۲
local-name-from-QName		۲	۲	۲	۲	۲	۲	۲
namespace-uri-from-QName		۲	۲	۲	۲	۲	۲	۲

core | sequence functions (シーケンス関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
distinct-values		۲	۲	۲	۲	۲	۲	۲
exists	۲	۲	۲	۲	۲	۲	۲	۲
first-items		۲	۲	۲	۲	۲	۲	۲
generate-sequence		۲	۲	۲	۲	۲	۲	۲
group-adjacent		۲	۲		۲	۲	۲	۲
group-by		۲	۲		۲	۲	۲	۲
group-ending-with		۲	۲		۲	۲	۲	۲
group-into-blocks		۲	۲		۲	۲	۲	۲
group-starting-with		۲	۲		۲	۲	۲	۲
item-at		۲	۲	۲	۲	۲	۲	۲
items-from-till		۲	۲	۲	۲	۲	۲	۲

738	関数
-----	----

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
last-items		۲	۲	۲	۲	۲	۲	۲
not-exists	۲	۲	۲	۲	۲	۲	۲	۲
position	۲	۲	۲	۲	۲	۲	۲	۲
replicate-item		۲	۲	۲	۲	۲	۲	۲
replicate-sequence		۲	۲	۲	۲	۲	۲	۲
set-empty		۲	۲	۲	۲	۲	۲	۲
skip-first-items		۲	۲	۲	۲	۲	۲	۲
substitute-missing	۲	۲	۲	۲	۲	۲	۲	۲

core | string functions (文字列関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
char-from-code		۲	۲	۲	۲	۲	۲	۲
code-from-char		۲	۲	۲	۲	۲	۲	۲
concat	۲	۲	۲	۲	۲	۲	۲	۲
contains	۲	۲	۲	۲	۲	۲	۲	۲
normalize-space	۲	۲	۲	۲	۲	۲	۲	۲
starts-with	۲	۲	۲	۲	۲	۲	۲	۲
string-length	۲	۲	۲	۲	۲	۲	۲	۲
substring	۲	۲	۲	۲	۲	۲	۲	۲
substring-after	۲	۲	۲	۲	۲	۲	۲	۲
substring-before	۲	۲	۲	۲	۲	۲	۲	۲
tokenize		۲	۲	۲	۲	۲	۲	۲
tokenize-by-length		۲	۲	۲	۲	۲	۲	۲
tokenize-regexp		۲	۲	۲	۲	۲	۲	۲
translate	۲	۲	۲	۲	۲	۲	۲	۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
is-not-null					۲	۲	۲	۲
is-null					۲	۲	۲	۲
set-null					۲	۲	۲	۲
substitute-null					۲	۲	۲	۲

db | null processing functions (NULL 処理関数)

edifact | datetime functions (日付時刻関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
auto-format					۲	۲	۲	۲
to-date					۲	۲	۲	۲
to-datetime					۲	۲	۲	۲
to-duration					۲	۲	۲	۲
to-time					۲	۲	۲	۲

lang | datetime functions (日付時刻関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
age					۲	۲	۲	۲
convert-to-utc					۲	۲	۲	۲
date-from-datetime					۲	۲	۲	۲
datetime-add					۲	۲	۲	۲
datetime-diff					۲	۲	۲	۲
datetime-from-date-and-time					۲	۲	۲	۲
datetime-from-parts					۲	۲	۲	۲
day-from-datetime					۲	۲	۲	۲
day-from-duration					۲	۲	۲	۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
duration-add					۲	۲	۲	۲
duration-from-parts					۲	۲	۲	۲
duration-subtract					۲	۲	۲	۲
hour-from-datetime					۲	۲	۲	۲
hour-from-duration					۲	۲	۲	۲
leapyear					۲	۲	۲	۲
millisecond-from-datetime					۲	۲	۲	۲
millisecond-from-duration					۲	۲	۲	۲
minute-from-datetime					۲	۲	۲	۲
minute-from-duration					۲	۲	۲	۲
month-from-datetime					۲	۲	۲	۲
month-from-duration					۲	۲	۲	۲
now					۲	۲	۲	۲
remove-timezone					۲	۲	۲	۲
second-from-datetime					۲	۲	۲	۲
second-from-duration					۲	۲	۲	۲
time-from-datetime					۲	۲	۲	۲
timezone					۲	۲	۲	۲
weekday					۲	۲	۲	۲
weeknumber					۲	۲	۲	۲
year-from-datetime					۲	۲	۲	۲
year-from-duration					۲	۲	۲	۲

lang | file functions (ファイル関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built-In
read-binary-file								۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built-In
write-binary-file								۲

lang | generator functions (ジェネレーター関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
create-guid					۲	۲	۲	۲

lang | logical functions (論理関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
logical-xor					۲	۲	۲	۲
negative					۲	۲	۲	۲
numeric					۲	۲	۲	۲
positive					۲	۲	۲	۲

lang | math functions (数学関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
abs					۲	۲	۲	۲
acos					۲	۲	۲	۲
asin					۲	۲	۲	۲
atan					۲	۲	۲	۲
cos					۲	۲	۲	۲
degrees					۲	۲	۲	۲
divide-integer					۲	۲	۲	۲
ехр					۲	۲	۲	۲
log					۲	۲	۲	۲
log10					۲	۲	۲	۲
max					۲	۲	۲	۲

関数

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
min					۲	۲	۲	۲
pi					۲	۲	۲	۲
pow					۲	۲	۲	۲
radians					۲	۲	۲	۲
random					۲	۲	۲	۲
sin					۲	۲	۲	۲
sqrt					۲	۲	۲	۲
tan					۲	۲	۲	۲
unary-minus					۲	۲	۲	۲

lang | QName functions (QName 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
QName-as-string					۲	۲	۲	۲
string-as-QName					۲	۲	۲	۲

lang | string functions (文字列関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
capitalize					۲	۲	۲	۲
charset-decode								۲
charset-encode								۲
count-substring					۲	۲	۲	۲
empty					۲	۲	۲	۲
find-substring					۲	۲	۲	۲
format-guid-string					۲	۲	۲	۲
left					۲	۲	۲	۲
left-trim					۲	۲	۲	۲

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
lowercase					۲	۲	۲	۲
match-pattern					۲		۲	۲
pad-string-left					۲	۲	۲	۲
pad-string-right					۲	۲	۲	۲
repeat-string					۲	۲	۲	۲
replace					۲	۲	۲	۲
reversefind-substring					۲	۲	۲	۲
right					۲	۲	۲	۲
right-trim					۲	۲	۲	۲
string-compare-ignore-case					۲	۲	۲	۲
string-compare-ignore-case					۲	۲	۲	۲
uppercase					۲	۲	۲	۲

mime | mime

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built-In
content-encoding								۲
content-type								۲
decode-mime-entity								۲
mime-entity								۲

xbrl | Unit helpers

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
xbrl-measure-currency					۲	۲	۲	۲
xbrl-measure-pure					۲	۲	۲	۲
xbrl-measure-shares					۲	۲	۲	۲

xlsx | datetime functions (日付時刻関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
columnname-to-index		۲	۲		۲		۲	۲
date-to-xlsx		۲	۲		۲		۲	۲
datetime-to-xlsx		۲	۲		۲		۲	۲
index-to-columnname		۲	۲		۲		۲	۲
time-to-xlsx		۲	۲		۲		۲	۲
xlsx-to-date		۲	۲		۲		۲	۲
xlsx-to-datetime		۲	۲		۲		۲	۲
xlsx-to-time		۲	۲		۲		۲	۲

xpath2 | accessors

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
base-uri		۲	۲	۲				
node-name		۲	۲	۲				
string		۲	۲	۲				

xpath2 | anyURI functions (anyURI 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
resolve-uri		۲	۲	۲				

xpath2 | boolean functions (ブール値関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
false		۲	۲	۲				
true		۲	۲	۲				

xpath2 | constructors

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
xs:ENTITY		۲	۲	۲				
xs:ID		۲	۲	۲				
xs:IDREF		۲	۲	۲				
xs:NCName		۲	۲	۲				
xs:NMTOKEN		۲	۲	۲				
xs:Name		۲	۲	۲				
xs:QName		۲	۲	۲				
xs:anyURI		۲	۲	۲				
xs:base64Binary		۲	۲	۲				
xs:boolean		۲	۲	۲				
xs:byte		۲	۲	۲				
xs:date		۲	۲	۲				
xs:dateTime		۲	۲	۲				
xs:dayTimeDuration		۲	۲	۲				
xs:decimal		۲	۲	۲				
xs:double		۲	۲	۲				
xs:duration		۲	۲	۲				
xs:float		۲	۲	۲				
xs:gDay		۲	۲	۲				
xs:gMonth		۲	۲	۲				
xs:gMonthDay		۲	۲	۲				
xs:gYear		۲	۲	۲				
xs:gYearMonth		۲	۲	۲				
xs:hexBinary		۲	۲	۲				

746	関数	

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
xs:int		۲	۲	۲				
xs:integer		۲	۲	۲				
xs:language		۲	۲	۲				
xs:long		۲	۲	۲				
xs:negativeInteger		۲	۲	۲				
xs:nonNegativeInteger		۲	۲	۲				
xs:nonPositiveInteger		۲	۲	۲				
xs:normalizedString		۲	۲	۲				
xs:positiveInteger		۲	۲	۲				
xs:short		۲	۲	۲				
xs:string		۲	۲	۲				
xs:time		۲	۲	۲				
xs:token		۲	۲	۲				
xs:unsignedByte		۲	۲	۲				
xs:unsignedInt		۲	۲	۲				
xs:unsignedLong		۲	۲	۲				
xs:unsignedShort		۲	۲	۲				
xs:untypedAtomic		۲	۲	۲				
xs:yearMonthDuration		۲	۲	۲				

xpath2 | context functions (コンテキスト 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
current-date		۲	۲	۲				
current-dateTime		۲	۲	۲				
current-time		۲	۲	۲				
default-collation		۲	۲	۲				

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
implicit-timezone		۲	۲	۲				
last		۲	۲	۲				

xpath2 | durations, date and time functions (期間、日付、および時刻関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
adjust-date-to-timezone		۲	۲	۲				
adjust-date-to-timezone		۲	۲	۲				
adjust-dateTime-to-timezone		۲	۲	۲				
adjust-dateTime-to-timezone		۲	۲	۲				
adjust-time-to-timezone		۲	۲	۲				
adjust-time-to-timezone		۲	۲	۲				
day-from-date		۲	۲	۲				
day-from-dateTime		۲	۲	۲				
days-from-duration		۲	۲	۲				
hours-from-dateTime		۲	۲	۲				
hours-from-duration		۲	۲	۲				
hours-from-time		۲	۲	۲				
minutes-from-dateTime		۲	۲	۲				
minutes-from-duration		۲	۲	۲				
minutes-from-time		۲	۲	۲				
month-from-date		۲	۲	۲				
month-from-dateTime		۲	۲	۲				
months-from-duration		۲	۲	۲				
seconds-from-dateTime		۲	۲	۲				
seconds-from-duration		۲	۲	۲				
seconds-from-time		۲	۲	۲				

関数

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
subtract-dateTimes		۲	۲	۲				
subtract-dates		۲	۲	۲				
subtract-times		۲	۲	۲				
timezone-from-date		۲	۲	۲				
timezone-from-dateTime		۲	۲	۲				
timezone-from-time		۲	۲	۲				
year-from-date		۲	۲	۲				
year-from-dateTime		۲	۲	۲				
years-from-duration		۲	۲	۲				

xpath2 | node functions (ノード関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
lang		۲	۲	۲				
local-name		۲	۲	۲				
local-name		۲	۲	۲				
name		۲	۲	۲				
name		۲	۲	۲				
namespace-uri		۲	۲	۲				
namespace-uri		۲	۲	۲				
number		۲	۲	۲				
number		۲	۲	۲				

xpath2 | numeric functions (数値関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
abs		۲	۲	۲				
round-half-to-even		۲	۲	۲				

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQuer y 1.0	C#	C++	Java	Built- In
codepoints-to-string		۲	۲	۲				
compare		۲	۲	۲				
compare		۲	۲	۲				
ends-with		۲	۲	۲				
ends-with		۲	۲	۲				
lower-case		۲	۲	۲				
matches		۲	۲	۲				
normalize-unicode		۲	۲	۲				
replace		۲	۲	۲				
starts-with		۲	۲	۲				
string-to-codepoints		۲	۲	۲				
substring-after		۲	۲	۲				
substring-before		۲	۲	۲				
upper-case		۲	۲	۲				

xpath2 | string functions (文字列関数)

xpath3 | external information functions (外部情報関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
available-environment-variables			۲					
environment-variable			۲					
unparsed-text			۲					
unparsed-text-available			۲					
unparsed-text-lines			•					

xpath3 | formatting functions (書式設定関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
format-date			۲					
format-dateTime			۲					
format-integer			۲					
format-time			۲					

xpath3 | math functions (数学関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
acos			۲					
asin			۲					
atan			۲					
atan2			۲					
cos			۲					
ехр			۲					
exp10			۲					
log			۲					
log10			۲					
рі			۲					
pow			۲					
sin			۲					
sqrt			۲					
tan			۲					

xpath3 | URI functions (URI 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
encode-for-uri			۲					

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
escape-html-uri			۲					
iri-to-uri			۲					

xslt10 | xpath functions (xpath 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
lang	۲							
last	۲							
local-name	۲	۲	۲					
name	۲	۲	۲					
namespace-uri	۲	۲	۲					
position	۲							

xslt10 | xslt functions (xslt 関数)

関数	XSLT 1.0	XSLT 2.0	XSLT 3.0	XQue ry 1.0	C#	C++	Java	Built- In
currrent	۲							
document	۲							
element-available	۲							
function-available	۲							
generate-id	۲	۲	۲					
system-property	۲	۲	۲					
unparsed-entity-uri	۲							

7.10.1 core | aggregate functions (集計関数)

「集計」は合計、カウト、平均など1つの結果を取得するために同じ型の複数の値を処理することを意味します。MapForce内でavg、 count、max、および他の集計関数を使用してデータの集計を行うことができます。

以下の2つの引数はすべての集計関数に共通します

関数

- parent-context。この引数はオプションです。これによりデフォルトのマンとグゴンテキストをオーバーライドすることができます(この結果、関数のスコープが、おさよ、関数が反復する値が変更されます)。成功例に関しては<u>例 親コンテキストの変更</u>で詳細に説明されています。
- 2. values。この引数は、処理する値を与えるノースアイテムは接続されている必要があります。例えば、下に示されるマンピング内で sum 関数はノース XML ファイルを基にする数値のシーケンスを入力としてとます。ソース XML ファイル内の各アイテムのすっかに multiply 関数はアイテムの価格を個数と掛けが値を取得し sum 関数についます。sum 関数はすべての入力の値を集計 し、マンピングの出力である総計結果を生成します。...\MapForceExamples\Tutorial\ ディレクトリ内でこのマンピングを見 つけることができます。

	ipo File: ipo.xml File/String Sipo:purchaseOrder			
Þ	orderDate	Þ		
₽	⊕ ♦ ShipTo TYPE	₿		
₽		₿		
Þ	ipo:comment	Þ		
¢		Þ		
¢	⊡ <> item	Þ		
¢	= partNum	Þ		
¢	>productName	Þ	_f _x multiply f _x sum	➡ total
⊧	Quantity	-	value1 parent-context	total
¢	price	-	value2	
⊳	ipo:comment	⊧	•	
⊳	shipDate	⊳		
	-	4		

SimpleTotal.mfd

min、max、sum とavg、などの集計関数の一部は排他的に数値とのみ作業することができます。これらの関数の入力データがdecimal データ型に変換され、処理されます。

7.10.1.1 avg

入力シーケンスから得られた植の平均値を返します。空のセナの平均は結果は空のセナレゴンます。

	$f_{\rm 8}$ avg	
ļ	parent-context	
Ç	>values	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親エンテキスト</u> を提供します。サノブル親エンテキストの変更で詳細に説明されています。
values	この引数は実際のデータを与えるノースアイテムは接続されている必要があります。与えられる引数の値は数値 である必要があります。

サンプル

サンプルキー別ココードをグループ分けする方法を参照してくたさい

7.10.1.2 count

入力シーケンスを構成する個別のアイテムの数量を返します。入力が空セナのカウナトは0 しなります。

	f_8 count	
parent-context		
Ç	>nodes/rows	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

この関数はXSLT 1.0 内では機能が制限されています。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親エンテキスト</u> を提供します。サノブル親エンテキストの変更で詳細に説明されています。
nodes/rows	この引数は数えられるソースアイテムは接続されている必要があります。

サンプル

サノプル親エレテキストの変更、サノプルテーブル行のカウトを参照してくたさい。

7.10.1.3 max

入力シーケンス内の全ての数値の最小値を返します。空のセナの最大値は空のセナレゴンます。

f _x max	
parent-context	
values	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親コンテキスト</u> を提供します。サノブル親コンテキストの変更で詳細に説明されています。
values	この引数は実際のデータを与えるノースアイテムは接続されている必要かあります。与えられる引数の値は数値 である必要かあります。文字列のシーケンスから最大値を得るはt <u>max-string</u> 関数を使用してくたさい。

サンプル

サンプル・キー別コレコードをグループ分けする方法を参照してくたさい

7.10.1.4 max-string

入力シーケンス内の全ての文字列の値の最大値を返します。例えば、max-string("a"、"b"、"c") は"c" を返します。strings 引数が空のセルの場合、関数は空のセルを返します。

	$\mathbf{f}_{\mathbf{x}}$	max-string	
ţ	parent-context		racult N
Ç	> sti	rings	

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親エテキスト</u> を提供します。サノブル親エテキストの変更で詳細に説明されています。

引数	説明
strings	この引数は実際のデータを与えるソースアイテムは接続されている必要かあります。与えられた引数の値は xs:stringのシーケンズ(ゼロおは北複数)である必要かあります。

7.10.1.5 min

入力シーケンス内の全ての数値の最小値を返します。空のセナの最小値は空のセナレゴンます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの別数。親コンテキストを提供します。サンプル・親コンテキストの変更で詳細に説明されています。
values	この引数は実際のデータを与えるノースアイテムは接続されている必要があります。与えられる引数の値は数値 である必要があります。文字列のノーケンスから最小値を得るは <u>min-string</u> 関数を使用してくたさい。

サンプル サンプル キー別 ユコードをグループ分けする方法を参照してくたさい

7.10.1.6 min-string

入力シーケンス内の全ての文字列の値の最小値を返します。例えば、min-string("a"、"b"、"c") は"c" を返します。strings 引数が空のセルの場合、関数は空のセルを返します。

	$f_{\rm X}$ min-string	
ţ	>parent-context	
Ç	> strings	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親エテキスト</u> を提供します。サノブル親エテキストの変更で詳細に説明されています。
strings	この引数は実際のデータを与えるノースアイテムは接続されている必要があります。与えられた引数の値は xs:stringのシーケンズ、ゼロおけは複数)である必要があります。

7.10.1.7 string-join

入力シーケンスのすべての値を区切り文字として使用することを選択した文字列により区切られた一つの文字列に連結します。strings引数が空に設定されている場合関数は空の文字列を返します。

	$f_{\rm x}$ string-join	
ţ	parent-context	
Ç	> strings	result 🕻
ţ	>delimiter	

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
parent-context	オプションの引数。 <u>親エンテキスト</u> を提供します。 <u>サンプル 親エンテキストの変更</u> で詳細に説明されています。
strings	この引数は実際のデータを与えるノースアイテムは接続されている必要があります。与えられた引数の値は xs:stringのノーケンズ(ゼロまたは複数)である必要があります。
delimiter	オプションの引数。2つの連続した文字列の間に区切り文字を挿入するように指定します。

サンプル

以下に示されるサンプルではノースXML ファイルゴお以下の番号を持つ4つのArticle アイテムか含まれています: 1、2、3と4。


定数は "#" を区切り文字として提供しています。マンピングの結果にこのため1#2#3#4 です。区切り文字を提供しない場合、結果は 1234 しなります。

7.10.1.8 sum

入力シーケンス内のすべての値の合計値を返します。空のセナの合計は0 しています。

	f ₈ sum	
ţ	parent-context	
Ç	>values	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

15メーター

引数	説明
parent-context	オプションの引数。 <u>親エレテキスト</u> を提供します。サノブル親エレテキストの変更で詳細に説明されています。
values	この引数は実際のデータを与えるノースアイテムは接続されている必要があります。与えられる引数の値は数値 である必要があります。

サンプル サンプルノードの値の集計を参照してくたさい

7.10.2 core | conversion functions (変換関数)

明示的なデータ型変換をサポートするコよ、conversion ライブラル内で複数の変換関数を使用することができます。多くの場合 MapForce は必要な変換を自動的に作成するため、変換関数は常に必要ではありません。変換関数は日付および時刻の値を書式設定する場合、おけは値を比較する場合によいて通常役に立ちます。例えば、マメピングアイテムの一部が(整数と文字列など)異なる型の 場合 <u>number</u> 変換関数を使用して数値の比較を強制することができます。

7.10.2.1 boolean

arg の値をブール値の値として変換します。(equal、greater や フルターとif-else 条件 などの理論関数と作業する際に役に立ちます。ブール値 false を得るこは空の文字列、おける数値 0 を引数として与えます。ブール値 true を得るこは空の文字列、おける数値 1 を引数として与えます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
arg	必須の別数。変換される値を提供します。

7.10.2.2 format-date

型 xs:date の日付の植を特定のオプションゴ従い文字列と書式に変換します。

f_8 format-	f ₈ format-date		
⊳ value			
¢ format	result 🛱		
language			

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

引数	説明
value	変換するxs:date 値です。
format	日付がフォーマナされる方法を識別する文字列をフォーマナします。この引数は <u>format-dateTime</u> 関数内の format 引数と同じ方法で使用されます。
language	オプションの引数。与えられると、月の名前と週の曜日か特定の言語で返されます。有効な値

引数	説明	
	de	ドイン語
	en (default)	英語
	es	ネペル語
	fr	フランス語
	ja	日本語

サンプル

以下のマッピングは以下のようなフォーマットで現在の日付を出力します: "25 March 2020, Wednesday"。この植をスペイン語に翻訳 するコよ language 引数の値を es (設定します。



上のマピングはBuilt-in、C++、C#、おさはJava 変換言語のためにデザインされています。XSLT 2.0 では、同じ結果を以下のマピングで得ることができます:



7.10.2.3 format-dateTime

型xs:dateTimeの値を文字列に変換します。日時の文字列の表示はformat引数の値に従いフォーマナされます。

∫ ₈ format-d	$f_{\!\scriptscriptstyle R}$ format-dateTime		
⊳value			
₽ format] result 🖗		
language			

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明		
value	変換するxs:dateTime 値です。		
format	value がフォーマナされる方法を識別する文字列をフォーマナします。下の「説明」を確認してくたさい。		
language	オプションの引数。与えられると、月の名前と週の曜日か特定の言語で返されます。有効な値		
	de	ドイン語	
	en (default)	英語	
	es	えゃた語	
	fr	アン語	
	ja	日本語	

メモ 値はカーゲトの型へキャストされるため、関数の出力 (result) か文字列以外のアイテムへ接続されている場合、書式か失われる可能性かあります。自動的なキャストを無効化するコはターゲトコンポーネント内のコンポーネント設定内のターゲット型にターゲット値をキャストする チェックボックスをクリアします。

リマーク

format 引数 は、[Y] / [M] / [D] などの角括弧に囲まれた変数マーカーと呼ばれる文字列にと、構成されます。角括弧外の文字はレデラルな文字です。角括弧を定数文字として使用する場合、括弧を重ねあわせて記述する必要があります。

各変数マーカーは、表示する日付や時刻を表す識別子、オプィンの書式修飾子、プレゼテーション修飾子、幅修飾子により構成され、 これらオプィンの修飾子はエンマにより分割されます。

```
format := (literal | argument)*
argument := [component(format)?(presentation)?(width)?]
width := , min-width ("-" max-width)?
```

コーポーネントは以下のとおしです

識別子	説明	デフォルトの表示
Y	年(絶対値)	4桁の数値(2010)
м	その年の月	1-12
D	その月の日付	1-31
d	その年の日付	1-366

識別子	説明	デフォルトの表示	
F	その週の曜日	曜日の名前(言語に依存)	
w	その年の週	1-53	
w	その月の週	1-5	
Н	時(24時間)	0-23	
h	時(12時間)	1-12	
Р	A.M. おせまP.M.	アルファベト(言語に弦存)	
m	その時間の分	00-59	
s	その分の秒	00-59	
f	小数点以下の秒	数値、小数点1つ	
Z	UTC からのタイムゾーンオフセメ	+08:00	
z	GMT からのタイムゾーンオフセット	GMT+n	

書式修飾子は以下であることができます:

文字	説明	サンプル
1	最初に0を伴われ、数値書式:	1, 2, 3
01	2 桁の数値書式:	01, 02, 03
N	コンポーネントの名前、大文字1	MONDAY, TUESDAY 1)
n	コンポーネントの名前、小文字1	monday, tuesday 1)
Nn	コンポーネントの名前、頭文字が大文字1	Monday, Tuesday

脚注

1.N、n、およびNn修飾子は以下のエレポーネトのみによりサポートされています: M、d、D。

幅の修飾子は必要な場合最小の幅を示す小数が次に続くコンマにと表示されます。オプションで、最大の幅を示す他の小数が次に表示されるダッシュ文字を追加することもできます。例

- [D,2] はおで表示された先頭が0で表示された月の日です。
- [MNn, 3-3] は Jan、 Feb、 Mar など3 文字で表示された月の名前です。

サンプル

format-dateTime 関数を使用して、下のテーブルはxs:dateTime 値のフォーマトのサンプルを表示しています。"「値」列は value 引数に与えられる値を指定します。「書式」列は、format 引数の値を指定します。「結果」 列は、関数によし返される内容を表示しています。

値	走書	結果
2003-11- 03T00:0:00	[D]/[M]/[Y]	2003/03/11
2003-11- 03T00:0:00	[Y]-[M,2]-[D,2]	2003/11/03
2003-11- 03T00:0:00	[Y]-[M,2]-[D,2] [H,2]:[m]:[s]	2003-11-03 00:00:00
2010-06-02T08:02	[Y] [MNn] [D01] [F,3-3] [d] [H]:[m]:[s]. [f]	2010 June 02 Wed 153 8:02:12.054
2010-06-02T08:02	[Y] [MNn] [D01] [F,3-3] [d] [H]:[m]:[s].[f] [z]	2010 June 02 Wed 153 8:02:12.054 GMT+02:00
2010-06-02T08:02	[Y] [MNn] [D1] [F] [H]:[m]:[s].[f] [Z]	2010 June 2 Wednesday 8:02:12.054 +02:00
2010-06-02T08:02	[Y] [MNn] [D] [F,3-3] [H01]:[m]:[s]	2010 June 2 Wed 08:02:12

7.10.2.4 format-number

数値を文字列に変換し、特定のオプションに従い書式設定します。

${ m f}_8$ format-number	
¢ value	
¢ format	
decimal-point-character	result
s grouping-separator	

言語

 $\mathsf{Built-in},\mathsf{C}^{++},\mathsf{C}^{\#},\mathsf{Java},\mathsf{XSLT}\ 1.0,\mathsf{XSLT}\ 2.0,\mathsf{XSLT}\ 3.0_{\circ}$

引数	説明
value	必須の引数。フォーマナする数値を提供します。
format	必須の引数。数値がフォーマナされる方法を識別するフォーマナ文字列を提供します。下の 「説明」を確認してくたさい。
decimal-point-format	オプションの引数。小数点文字として使用される文字を提供します。デフォルトの値はフルストップ.)文字です。
grouping-separator	オプションの別数。数値グループを分割するながユ使用されるセッシーター/デミタ。デフォルトの値はエンマ(,))文字です。

メモ 値はターゲトの型へキャストされるため、関数の出力 (result) か文字列以外のアイテムへ接続されている場合、書式か失われる可能性があります。自動的なキャストを無効化するコはターゲトコンポーネント内のコンポーネント設定内のターゲット型にターゲット値をキャストする チェックボックスをクリアします。

リマーク

format 引数は以下の書式を取ります:

```
format := subformat (;subformat)?
subformat := (prefix)? integer (.fraction)? (suffix)?
prefix := any characters except special characters
suffix := any characters except special characters
integer := (#)* (0)* ( allowing ',' to appear)
fraction := (0)* (#)* (allowing ',' to appear)
```

最初のサブフォーマナは正の数を書式設定するオータンス使用され、2番目のサブフォーマナは負の数をフォーマナするオータンを用されます。1つのサブフォーマナのみが指定されている場合、同じサブフォーマナが負の数のオータンスは使用されますが、マイナスのシンドルがプレ フィックスに追加されます。

特別な文字	デフォルト	説明
0 桁	0	結果内のこの時点で桁が常に表示されます。
桁	#	余剰な先行おけまそ的に0 になる場合を除き、結果の文字列に てこの場所に常に数値が表示されます。
小数点	o	数値の整数部と小数点以下を分けるために使用されます。
grouping-separator	3	桁のグループを区切ります。
パーセト記号	%	数値を100倍して、パーセンテージとして表示します。
パーミル	‰	数値を1000倍して、パーミルとして表示します。

下のテーブルでは書式設定文字列とその結果のサンプルが表示されています。

メモ format-number 関数にお使用される端数処理メンドは「ハーフアップ」です。これは値が0.5 お先大き、または等しい場合丸がられることを意味します。0.5 未満であれば別と捨てが行われます。このメンドは、生成されたコードとビルトイン実行エンジンでのみ使用することができます。XSLT 1.0 では四捨五入のモードが定義されていません。XSLT 2.0 では端数処理モードは偶数への丸めです。

数值	書式設定文字列	結果
1234.5	#,##0.00	1,234.50
123.456	#,##0.00	123.46
1000000	#,##0.00	1,000,000.00
-59	#,##0.00	-59.00
1234	####0.0####	1234.0

数值	書式設定文字列	結果
1234.5	### # 0.0####	1234.5
.00025	### # 0.0####	0.0003
.00035	### # 0.0####	0.0004
0.25	#00%	25%
0.736	#00%	74%
1	#00%	100%
-42	#00%	-4200%
-3.12	#.00;(#.00)	(3.12)
-3.12	#.00;#.00CR	3.12CR

サンプル

下で説明されるマピングはノースXML からデータを読み取りXML ターゲオに書き込みます。ソース内に以下の小数値を含む複数の SinglePrice 要素が存在します: 25, 2.30, 34, 57.50.このマピングにおつのゴールがおます:

- 1. 全ての値の左側をゼロでパドし、重要な部分がを桁になるようします。
- 2. 全ての値の左側をゼロでパドし、重要な部分が2桁になるようします。

これを達成するために、フォーマホ文字列 00000.00 がformat-number 関数の引数とて提供されています。



PreserveFormatting.mfd

この結果、ターゲナト内の値は以下のようこなります。

00025.00 00002.30 00034.00 00057.50 マピングデザインを次の マを使用して見つけることができます: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\PreserveFormatting.mfd。

7.10.2.5 format-time

xs:time 入力值を文字列へ変換します。

$f_{\rm R}$ format-time	
⊳ value	secult [
⊳ format	result L

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value	必須の引数。フォーマナするxs:dateTime 値を提供します。
format	必須の引数。フォーマナ、文字列を提供します。この引数は <u>format-dateTime</u> 関数内の format 引数と同じ方法で使用されます。

サンプル

以下のマンピングは以下の2:15 p.m. ような書式で現在の時刻を出力します。これを達成するコは以下の書式のフォーマナ文字列 [h]:[m] [P] を使用します:

- [h] は12 時間書式の現在の時間数です
- [m] は現在の分数です
- [P] は"a.m." おは"p.m."の部分です。



上のマピングはBuilt-in、C++、C#、おさまJava 変換言語のためにデザインされています。XSLT 2.0 では、同じ結果を以下のマピングで得ることができます。



7.10.2.6 number

arg が文字列ませまプール値である個所で arg の値を数値に変換します。 arg は文字列で、 MapForce は数値とて解析を試みます。 例えば、 "12.56" などの文字列が小数の値 12.56 に変換されます。 arg がブール値 true の場合、数値 1 に変換されます。 arg がブール値 false の場合、数値 0 に変換されます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
arg	必須の別数。変換される値を提供します。

サンプル

下のサンプルでは、最初の定数は型 string で、文字列 "4" を含んで、ます。2番目の定数は、定数 12 を含んで、ます。2つの値が 数値として比較されるためには対い一致する必要がみます。



number 関数を最初の定数に追加するコよ、文字列 "4" の定数を数値 4 に変換します。比較の結果は "true" にないます。 number 関数が使用されない場合結果が 「false" の文字列の比較が発生します。 (すなオち "4" か直接 a に接続されています)。

7.10.2.7 parse-date

文字列を日付に変換します。この関数はparse-dateTime 関数をベースコンたもので、時刻コンポーネントが無視されます。出力はxs:date 型しています。

$f_{\!\scriptscriptstyle R}$ parse-date		
value	racult [
format	result	

言語

Built-in、C++、C#、Java。

パラメーター

引数	説明
value	必須の別数。変換される文字列の値を提供します。
format	必須の引数。フォーマナ、文字列を提供します。この引数は <u>parse-dateTime</u> 関数内の format 引数と同じ方法で使用されます。

サンプル

下のマピングは "01 Apr 2015" を解析し、日付に変換し結果を型 xs:dateのターゲルアイテム(pubdate) に書き込みます。こ れは書式 [D01] [MNn,3-3] [Y] を使用して達成することができます:

- [D01] は2桁で表示された月の日です。
- [MNn, 3-3] は月の名前です。最小と最大の幅は3文字です。
- [Y] は年数です



(XML と名前宣言を除く)結果は以下の通りです:

```
<book>
<title>Linux Bible</title>
<pubdate>2015-04-01</pubdate>
</book>
```

7.10.2.8 parse-dateTime

文字列出て表されている日付と時刻を型 xs:dateTime の値に変換します。

f ₈ parse-dateTime	
⊳ value	ra cult P
⊳format	result

言語

Built-in、C++、C#、Java。

パラメーター

引数	説明
value	変換される文字列の値
format	valueに適用される書式マスクを指定します。

リマーク

書式マスクは以下のコンポーネントによく構成されることができます

コンポーネント	説明	デフォルトの表示
Y	年(絶対値)	4桁の数値(2010)
м	その年の月	1-12
D	その月の日付	1-31
d	その年の日付	1-366
Н	時(24時間)	0-23
h	時(12時間)	1-12
Р	A.M. ませまP.M.	アレファベナ(言語)弦存)
m	その時間の分	00-59
s	その分の秒	00-59
f	小数点以下の秒	数値、小数点1つ
Z	UTC からのタイムゾーンオフセット	+08:00
z	GMT からのタイムゾーシオフセナ	GMT+n

上記のエレポーネントの一部は識別子を取ります(例えば、日付を1桁、ませまれとて解釈するために使用されます):

識別子	説明	サンプル
1	最初に0を伴われ 数値書式:1,2,3,	1, 2, 3
01	2 桁の数値書式:01,02,03,	01, 02, 03
N	コンポーネント名、大文字	FEBRUARY, MARCH
n	⊐ンポーネント名、小文字	february, march
Nn	コンポーネント、頭文字が大文字	February, March

メモ N、n、とNn 識別子はエレポーネト M(月)のみをサポートします。

幅の修飾子は必要な場合最小の幅を示す小数が次に続くコンマにと、表示されます。オプションで、最大の幅を示す他の小数が次に表示されるダッシュ文字を追加することもできます。例:

- [D,2] はおで表示された先頭が0 で表示された月の日です。
- [MNn, 3-3] は Jan、 Feb、 Mar など3 文字で表示された月の名前です。

下のテーブルは書式のサンプルー部をリストしています

值	書式	結果
21-03-2002 16:21:12.492 GMT+02:00	[D]-[M]-[Y] [H]:[m]:[s].[f] [z]	2002-03- 21T16:21:12.492+02:00
315 2004 +01:00	[d] [Y] [Z]	2004-11-10T00:00:00+01:00
1.December.10 03:2:39 p.m. +01:00	[D].[MNn].[Y,2-2] [h]:[m]:[s] [P] [Z]	2010-12-01T15:2:39+01:00
20110620	[Y,4-4][M,2-2][D,2-2]	2011-06-20T00:0:00

サンプル

文字列の植 2019-12-24 19:43:04 +02:00 はオーマナマスク [Y]-[M]-[D] [H]:[m]:[s] [Z] を適用することにおり dateTime の等価に変換されます。



(XMLと名前宣言を除く)結果は以下の通りです:

```
<FlightInformation>
<FlightInfo departuredatetime="2019-12-24T19:43:04+02:00">
<Station airportcode="KIV"/>
</FlightInfo>
</FlightInformation>
```

7.10.2.9 parse-number

指定されたフォーマトに従い文字列を小数に変換します。



言語

Built-in、C++、C#、Java。

1 NJC - 5
· · · · · · · · · · · · · · · · · · ·

引数	説明
value	数値とて変換される文字列。
format	オプションの引数。数値が現在書式処理されている方法を識別する書式文字列書式文字列はformat-number 内で使用されたものと同じにないます。
	デフォノトは "#,##0.#" してなります。

decimal-point-character	オプションの別数。小数点文字として使用される文字を指定します。デフォルトは、'、文字です。
grouping-separator	オプンョンの引数。数値グループを分割するために使用されるセルーター/デミタを指定します。デンォルトは、、ベス字です。

サンプル

以下のマピングは書式マスク#,##0.00 を使用して文字列の値 "1,234.50" を小数の等価に解析します。このマピング内ではデ フォルトの値が入力文字列の書式と一致するため decimal-point-character とgrouping-separator 引数を接続する必要は あません。



(XMLと名前宣言を除くマピングの結果は以下の通りです:

```
<Article>

<Number>1</Number>

<Name>Office chair</Name>

<SinglePrice>1234.5</SinglePrice>

</Article>
```

7.10.2.10 parse-time

文字列をxs:time 値に変換します。この関数はparse-dateTime 関数をベースコンたもので、その中でも日付コンポーネントが無視されます。

$f_{\rm x}$ parse-time	
l>value	
¢ format	result

言語 Built-in、C++、C#、Java。

引数	説明
value	必須の別数。変換される文字列の値を提供します。
format	必須の引数。フォーマナ文字列を提供します。この引数は <u>parse-dateTime</u> 関数内の format 引数と同じ方法で使用されます。

7.10.2.11 string

入力値を文字列に変換します。ノードのテキストコンテンンを抽出するために関数を使用することもできます。XML 複合型が入力ノードとなる場合、全ての子要素も単一の文字列として出力されます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
arg	必須の引数。変換される値を提供します。

7.10.3 core | file path functions (ファイルパス関数)

file path 関数によりマピング内で更に処理するためにフォルダー、ファイル名、および拡張子などのファイル スデータに直接アクセス、おこに操作することができます。これらの関数は MapForce にて使用可能な全ての言語で使用することができます。

7.10.3.1 get-fileext

ドット ""文字を含むファイル マの拡張子が削除されます。



言語

引数	説明
filepath	必須の引数。処理されるファイルマを提供します。

サンプル

"c:\ata\Sample.mfd"を引数とて提供すると結果は.mfd です。

7.10.3.2 get-folder

ファイル なから、スラッシュや シクスラッシュ(日本語環境では円マーク)を含むフォルダー名が返されます。

∫₈ get-folder ⊳filepath folder ⊳

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
filepath	必須の引数。処理されるファイルマを提供します。

サンプル

"c:\ata\ample.mfd"を引数とて提供すると結果はc:\data\です。

7.10.3.3 main-mfd-filepath

マピングデザインファイル(.mfd)のフルレマを返します。mfd ファイルか現在保存されている場合、空の文字列が返されます。

∫_x main-mfd-filepath filepath ▷

言語

 ${\sf Built-in,C++,C\#,Java,XQuery,XSLT\,1.0,XSLT\,2.0,XSLT\,3.0_{\circ}}$

7.10.3.4 mfd-filepath

関数がシインマンピング内で呼び出された場合、main-mfd-filepath関数と同一の値(メインマンピングが含まれている.mfd ファイルのフル パマンが返されます。mfd ファイルが現在保存されていない場合、空の文字列が返されます。.mfd ファイルによりインポート済みのユーザー 定義関数を呼び出すと、ユーザー定義関数の定義を含むインポート済みの.mfd ファイルのフルレマか返されます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

7.10.3.5 remove-fileext

ドット文字を含むファイルマの拡張子が削除されます。

∫₈ remove-fileext ▷ filepath result-filepath ▷

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
filepath	必須の引数。処理されるファイルマを提供します。	

サンプル

"c:\data\Sample.mfd"を引数とて提供すると結果はc:\data\Sample です。

7.10.3.6 remove-folder

ファイル なから、スラッシュやバックスラッシュ(日本語環境では円マーク)を含むフォルダー名が削除されます。

∫_∗ remove-folder ⊳filepath filename ⊳

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
filepath	必須の引数。処理されるファイル、文を提供します。	

サンプル

"c: ¥data ¥Sample.mfd" を引数とて提供すると結果は Sample.mfd です。

7.10.3.7 replace-fileext

filepath / デメーターには提供されたファイル Co拡張子を extension / デメーターへの接続には提供されたものと置き換えます。

f ₈ replace-fileext		
⊳filepath		recult files ath
extension		result-filepath L

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
filepath	必須の引数。処理されるファイルンを提供します。	
extension	必須の引数。使用する新規の拡張子を提供します。	

サンプル

"c: \data \Sample.log"をfilepath 出て、".txt"をextension 出て提供すると結果はc:\data\Sample.txt になます。

7.10.3.8 resolve-filepath

ベースフォルダーイン対して相対的なファイル やを解決します。関数は、、、、現在のディレクトリと、、、、親ディレクトリ をサポートします。

	f ₈ resolve-filepath		
0	>ba	sefolder	recult filemath
0	⊳filepath		result-filepath L

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
basefolder	必須の引数。ノマカ解決される相対的なベースディレクトリを提供します。これは絶対的ませま 相対的なノマであることができます。	
filepath	必須の引数。解決される相対的なファイルレマを提供します。	

サンプル

下のマピングでは相対的なファイル マ... route.gpx は C: \data ディレトリマ北て解決されます。



マピングの結果はC:\route.gpx です。

7.10.4 core | generator functions (ジェネレーター関数)

core / generator 関数ライブラルコは植を生成する関数か含まれています。

7.10.4.1 auto-number

(例えば、1,2,3,4, ... などの)シーケンスで整数を生成します。 インクリメント 値である最初の整数と他のオプションをパラメーターを使用して設定することが可能です。

${ m f}_{ m s}$ auto-number	
>global-id	
start-with	
increment	result
restart-on-change	

生成されたマンピングコードによい呼び出される関数の順序は定義されていません。MapForce ではキャッシュされた計算結果を再利用するか、任意の順序で条件式を再評価する必要があります。同じ入力パラメーターを使用して複数回呼び出されると他の関数と異なりauto-number 関数は異なる結果を返します。このためauto-number 関数は注意して使用することが奨励されます。一部の場合、同じ結果を達成するためにposition 関数を代わりに使用することができます。

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
global-id	任意の、ラメーターです。マンピングデザインか複数のauto-number 関数を含む場合複製 (重複する) 番号と共にシーケンスを生成します。全てのauto-number 関数が決互しを考 慮するように、重複しないシーケンスを生成するには、各 auto-number 関数の例えば、 定数などの global-id 入力に共通の文字列を接続します。	
start-with	任意の デメーターです。生成されナシーケンスが開始する整数を指定します。 デフォルト の値 は1 です。	
increment	任意の ちメーターです。 インクリメント 値を指定します。 デフォルト の値は1 です。	
restart-on-change	任意の ・ ラメーターです。 接続されているアイテムのコンテン が変更されると、 start-with への カウンターを ルセナします。	

サンプル

以下のマピングはParentContext.mfd マピングの リエーションで、サンプル 親コンテキストの変更 で詳細に説明されています。

下て説明されているマピングの目的はノースXML ファイル内でそれそれの部署にコンずン複数のXML ファイルを生成することです。同じ 名前を持つ部署が存在します(これはそれぞれが異なる親部署に属するためです)。このために、生成されたそれぞれのファイル名は連番で 始まる必要があります、例えば、1-Administration.xml、2-Marketing.xml などです。



マピングの目的を達成するコよauto-number 関数が使用されます。この関数の結果はダッシュ文字の後に部署名および".xml"文字 列か表示されている形で連結されています。重要な点は concat 関数の3番目の ラメーター(部署名)に priority context か適用さ れていることです。これコよauto-number 関数が各部署のコンテキスト内で呼び出され、必要とされる連番の値を生成する効果がありま す。優先コンテキストが使用されない場合 auto-number 関数が、コンテキストが存在したい場合)番号 1を生成し続け、複製ファイル 名か結果として生成されます。

7.10.5 core | logical functions (論理関数)

logical 関数は(通常) 入力されたデータを比較して ブール値 [true] おけま [false] を返すために使用されます。通常、フルターを使用 してサブセットを度す前段階で、データを評価するために使用されます。 ほぼすべての理論関数には以下の構造が存在します:

入力/ デメーター= a | b おけはvalue1 | value2 出力/ デメーター= 結果

評価結果は、入力の値と比較に使用されたデータ型によと異ないます。例:整数の値 4 と12 の[より小さい]比較は、4 が12 より小さい ため、ブール値 [true] を返します。2 つの入カパラメーターが文字列の値 [4] と[12] を含む場合、 [4] はアルファベナの順序でオペ ラボ (12) の最初の文字 [1] よりた大きいか、構文的分析はの出力値 "false" を返します。

全ての入力値が同じデータ型の場合、比較は共通の型のために行われます。入力値が異なる型の場合、(例整数と文字列ませま文字 列と日付)そして、使用されるデータ型は、2つの入力型のうち最も一般的な(最も制限のよ)、入力データ型です。

具体的には、異なる型の2つの値を比較する前に、すべての入力の値は、共通のデータ型に変換されます。前のサンプルで説明すると、デー タ型 文字列 は整数よ制限が少ない ことがわかます。 整数の値 4 を文字列 [12] と比較し、 整数の値 4 を文字列 [4] に変換し、 文字列 [12] と比較します。

メモ Logical 関数をnull 値の存在をテストするために使用することはできません。null 値をlogical 関数の引数とて与えると null 値が返されます。null 値の扱い方に関しての詳しい情報は、<u>Nil 値 / Nillable</u> を参照してがさい。

7.10.5.1 equal

a がb と等価の場合、ブール値 true か返され、それ以外の場合 false どれます。

	$\mathbf{f}_{\mathbf{x}}$	equal	
ç	>a	recult [l
Ç	>b	result L	ſ

言語

 $\mathsf{Built-in},\mathsf{C++},\mathsf{C\#},\mathsf{Java},\mathsf{XQuery},\mathsf{XSLT}\ 1.0,\mathsf{XSLT}\ 2.0,\mathsf{XSLT}\ 3.0_\circ$

パラメーター

引数	説明	
а	必須の「ラメーターです。比較する最初の値を提供します。	
b	必須の ラメーターです。比較する2番目の値を提供します。	

7.10.5.2 equal-or-greater

a がb と等価おけまたき、場合、ブール値 true が返され、それ以外の場合 false どれます。

	$\mathbf{f}_{\mathbf{x}}$	equa	l-or-greater	
Ç	>a			
Ç	>b		result	

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

引数	説明	
а	必須の「ラメーターです。比較する最初の値を提供します。	
b	必須の ᡨメーターです。比較する2番目の値を提供します。	

780 関数

7.10.5.3 equal-or-less

a がb と等価おられる、ブール値 true が返され、それ以外の場合 false どいます。

	$\mathbf{f}_{\mathbf{x}}$	equ	ual-or-less	
C	>a		racult N	
C	>b		result	

言語

 $\mathsf{Built-in},\mathsf{C++},\mathsf{C\#},\mathsf{Java},\mathsf{XQuery},\mathsf{XSLT}\ 1.0,\mathsf{XSLT}\ 2.0,\mathsf{XSLT}\ 3.0_\circ$

パラメーター

引数	説明
а	必須の「ラメーターです。比較する最初の値を提供します。
b	必須の ラメーターです。比較する2番目の値を提供します。

7.10.5.4 greater

a がb よけき 場合、ブール値 true が返され、それ以外の場合 false どいます。

	$\mathbf{f}_{\mathbf{x}}$	greater	
Ç	>a	recult [ļ
Ç	>b	result L	ſ

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

引数	説明
а	必須の「ラメーターです。比較する最初の値を提供します。
b	必須の ᡨメーターです。比較する2番目の値を提供します。

7.10.5.5 less

a がb より小さ、場合、ブール値 true が返され、それ以外の場合 false どなます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
а	必須の「ラメーターです。比較する最初の値を提供します。
b	必須の ラメーターです。比較する2番目の値を提供します。

7.10.5.6 logical-and

各入力値がtrue の場合のみブール値 true を返し、それ以外の場合はfalse を返します。結果を他のlogical-and 関数に接続 することができ、このために全てがtrue を返すようにテストするために、論理 AND と任意の数の条件をジョインすることができます。また、こ の関数は追加の引数を取るように拡張することができます。<mark>関数の引数の追加ませば削除</mark>を参照してください。

f _x logic	al-and
kalue1	rocult C
value2	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

引数	説明
value1	必須の「ラメーターです。比較する最初の値を提供します。
value2	必須の ラメーターです。比較する2番目の値を提供します。

サンプル

下で説明されているマピングはlogical-and への全ての入力値がtrue であるためtrue を返します。入力値の、ずれかがfalseの場合、マピングの結果はfalse になります。



サノプルルックアップと連結を参照してくたさい。

7.10.5.7 logical-not

論理状態/結果を反転します。例えば 値 がtrue の場合、関数の結果はfalse です。 値 がfalse の場合、関数の結果はtrue です。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value	必須の、ウメーターです。入力、ウメーターを提供します。

7.10.5.8 logical-or

両方の入力値をブール型に揃える必要があります。少なくとも入力値の1つがtrueの場合、結果はtrueにないます。それ以外の場合結果はfalseにないます。

この関数は追加の引数を取るように拡張することができます。
関数の引数の追加ませる
削除を参照してくたさい。

_f _≈ logi	cal-or
2value1	racult P
value2	result L

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value1	必須の ウメーターです。比較する最初の値を提供します。
value2	必須の ちメーターです。比較する2番目の値を提供します。

サンプル

少などと1つの関数の引数がtrueのため、下のマピングの結果はtrueです。



7.10.5.9 not-equal

a がb と等価ではは、場合、ブール値 true が返され、それ以外の場合 false どいます。

	$f_{\!\!8}$ not-equal			
C	>a		recult [ĺ
Ç	>b		result L	ſ

言語

引数	説明
а	必須の ちメーターです。比較する最初の値を提供します。
b	必須の ラメーターです。比較する2番目の値を提供します。

7.10.6 core | math functions (数学関数)

データを使す。初歩的な計算を行うために数学関数を利用することができます。これに関数を使用して期間やdatetime(日時)の計算はできないという点に注意してくたさい。

はますべての数学関数は数学オペーションのオペランドである2つの入力パラメーター(value1、value2)を取ります。更に処理するため に入力値はdecimal へ自動的に変換され、処理が行われます。数学関数の結果もdecimal型です。



上に示されるサンプルでは、ターゲナコンポーネントにマンピングされた各製品の価格に対して20%の消費税が上乗せされます。

7.10.6.1 add

value1 をvalue2 に追加して結果を小数の値で返します。この関数は追加の引数を取るように拡張することができます。<u>関数の引数の</u> 追加または削除を参照してくたさい。

f_{8} add	
kalue1	
value2	result

言語

引数	説明	
value1	必須の ᡨメーターです。最初のオペランドを提供します。	
value2 必須の ラメーターです。2番目のオペランドを提供します。		

7.10.6.2 ceiling

value よけき、およ 等しい一番小さ、整数を返します。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value	必須の、ラメーターです。関数の入力、ラメーターを提供します。

サンプル

入力値は11.2 で ceiling 関数を適用して結果を11.2 以大きい番小さ、整数である12 します。

7.10.6.3 divide

value1 をvalue2 で割り、結果を小数として返します。除算結果の精度はケーゲト言語により変化します。round-precision 関数を使用することで、結果の精度を定義することができます。

	f_8 divide		
0	>value1	recult [
0	>value2	result	2

言語

引数	説明	
value1	必須の、ラメーターです。最初のオペランドを提供します。	
value2 必須の ラメーターです。2番目のオペライを提供します。		

7.10.6.4 floor

value より小さい、おけよ 等しい一番大き、整数を返します。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value	必須の ҕメーターです。関数の入力ノ ҕメーターを提供します。

サンプル

入力値は11.7 でfloor 関数を適用して結果を11.7 より小さい番大き、整数である11 します。

7.10.6.5 modulus

value1をvalue2で割った値の余いが返されます。

	f ₈ modulus		
Ç	>value1	rocult (L
Ç	>value2	result L	ſ

言語

 ${\sf Built-in,C++,C\#,Java,XQuery,XSLT\,1.0,XSLT\,2.0,XSLT\,3.0_{\circ}}$

引数	説明	
value1	必須の、ラメーターです。最初のオペランドを提供します。	
value2 必須の デメーターです。2番目のオペランドを提供します。		

サンプル

入力値が1.5 と1 の場合、modulus 関数の結果は0.5 です。説明は1.5 / 1 はあまとして0.5 を返します。

入力値が9と3の場合9/3か結果のため、余りばぶ結果は0です。

7.10.6.6 multiply

value1 をvalue2 と乗算し結果を小数とて返します。

	∫ ₈ mult	iply
¢	value1	
þ	value2	result
. 0		

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明	
value1	必須の┍━メーターです。最初のオペランドを提供します。	
value2	必須の ҕメーターです。2番目のオペライドを提供します。	

7.10.6.7 round

一番近い整数に値を丸め処理します。値か2つの整数の中央にある場合、「正の無限大に向けて半分に丸める」アルゴリズムが使用されます。例えば、値 [10.5] は [11] に丸められ端数処理され、値 [-10.5] はこれめられ [-10] に満数処理されます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value	必須の、ラメーターです。関数の入力、ラメーターを提供します。

7.10.6.8 round-precision

N が小数の引数である箇所で、入力値をN 小数の桁に丸め端数処理します。

	f ₈ round-precision	
ç	>value	reault C
ç	> decimals	result

言語

Built-in、C++、C#、Java。

パラメーター

引数	説明
value	必須の、ラメーターです。関数の入力、ラメーターを提供します。
decimals	必須の ちメーターです。丸めて端数処理する小数の桁を指定します。

サンプル

値 2.777777 を2 桁の小数に丸めると2.78 ばいます。値 0.1234 を3 桁の小数に丸めると0.123 ばいます。

7.10.6.9 subtract

value2 をvalue1 から減算し、結果を小数値として返します。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
value1	必須の「ラメーターです。最初のオペランドを提供します。
value2	必須の ちメーターです。2番目のオペランドを提供します。

7.10.7 core | node functions (ノード関数)

core | node functions ライブラカらの関数は名前お日お主釈などのマピングエンポーネント上のノードに関する情報についてのアクセスお日本illable 要素の処理を許可します。Nil 値 / Nillable も参照してくたさい。

ノード関数を必要とせずコノード名にアクセスする代替法が存在することに注意してくたさい。ノード名のマノピングを参照してくたさい。

下で表示されているマピングはノースXML ファイルのmsg:InterchangeHeader ノードから情報を取得するノード関数の、くつかを表示しています。具体的には、以下の情報が抽出されます:

- 1. node-name 関数はノードプレフィックスを含むノードの修飾名を返します。
- 2. local-name 関数はローカルのパトのみを返します。
- 3. static-node-name 関数はnode-name 関数に類似したものを返します。XSLT 1.0 でも使用することができます。
- 4. static-node-annotation 関数はスキーマ内で定義された通い要素の注釈を取得します。



(XML と名前宣言を除く)マピングの出力は以下の通りです:

```
<row>
        <coll>msg:InterchangeHeader</coll>
        <col2>InterchangeHeader</col2>
        <col3>msg:InterchangeHeader</col3>
        <col4>Interchange header</col4>
        </row>
```

7.10.7.1 is-xsi-nil

element 要素のxsi:nil 属性が[true] に設定されている場合 [true] を返します。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
element	必須の、ラメーターです。チェックされるソースノードに接続されている必要があります。

サンプル

下に表示されるマピングデザインはデータをソースからターゲナ XML ファイルに条件付きでピーしis-xsi-nil を含む複数の関数の使用法を表示しています。このマピングは HandlingXsiNil.mfd と呼ばれ <マイドキュメント >\Altova\MapForce2021\MapForceExamples\ ディレクトリ内で見つけることができます。



上て説明されるis-xsi-nil 関数はxsi:nil 属性がノースファイル内のstate アイテムのために[true] かどうかをチェックします。この属性が[false] の場合フィルターは親 Address 要素をターゲオーによす。(XML と名前宣言を除く)ソース XML ファイルは以下の通りです:

```
<BranchOffices>
   <Name>Nanonull</Name>
   <Office>
      <Name>Nanonull Research Outpost</Name>
      <EMail>sp@nanonull.com</EMail>
      <Fax xsi:nil="true"/>
      <Phone>+8817 3141 5926</Phone>
      <Address>
         <city>South Pole</city>
         <state xsi:nil="true"/>
         <street xsi:nil="true"/>
         <zip xsi:nil="true"/>
      </Address>
      <Contact>
         <first>Scott</first>
         <last>Amundsen</last>
      </Contact>
   </Office>
</BranchOffices>
```

ソース内にAddress が一つのみ存在し、xsi:nil 属性がstate 要素のために[true] に設定されているために、にマピングの結果 はAddress はターゲルにエピーされません。この結果、マピングの出力は以下の通りです:

7.10.7.2 local-name

ノードのコーカル名を返します。node-name 関数とは異なりlocal-name はノードのプレフィックスを返しません。ノードがプレフィックスを 持たない場合 local-name とnode-name は同じ値を返します。

```
∫<sub>8</sub> local-name
⊳node name⊳
```

言語

引数	説明
node	必須の「ラメーターです。この入力をノード名を取得するノードに接続してくたさい。

7.10.7.3 node-name

接続されているノードの修飾名(QName)を返します。ノードがXML text() ノードの場合、空のQname が返されます。この関数は名前付きのノードに対してのみ使用することができます。(fn:node-name を呼び出す)XSLT 2.0をターゲル言語としている場合、関数 は名前を持っていないノードに対して空のシーケンスを返します。

メモ ノード名の取得はノード、データベーステーブル、おけはフィールド、XBRL、Excel、JSON おけはプロトコールレッファーフィールド はファイル入力]フィールドのけがははサポートされていません。

<mark>∫₈ node-name</mark> ⊳node name⊳

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
node	必須の、ラメーターです。この入力をノード名を取得するノードに接続してくたさい。

7.10.7.4 set-xsi-nil

ターゲオノードをxsi:nil にセオします。

∫₈ set-xsi-nil result⊅

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

7.10.7.5 static-node-annotation

接続されたノードの注釈と共に文字列を返します。入力は以下である必要があります:(i)ソースコンポーネ・トノード、または(ii)呼び出しマピング内のノードに直接接続されている<u>パラメーター</u>に直接接続されている型 <u>インラインのユーザー</u>定義関数。
接続は直接的である必要があります。フィルター、おけは標準(非[インライン])ユーザー定義関数をパマスルーすることはできません。これは、 接続されたノードから取得されたテキストの生成時に置き換えられた擬似関数で、全ての言語で使用することができます。

	$f_{\!8}$ static-node-annotation		
ç	>no	de	name 🗘

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
node	必須の、ラメーターです。この入力を注釈を取得するノードに接続してくたさい。

7.10.7.6 static-node-name

接続されたノードの名前と共に文字列を返します。入力は以下である必要があります:(i)ソースエンポーネントノード、まけよ(ii)呼び出し マパング内のノードに直接接続されている<u>パラメーター</u>に直接接続されている型 <u>インライン</u>のユーザー定義関数。

接続は直接的である必要があります。フィルター、ませま非インラインユーザー定義関数をパススルーすることはできません。これは、接続され たノードから取得されたテキストの生成時に置き換えられた擬似関数で、全ての言語で使用することができます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
node	必須の・ラメーターです。この入力をノード名を取得するノードに接続してくたさい。

7.10.7.7 substitute-missing-with-xsi-nil

単純型コンテンを持つノードでは、この関数はノースコンポーネントの不足する値(おけよNULL値)をターゲトノード内でxsi:nil属 性と置き換えます。 ∫x substitute-missing-with-xsi-nil ⊳input result ⊳

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

引数	説明
input	必須の ちメーターです。この入力をノード名を取得するノードに接続してくたさい。

7.10.8 core | QName functions (QName 関数)

QName 関数は XML ドキュメント内の修飾名 (QName) を操作する方法を提供します。

7.10.8.1 QName

名前空間 URI とコーカルドトから QName を構築します。 Qname をターゲナオンポーネント内で作成するためこの関数を使用します。 定数関数により uri と node-name ノラメーターを提供することができます。

	$f_{\! 8}$ QName	
ç	> uri	recult [
Ç	>node-name	result

言語

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	説明
uri	必須。URI を提供します。
node-name	必須。ノードの名前を提供します。

7.10.8.2 local-name-from-QName

型 xs: QName の値からローカル名の部分が抽出されます。 node のローカル名を返す <u>local-name</u> 関数とは異なり、この関数は **qname** 入力に接続されているアイテムの content を処理します。

f _x local-name-from-QName		from-QName	
qname		name	result 🗘

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
qname	必須。型 xs:QName の関数入力値を提供します。

7.10.8.3 namespace-uri-from-QName

引数として与えられている Qname 値の名前空間 URI 部分が返されます。

f ₈ namespace-u	a namespace-uri-from-QName	
qname	result 🗘	

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
qname	必須。関数の入力・ラメーターを提供します。

サンプル

次のXML ファイルコはQName 値 o:name かきまれています。プレフィックス[o] は名前空間 http://NamespaceTest.com/Order にマップされています。

QName 値を処理し、名前空間 URI を取得するマピングは下で表示されています:



このマピングの出力はhttp://NamespaceTest.com/Order です。

7.10.9 core | sequence functions (シーケンス関数)

シーケンス関数は入力シーケンスの処理とそのエレテンソのグループ化を許可します。

7.10.9.1 distinct-values

values 入力に接続されている値のシーケンスを処理し、シーケンスとして一意の値のみを返します。これはシーケンスにて重複した値を取り除き、ユニーケムアイテムをターゲトコンポーネントヘニピーする際に役に立ちます。

∫_x distinct-values ▷values results ▷

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
values	この入力はゼロおけま他の値を提供するマンピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィールド、データベースレコードなど、から始まっている場合があります。

サンプル

次のXML ファイルコは架空の会社の社員に関する情報が含まれています。社員の一部は同じロールを有しているため、「ロール」属性ロール には複数の値が含まれています。例えば[Loby Matise] と[Susi Sanna] は同じロール[Support] を持っています。

```
<?xml version="1.0" encoding="UTF-8"?>
<KeyValueList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="KeyValueList.xsd">
<Item>
<Property Key="role">Manager</Property>
<Property Key="role">Manager</Property>
<Property Key="First">Vernon</Property>
<Property Key="Last">Callaby</Property>
```

<item></item>	
<property< th=""><th>Key="role">Programmer</th></property<>	Key="role">Programmer
<property< th=""><th>Key="First">Frank</th></property<>	Key="First">Frank
< Property	Key="Last">Further
<item></item>	
<property< th=""><th>Key="role">Support</th></property<>	Key="role">Support
< Property	Key="First">Loby
< Property	Key="Last">Matise
<item></item>	
<property< th=""><th>Key="role">Support</th></property<>	Key="role">Support
<property< th=""><th>Key="First">Susi</th></property<>	Key="First">Susi
<property< th=""><th>Key="Last">Sanna</th></property<>	Key="Last">Sanna

このXML ファイル内で発生するすべての一意のロール名のノストを抽出する必要があると仮定します。これは以下のマピング内で達成することができます。



上記のマピングでは欠か発生します

- ソースXML ファイルからの各 Property 要素はフィルターによい処理されます。
- フィルターのbool 入力への接続はKey 属性が[role] に等しい個所のみで Property 要素がターゲトコンポーネントに提供されることを保証します。文字列 [role] は定数により与えられます。(フィルターの条件を満たす [Support] プロ・ティかどつ存在するかかコフィルターの出力はおこの時点では複製を作成することに注意してくたさい。
- フィルターによれたたわるシーケンスは複製値を除外するdistinct-values 関数により処理されます。

この結果(XMLと名前宣言を除くマピングの出力は以下の通りです:

```
<items>
<item>Manager</item>
<item>Programmer</item>
<item>Support</item>
</items>
```

7.10.9.2 exists

接続済みのノードか存在する場合はtrueを返します。それ以外の場合はfalseを返します。ブール値を返すためこの関数は子要素または属性を持つ(または持たない)レコードのみをフィルターするために通常フィルターと共に使用されます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
node	存在を確認するノード。

サンプル

次のマピングはexists 関数を使用してデーダをフィルターする過程を表示しています。このマピングは PersonListsForAllBranchOffices.mfd と呼ばれくマイドキュメント>¥AltovaWapForce2021WapForceExamples¥ ディレクトリ内で見つけることができます。



PersonListsForAllBranchOffices.mfd

ソースファイル BranchOffices.xml 内はおつのOffice 要素が存在します。オフィスの一つにContact 子要素が存在しないことに気が対きます。マピングのゴールは複数存在します:

- a) 各オフィスのために、そのオフィス内に存在する連絡先のリストを抽出する
- b)各オフィスのためコオフィスと同じ名前を持つ個別のXMLファイルを作成する
- c) オフィスに連絡先が存在しない場合 XML ファイルを生成しない

この目的を達成するかかに、フィルターがマナングに追加されました。Contact アイテムかりなくとも1つか存在する箇所でフィルターは Office アイテムをターゲナトに、マレます。ブール値条件はexists 関数により提供されます。関数の結果がtrue の場合、ターゲナファ イル名を生成するためにオフィスの名前は文字列.xml と連結されます。マナングからのファイル名の生成の詳細に関しては、複数の入力 封まは出力ファイルを動的に処理を参照してくたさい。

もう一つの例は次のマピングです、くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\HasMarketingExpenses.mfd。expense-item がノースXML にて存在する場合、ターゲナ XML/スキーマファイルのhasExpenses 属性がtrue どいます。



HasMarketingExpenses.mfd

例ノードが存在したい場合の例外を参照してくたさい。

7.10.9.3 first-items

Nがcount パラメーターには提供される個所で、入力シーケンスの最初のN個のアイテムを返します。

f ₈ first-items		
nodes/rows		
count		results

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロまけ」は他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
count	任意の デメーターです。 入力シーケンスから抽出されるアイテムの数量を指定します。 デンォルト の値は1 です。

サンプル

次の模擬マピングは10の値のシーケンスを生成します。first-items 関数によりシーケンスは処理され、結果はターゲット XML ファイルに書き込まれます。



count 引数に値が存在しないため、1のデフォルトの値が適用されます。個の結果、シーケンスからの最初の値のみがマピング出力内に 生成されます:



更に現実的なサンプルは関しては、マンピングレンデメーターを提供する内のFindHighestTemperatures.mfdマンピングを参照して
くけさい。

7.10.9.4 generate-sequence

[from]と[to] パラメーターを境界として使用し、整数のシーケンスを作成します。

	f_8 generat	te-sequence	
ţ	>from	roculte D	
Ç	> <mark>to</mark>	results c	

言語

 ${\sf Built-in,C++,C\#,Java,XQuery,XSLT\,2.0,XSLT\,3.0_{\circ}}$

パラメーター

名前	説明
from	任意の デメーターです。シーケンスが開始する整数(下限の数量)を指定します。 デフォルト の 値は1 です。
宛先	必須の ちメーターです。シーケンスが終了する整数(上限の数量)を指定します。

7.10.9.5 group-adjacent

group-adjacent 関数は、key 入力に接続されているキーにより nodes/rows 入力に接続されているアイテムをグループ分けします。この関数は、アイテムが隣接してい場合、同じキー情報を共有するアイテムを個別のグループに分けます。複数の連続した(隣接した)アイテムが同じキーを共有する場合、これらのアイテムは同じグループに配置されます。

	f _x group-adj	jacent
ç	>nodes/rows	groups 🗘
ç	>key	key 🗘

例えば、下記の抽象的な変換に示されるように、グループキーはDepartment」です。このダイアグラムの左側は入力データを示し、右側は グループ分けの後の出力データを示しています。次は、変換の実行中に発生します:

- 最初は最初のキー「Administration」は新規のブループを作成します。
- 次のキーは異なるため、2番目のグループである「Marketing」が作成されます。
- 3番目のキーもおこ異なるための「Engineering」と呼ばれる他のグループが作成されます。
- 4番目のキーはる番目と同じキーです。このため、レコードは既存のグループに配置されます。
- 最後に、5番目のキーは番目のキーと異なるため、最後のブループを作成します。

下に説明されるとおり、「Michelle Butler」と「Fred Landis」は隣接する同じキーを持つかれて一緒にグループ分けされています。しかしなから、「Vernon Callaby」と「Frank Further」は同じキーを所有しているにもかかわらず隣接していないため、異なるグループに分けれています。

			Department	Name
			Administration	Vernon Callaby
Department	Name			
Administration	Vernon Callaby		Department	Name
Marketing	Susi Sanna		Marketing	Susi Sanna
Engineering	Michelle Butler	group-adjacent()		
Engineering	Fred Landis			
Administration	Frank Further		Department	Name
			Engineering	Michelle Butler
			Engineering	Fred Landis
			Department	Namo

Administration Frank Further

言語

Built-in C++ C#, Java XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	説明
nodes/rows	この入力はゼロませま地の値を提供するマメピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
key	アイテムをグループするキー

サンプル

使用するノースデータは次のコンテンンを持つXMLファイルです(下のコードリストでは名前空間とXML宣言は簡素化のために消除されて しることに注意してください。

<company></company>	
<person< th=""><th><pre>department="Administration" name="Vernon Callaby"/></pre></th></person<>	<pre>department="Administration" name="Vernon Callaby"/></pre>
<person< th=""><th><pre>department="Marketing" name="Susi Sanna"/></pre></th></person<>	<pre>department="Marketing" name="Susi Sanna"/></pre>
<person< th=""><th><pre>department="Engineering" name="Michelle Butler"/></pre></th></person<>	<pre>department="Engineering" name="Michelle Butler"/></pre>
<person< th=""><th><pre>department="Engineering" name="Fred Landis"/></pre></th></person<>	<pre>department="Engineering" name="Fred Landis"/></pre>
<person< th=""><th>department="Administration" name="Frank Further"/></th></person<>	department="Administration" name="Frank Further"/>

ビジネスの条件は、隣接する部署の個人レコードをグループ分けします。これを達成するコは、次のマンピングがgroup-adjacent 関数 を呼び出し、 department を key として提供します。



マパングの結果は以下のとおりです



```
<record key="Engineering" value="Fred Landis"/>
</group>
        <record key="Administration" value="Frank Further"/>
        </group>
</groups>
```

他のグループサンプルと共に、このサンプルは、次のマメングファイルの一部です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。出力 タブをクルクする前に、プレ ビューする関数に適用することのできる プレビュー ^{COD} ボタンをクルクします。

7.10.9.6 group-by

group-by 関数は指定するグループキーに従いレコードのグループを作成します。

f ₈ group-by	
nodes/rows	groups Þ
key	key Þ

例えば、下で示される抽象的な変換では、グループキーは、「Department」です。トータルで3つの一意の部署が存在するため、 group-by を適用すると3つのグループが作成されます:

			Department	Name
			Administration	Vernon Callaby
Department	Name		Administration	Frank Further
Administration	Vernon Callaby			
Marketing	Susi Sanna			
Engineering	Michelle Butler	group-by()	Department	Name
Engineering	Fred Landis		Marketing	Susi Sanna
Administration	Frank Further			
			-	
			Department	Name
			Engineering	Michelle Butler

Engineering

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

15メーター

名前	説明
nodes/rows	この入力はゼロまけ」は他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィールド、データベースレコードなど、から始まっている場合があります。

Fred Landis

名前	説明
key	アイテムをグループするキー

サンプル

使用するノースデータは次のエレテンを持つXMLファイルです(下のコードリストでは名前空間とXML宣言は簡素化のために削除されていることに注意してくたとい。

<company></company>	
<person< th=""><th><pre>department="Administration" name="Vernon Callaby"/></pre></th></person<>	<pre>department="Administration" name="Vernon Callaby"/></pre>
<person< th=""><th><pre>department="Marketing" name="Susi Sanna"/></pre></th></person<>	<pre>department="Marketing" name="Susi Sanna"/></pre>
<person< th=""><th><pre>department="Engineering" name="Michelle Butler"/></pre></th></person<>	<pre>department="Engineering" name="Michelle Butler"/></pre>
<person< th=""><th><pre>department="Engineering" name="Fred Landis"/></pre></th></person<>	<pre>department="Engineering" name="Fred Landis"/></pre>
<person< th=""><th><pre>department="Administration" name="Frank Further"/></pre></th></person<>	<pre>department="Administration" name="Frank Further"/></pre>

ビジネスの条件は、部署別に個人のレコードをグループ分けします。これを達成するコよ、次のマソピングがgroup-by 関数を呼び出し department を提供します。



マパングの結果は以下のとおりです

他のグループサンプルと共に、このサンプルは、次のマメピングファイルの一部です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。出力 タブをクルクする前に、プレ ビューする関数に適用することのできる プレビュー ¹⁰⁰ ボタンをクルクします。

7.10.9.7 group-ending-with

group-ending-with 関数はブール式の条件を引数として取ります。関数はブール式の条件がtrueの場合、条件を満たすレコードが 最後になるように、新規のブループが作成されます。下のサンプルでは、「Key」が「trailing」と等価であることが条件です。3番目と5番 目のレコードのために、この条件がtrueの場合、2つのブループが結果として作成されます。



メモ 条件を満たす最後のレコードの前にレコードが存在する場合、追加のグループが作成されます。例えば、「line」レコードが最後の「heading」レコードより比多く存在する場合、これらは新しレグループ内に配置されます。

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロまけ」は他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
bool	true の場合、新規のグループを開始するブール式を提供します。

サンプル

使用するノースデータは次のエレテンと持つXML ファイルです(下のコードリストでは名前空間とXML 宣言は簡素化のために削除されていることに注意してくたとい。

```
<records>
  <record key="line" value="A"/>
  <record key="line" value="B"/>
  <record key="trailing" value="Total 1"/>
  <record key="line" value="C"/>
  <record key="trailing" value="Total 2"/>
  </records>
```

ビジネスの条件は 各 [行末] レコードのナックコグループを作成することです。 各グループは [行末] レコードの前にある [line] レコードを含む 必要があります。 これを達成するコよ、次のマメピングがgroup-ending-with 関数を呼び出します。 下のマメピングでは key 名が [行末] に等し 場合、 bool に与えられている引数がtrue しなり、新規のグループが作成されます。



マピングの結果は以下のとおりです

他のグループサンプルと共に、このサンプルは、次のマメピングファイルの一部です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。出力 タブをクルクする前に、プレ ビューする関数に適用することのできる プレビュー 100 ポタンをクルクします。

7.10.9.8 group-into-blocks

group-into-blocks 関数は、N がblock-size 引数に与えられる値である、N アイテムを含む等価のグループを作成します。最後のグループは、ソース内のアイテムの数に従い、N つのアイテム、おけよ少なは数のアイテムを含むことができます。下のサンプルでは、 block-size は2 です。総数で5 つのアイテムが存在し、最後のアイテムを除き、それぞれのグループはつのアイテムを含んています。



言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロませま地の値を提供するマメピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
block-size	各グループのサイズを指定します。

サンプル

使用するノースデータは次のエレテンンを持つXMLファイルです(下のコードリストでは名前空間とXML 宣言は簡素化のために消除されていることに注意してくたさい。

<company></company>	
<person< th=""><th><pre>department="Administration" name="Vernon Callaby"/></pre></th></person<>	<pre>department="Administration" name="Vernon Callaby"/></pre>
<person< th=""><th><pre>department="Marketing" name="Susi Sanna"/></pre></th></person<>	<pre>department="Marketing" name="Susi Sanna"/></pre>
<person< th=""><th><pre>department="Engineering" name="Michelle Butler"/></pre></th></person<>	<pre>department="Engineering" name="Michelle Butler"/></pre>
<person< th=""><th><pre>department="Engineering" name="Fred Landis"/></pre></th></person<>	<pre>department="Engineering" name="Fred Landis"/></pre>
<person< th=""><th>department="Administration" name="Frank Further"/></th></person<>	department="Administration" name="Frank Further"/>

個人のレコードを2つのアイテムを持つブロックルビグループ分けすることかビジネスの必要条件とします。これを達成するオーカン、以下のマッピングはgroup-into-blocks 関数を呼び出し block-size として整数 [2] を提供します。



マパングの結果は以下のとおりです

```
<groups>
<group>
<record key="Administration" value="Vernon Callaby"/>
<record key="Marketing" value="Susi Sanna"/>
</group>
<group>
<record key="Engineering" value="Michelle Butler"/>
<record key="Engineering" value="Fred Landis"/>
</group>
```



アイテムの総数(5)を2で割ることができないため、最後のグループコンのアイテムのみ含まれていることに注意してくたさい。

他のグループサンプルと共に、このサンプルは、次のマメングファイルの一部です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。出力 タブをクルクする前に、プレ ビューする関数に適用することのできる プレビュー ¹⁰⁰ ボタンをクルクします。

7.10.9.9 group-starting-with

group-starting-with 関数はブール式の条件を引数として取ります。関数はブール式の条件がtrue の場合、条件を満たすレコードから、新規のグループが作成されます。下のサンプルでは、「Key」が「heading」と等価であることが条件です。最初と4番目のレコードのために、この条件がtrueの場合、2つのグループが結果として作成されます。

			Кеу	Value
Кеу	Value		heading	Intro
heading	Intro		line	A
line	A		line	в
line	в	group-starting-with()		
heading	Body			
line	С		Кеу	Value
			heading	Body
			line	С

メモ 条件を満たす最初のレコードの前にしコードが存在する場合、追加のグループが作成されます。例えば、「line」レコードが最初の「heading」レコードより化多く存在する場合、これらは新しいグループ内に配置されます。

言語

Built-in、C++、C#、Java、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明	
nodes/rows	この入力はゼロまけ」地の値を提供するマンピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。	
bool	true の場合、新規のグループを開始するブール式を提供します。	

サンプル

使用するノースデータは次のエレテンを持つXML ファイルです(下のコードリストでは名前空間とXML 宣言は簡素化のために削除されていることに注意してくたさい)。

```
<records>

<record key="heading" value="Intro"/>

<record key="line" value="A"/>

<record key="line" value="B"/>

<record key="heading" value="Body"/>

<record key="line" value="C"/>

</records>
```

ビジネズの条件は、各 [heading] レコードのナガレープを作成することです。各グループは [heading] レコードの後にある [line] レコー ドを含んでいる必要があります。これを達成するコよ、次のマメングがgroup-starting-with 関数を呼び出します。 下のマメング では、key 名が 「heading」 に等し、場合、bool に与えられている引数が true しなり、新規のグループが作成されます。



マピングの結果は以下のとおりです

```
<groups>
    <groups>
        <record key="heading" value="Intro"/>
            <record key="line" value="A"/>
            <record key="line" value="B"/>
            </group>
            <record key="heading" value="Body"/>
            <record key="line" value="C"/>
            </group>
        </groups>
```

他のグループサンプルと共に、このサンプルは、次のマメピングファイルの一部です: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\GroupingFunctions.mfd。出力 タブをクルクする前に、プレ ビューする関数に適用することのできる プレビュー 🏧 ポタンをクルクします。

7.10.9.10 item-at

position 引数により提供されたポジンョンで引数として提供された nodes/rows のシーケンスからアイテムを返します。最初のアイテムはポジンョン1 にあります。

$f_{\rm x}$ item-at	
hodes/rows	rocult [
position	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロませま地の値を提供するマンピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
position	個の整数はアイテムのシーケンスからどのアイテムが返されるかを指定します。

サンプル

次の模擬マッピングは10の値のシーケンスを生成します。item-at 関数によりシーケンスは処理され、結果はターゲット XML ファイルに 書き込まれます。



position 引数が3 に設定されているため、シーケンスからの3番目の値がターゲットについたれます。そのため、(XML と名前宣言を除く)マッピングの出力は以下の通りです:



7.10.9.11 items-from-till

[from] と[till] パラメーターを境界として使用し、nodes/rows のシーケンスを返します。最初のアイテムはポジション1 にあります。

	f _x items-from-till				
Ç	>nodes/rows				
Ç	>from	results 🛱			
Ç	>till				

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロまけ」は他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
from	この整数は抽出されるアイテムの開始のポジンコンを指定します。
till	この整数は抽出されるアイテムまでのポジションを指定します。

サンプル

次の模擬マンピングは10の値のシーケンスを生成します。items-from-till 関数によりシーケンスは処理され、結果はターゲオ XML ファイルに書き込まれます。



from とtill 引数は3と5にそれそれ設定されているため、3から5からの値のサブセナのみがターゲナーにいてされます。そのため、(XMLと名前宣言を除くマルングの出力は以下の通りです:

<items></items>		
<item>3</item>		
<item>4</item>		
<item>5</item>		

7.10.9.12 last-items

N が count / デメーターにお提供される個所で、入力シーケンスの最後のN 個のアイテムを返します。最初のアイテムはポジュン[1] に あります。



言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロまけ」は他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。
count	任意の ちメーターです。 入力シーケンスから抽出されるアイテムの数量を指定します。 デフォルト の値は1 です。

サンプル

次の模擬マピングは10の値のシーケンスを生成します。last-items 関数によりシーケンスは処理され、結果はターゲット XML ファイ ルに書き込まれます。



カウト引数が3に設定されているため、シーケンスからの最後の3つの値がターゲットについてされます。そのため、(XML と名前宣言を除く) マピングの出力は以下の通りです:



7.10.9.13 not-exists

接続済みのノードカ存在する場合はfalse を返します。それ以外の場合はtrue を返します。この関数は<u>exists</u> 関数の逆ですが、使用 方法は同じです。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
node	不在を確認するノード。

7.10.9.14 position

現在処理されているアイテムのシーケンス内のアイテムのポジションを返します。これは、例えば、アイテムに自動的に付番する場合に使用することができます。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
node	この入力はゼロおけれ他の値を提供するマッピングアイテムからの接続を受け取る必要があります。例 えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場合があります。

サンプル

次のマピングではマピングにお生成されたデータ内で一意の識別地を生成するためにposition 関数が使用されています。このマピングゴお以下の なで見つけることができるマピングデザインファイルが存在します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\ContactsFromBranchOffices.mfd.



ContactsFromBranchOffices.mfd

上のマピングではノース XML ファイル内に3 つのブランチオフィスか含まれています。ブランチオフィスコは Contact 子アイテムの任意の番号を含む場合かあります。マピングの目的は以下の通りです:

- ソースXML ファイルからすべての Contact アイテムを抽出し、ターゲオ XML ファイルに書き込みます。
- 各連絡先に一意の識別番号か割り当てられる必要かあります(ターゲナ XML 内の ID アイテム)。
- 各連絡先のIDはXがフィス番号を識別し、Xが連絡先番号を識別するフォームcxx-yyyyyを取る必要があります。オフィス番号が文字以下の場合、左側はゼロで埋め込まれる必要があります。同様に、連絡先番号が5文字以下を取る場合、左側はゼロで埋め込まれる必要があります。この結果、最初のオフィスからの最初の連絡先の有効な識別番号はco1-00001のようにないます。

マピングの目的を達成するコはposition 関数を含むくつかのMapForce 関数が使用されます。上のposition 関数は各オフィスのポジョンを取得します。下の関数は各オフィスのコンテキストで各連絡先のポジョンを取得します。

position 関数を使用する場合、現在のマレングコンテキストを考慮してください。具体的には、マレングの実行中に、関数を介して間 接的に、最初のマレングコンテキストが接続されるソースアイテムのターゲナ・コンポーネントのルートアイテムから確率されます。このサンプル内 では上のposition 関数は全てのオフィスのシーケンスを処理し、シーケンス内の最初のオフィスに対応する値1を最初に生成しま す。下のposition 関数は1、2、3 などの)そのオフィスのコンテキストで連絡先のポジンコンス対応するシーケンシャル番号を生成しま す。次のオフィスが処理されるとこの、内部シーケンスが設定され(このナンス 再度1 から開始します)。前に記述された必要条件に従い pad-string-left 関数の両方はディングを生成された番号に適用します。(ターゲナ Contact へのノースからの親接続のため) Concat 関数は各連絡先のコンテキストで作動します全ての値をジョインし、各連絡先の一意の識別番号を返します。

上のマピングから生成有れた出力は以下に表示される通りです(しくつからのノコードは読みやすさのため、消除されています):

```
<Contacts>

<Contact>

<ID>CO1-00001</ID>

<First>Vernon</First>

<Last>Callaby</Last>

</Contact>

<ID>CO1-00002</ID>

<First>Frank</First>

<Last>Further</Last>

</Contact>

<!-- ...->

<Contact>

<ID>C02-00001</ID>

<First>Steve</First>
```

```
<Last>Meier</Last>
</Contact>
<Contact>
<ID>C02-00002</ID>
<First>Theo</First>
<Last>Bone</Last>
</Contact>
<!-- ...->
</Contacts>
```

<u>フィルター</u>の適用後にアイテムのドジョンを取得する必要かある場合かあます。フィルターコンポーネトはシーケンス関数ではお、サンクフィ ルターされたアイテムのドジュンを検索する関数と共にはposition 関数と共に*直接*使用することはできません非間接的にこれは変数 コンポーネトを追加することに、以同能にないます。例えば、下のマメピングは前のマメピングを簡素化したものです。マメピングデザインファイル は次の、次で検索することかできます: <マイドキュメント

>\Altova\MapForce2021\MapForceExamples\PositionInFilteredSequence.mfd.



MapForce 内の変数の結果は常にシーケンスです。このために上にマッピングではposition 関数は変数により作成されたシーケンスを 反復し、そのシーケンス内の各アイテムのポジェンを返します。サンプルノードのフィルターと番号付けで詳しく説明されています。

7.10.9.15 replicate-item

入力シーケンス内の全てのアイテムを count 引数内で指定された回数繰り返します。単一のアイテムを node/row 引数は装続すると、 関数は、 N が count 引数の値である箇所の N アイテムを返します。 アイテムのシーケンスを node/row 引数は接続すると、 1度につ のアイテムを処理し、 関数は、 シーケンス内のそれぞれのアイテムを count 回繰り返します。 例えば、 カウト が2 の場合、 シーケンス 1,2,3 は1,1,2,2,3,3 を生成します。 下のサンプルで説明されてい る通り異なる count 値を入力シーケンス内の各アイテムこ与える ことができます。

f _x replicate-item		
node/row	raculta b	
⊳ <mark>count</mark>	results	

言語

Built-in, C++, C#, Java, XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	説明
nodes/rows	この入力はゼロませま地の値を提供するマンピングアイテムからの接続を受け取る必要かあります。例 えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場 合かあります。
count	node/row は接続されている各アイテムませまシーケンスを複数する回数を指定します。

サンプル

以下の構造を持つノースXML ファイルが存在すると想定します:



replicate-item 関数を使用して各個人名を異なる回数ターゲオコンポーネント内で繰り返すことができます。これを達成するために 各個人のcount ノードを関数の replicate-item 関数のcount 入力に接続します。



出力は以下のようてなります

```
<TargetLists>
<TargetList>
<TargetString>Michelle</TargetString>
```

```
<TargetString>Michelle</TargetString>
</TargetList>
<TargetList>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
<TargetString>Ted</TargetString>
</TargetList>
<TargetList>
<TargetString>Ann</TargetString>
<TargetString>Ann</TargetString>
<TargetString>Ann</TargetString>
<TargetString>Ann</TargetString>
<TargetList>
</TargetList>
```

7.10.9.16 replicate-sequence

入力シーケンス内の全てのアイテムを count 引数内で指定された回数繰り返します。例えば、カウトが2の場合、シーケンス1,2,3 は1,2,3,1,2,3を生成します。

f ₈ replicate-sequence		
nodes/rows		
¢ count	results L	

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロませませの値を提供するマンピングアイテムからの接続を受け取る必要かあります。例 えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場 合かあります。
count	接続されているシーケンスを複数する回数を指定します。

サンプル

次の模擬マソピングはシーケンス1,2,3を生成します。replicate-sequence 関数によりシーケンスは処理され、結果はターゲオ XML ファイルに書き込まれます。



count 引数が2 に設定されているため、シーケンスは度複製されターゲオにい ひされます。そのため、(XML と名前宣言を除く)マピングの出力は以下の通りです:

<items></items>		
<item>1</item>		
<item>2</item>		
<item>3</item>		
<item>1</item>		
<item>2</item>		
<item>3</item>		

7.10.9.17 set-empty

空のシーケンスを返します。



言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

7.10.9.18 skip-first-items

count 引数によりN が提供される個所で入力シーケンスの最初のN 個のアイテムをスキップし、残りのシーケンスを返します。

	f ₈ skip-first-items		
nodes/rows			
ł	> count		

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
nodes/rows	この入力はゼロませませの値を提供するマンピングアイテムからの接続を受け取る必要かあります。例 えば、接続はノースXML アイテム、CSV フィーリド、データベースレコードなど、から始まっている場 合かあります。
count	オプションの引数。スキップするアイテムの数量を指定します。デフォルトの値は1です。

サンプル

次の模擬マピングボンーケンス1,2,3を生成します。skip-first-items 関数によりシーケンスは処理され、結果はターゲット XML ファイルに書き込まれます。



count 引数が2 に設定されているため、最初の2つのアイテムはスキップされ、残りのアイテムはターゲットに いたれます。そのため、(XML と名前宣言を除く、マッピングの出力は以下の通りです:



7.10.9.19 substitute-missing

この関数は<u>exists</u> と<u>if-else 条件</u>の組み合わせには構成されます。node 入力に接続されているアイテムが存在すると、コンテンソよターゲットにエピーされます。それ以外の場合、replace-with 入力に接続されているアイテムのエレテンソよターゲットにエピーされます。

	$f_{\rm x}$ substitute-missing		
Ç	>node		
Ç	replace-with result i		

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
node	この入力はゼロませ」されの値を提供するマノピングアイテムからの接続を受け取る必要があります。例

名前	説明
	えば、接続はノースXMLアイテム、CSV フィールド、データベースレコードなど、から始まっている場合があります。
replace-with	この入力は置換値を提供するマンピングアイテムから接続を受け取る必要があります。

7.10.10 core | string functions (文字列関数)

文字列関数はデータの一部分を取得、サブ文字列のテスト、文字列からの精報の取得、文字列の分割などを行うために文字列データを操作することができます。

7.10.10.1 char-from-code

引数とて提供されている小数 Unicode 値 (コード)の文字表示を返します。ヒント: 文字のUnicode 小数コードを検索するコよ code-from-char 関数を使用することができます。

∫_x char-from-code ▷ code char ▷

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
code	小数とてのUnicode 値。

サンプル

Unicode Web サイトで使用可能なチャートによると(<u>https://www.unicode.org/charts/</u>) 感嘆符文字は0021 の16 進数値を 持っています。小数フォーマナの対応する値は33 です。このため、引数として33 を提供すると、char-from-code 関数は! 文字を 返します。<u>特殊文字の置換</u>も参照してください。

7.10.10.2 code-from-char

文字列とて提供されている文字の小数 Unicode 値 (コード) が返されます。文字列とて提供されている文字列が複数の文字を持つ場合、最初の文字のコードが返されます。

f_8 code-	from-char
🗟 char	code 🖓

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
char	入力文字列の道。

サンプル

入力 char はる (ドルサイン) 文字です。 関数はこの文字のための小数 Unicode 値である) 36 を返します。

7.10.10.3 concat

2 個まけは以上の値を連結(追加)して、単一の結果文字列を生成します。全ての入力値は自動的に「文字列」型へ変換されます。デフォ ルではこの関数には2つの デメータのみ存在しますが、更に デメーターを追加することができます。パテメーターの追加(...) おけよパラ メーターの削除(図)をクリックしていデメーターを追加ませば削除します。関数の引数の追加、ませば削除も参照してくたとい。

_f ₈ cond	at
value1	rocult C
value2	result L

ᄯ concat 関数へのすべての入力は値を必要とます。入力の1つか値を持たない場合、関数が呼び出されず、エラーが発生しま す。空の文字列は有効な入力値ですが、(set-empty 関数の結果など)空のシーケンスは無効な値で、この結果、関数は 失敗します。これを回避するコよsubstitute-missing 関数を使用して値を最初に処理し concat 関数に結果を入力として 提供します。

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
value1	最初の入力値。
value2	2番目の入力値。
valueN	N の入力値。

サンプル

下で説明されるマピングでは、concat 関数は最初の名、定数 " "、そして姓を結合します。返される値は、FullName ターゲルアイ テムに書き込まれます。この関数のマンピングは次の、又で見つけることができます: くマイドキュメント

>\Altova\MapForce2021\MapForceExamples\HasMarketingExpenses.mfd。



HasMarketingExpenses.mfd

7.10.10.4 contains

引数の値として提供されている文字列の値に引数として提供されているサブ文字列が含まれている場合、ブール値 true を返します。

	$f_{\!\!8}$ contain	s
Ç	>value	
Ç	>substring	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
value	入力値(すなすち [haystack])。
substring	(すなわち [needle])を検索するサブ文字列。

サンプル

入力 value が[category] で substring が[cat] の場合、関数は true を返します。

7.10.10.5 normalize-space

正規化された入力文字列を返します。正規化は行頭および行末スペースが削除され、複数の連続した空白文字の各シーケンスが単一の 空白スペースと置き換えられていることを意味します。空白スペースのためのUnicode文字は(U+0020)です。



言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
string	正規化するための入力文字列。

サンプル

入力文字列が The quick brown fox の場合、関数はThe quick brown fox を返します。

7.10.10.6 starts-with

引数とて提供されているサブ文字列から開始する引数とて文字列が提供されている場合、ブール値 true を返します。それ以外の場合 false を返します。

	f_8 start	s-with	
ç	> string	recult P	l
ç	> substr	result L	

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
string	入力文字列。
substr	チェックするサブ文字列。

サンプル

入力 string が category で、substr が cat の場合、関数は true を返します。

7.10.10.7 string-length

引数として与えられている文字列内の文字の数量を返します。

∫₈ string-length ▷ string result ▷

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
string	入力文字列。

サンプル

入力文字列がcar の場合、関数は3 を返します。入力文字列が空白の場合、関数は0 を返します。

7.10.10.8 substring

start とlength パラメーターには指定されている文字列の部分を返します。

f _x substring	
string	
⊳ start	result 🛱
length	

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
string	入力文字列。

名前	説明
start	サブ文字列が抽出される開始位置(インデックス)を指定します。最初のインデックスは1です。
length	任意。抽出されるアイテムの数量を指定します。length パラメーターが指定されていない場合、結果はstartから開始する文字列の最後までのフラグメートです。

サンプル

入力文字列がMapForce の場合、開始は1 で、長さは3 で、関数はMap を返します。入力文字列がMapForce の場合、開始は 4 で、長さが提供されていない場合、関数はForce を返します。

7.10.10.9 substring-after

substr の最初の発生後に発生する文字列の部分を返します。substr がstring で発生しない場合、関数は空の文字列を返します。

	$f_{\!8}$ substring-after		
string		recult [
⊳substr		result	

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明	
string	入力文字列。	
substr	サブ文字列。substr の最初の発生の後の文字は関数の結果です。	

サンプル

入力文字列がMapForce で substr がMap の場合、関数は Force を返します。入力文字列が2020/01/04 で substr が/の場合、関数は 01/04 を返します。

7.10.10.10 substring-before

substr の最初の発生前に発生する文字列の部分を返します。substr がstring で発生しない場合、関数は空の文字列を返します。

	$f_{\rm x}$ substring-before		
string		rocult P	L
⊳ substr		result	ſ

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明	
string	入力文字列。	
substr	サブ文字列。substr の最初の発生の前の文字は関数の結果です。	

サンプル

入力文字列がMapForce で substr が Force の場合、関数は Map を返します。入力文字列が2020/01/04 で substr が/の場合、関数は 2020 を返します。

7.10.10.11 tokenize

引数として与えられた区切り文字を使用して入力文字列を文字列のシーケンスに分割します。

f ₈ tokenize	
l>input	
delimiter result	

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
input	入力文字列。
delimiter	使用する区切り文字。

サンプル

入力文字列がA,B,Cの場合、区切り文字が,の場合、関数は3つの文字列のシーケンスを返します A、B、およびC。



上の模擬マピングでは関数の結果は文字列のシーケンスです。一般的なマピングノレールによると、ソースシーケンス内の各アイテムのため に新規item がターゲトコンポーネント内に作成されます。この結果、マピングの出力は以下のようこなります。



更に詳細なサンプルは関してはくマイドキュメント> Altova MapForce2021 MapForceExamples ¥フォルダー内の tokenizeString1.mfd マメピングを参照してくたさい。



tokenizeString1.mfd

ソースXML ファイルは以下で表示されています。Tool 要素には2つの属性が存在します: Name と Code。 Tool 要素データはエンマで 区切られたテキストで構成されています。

```
<!-- ...->
</AltovaTools>
```

マピングは以下を行います

- Tokenize 関数はTool ソースアイテムからデータを受け取り、コンマ,区切り文字を使用してデータを個別のチャンカニ分割します。最初のチャンカは[XML editor] および登番目のチャンカは[XSLT editor] などごなります。
- tokenize 関数から結果する各チャンクのために、ターゲット内に新規の行か主成されます。これは関数の結果間の接続とター ゲットコンポーネント内のRows アイテムにより発生します。
- tokenize 関数の結果は各チャンクの先頭の空白文字を削除するleft-trim 関数にもマップされます。
- left-trim 関数の結果(各チャンク)はターゲオコンポーネオのFeature アイテムに書き込まれます。
- ターゲナコンポーネト出力ファイルはフィールド区切り文字がたコロンでCSV ファイル(AltovaToolFeatures.csv) とて 定義されています(設定を確認するナムリコンポーネトをダブルクトックしてくたさい)。

マピングの結果は、tokenize 関数により作成された各チャンクのためにターゲオ CSV ファイル内で新規の行が作成されます。マピングの出力のフラグメトは以下のようになります。

Tool;Feature XMLSpy;XML editor XMLSpy;XSLT editor XMLSpy;XSLT debugger XMLSpy;XQuery editor XMLSpy;XQuery debugger XMLSpy;XQUery debugger XMLSpy;XML Schema / DTD editor XMLSpy;WSDL editor XMLSpy;SOAP debugger MapForce;Data integration MapForce;tata integration MapForce;text conversion MapForce;text conversion MapForce;EDI translator MapForce;EXCel mapping

7.10.10.12 tokenize-by-length

入力文字列を文字列のシーケンスに分割します。結果する各文字列のサイズはlength パラメーターにより決定されます。

$f_{\rm 8}$ tokenize-by-length	
l>input	recutt f
length	result

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。
パラメーター

名前	説明	
input	入力文字列。	
length	文字列の生成されたシーケンス内の各文字列の長さを決定します。	

サンプル

入力文字列がABCDEF の場合、長さが2 の場合、関数は3つの文字列のシーケンスを返します: AB、CD、およびEF。



上の模擬マピングでは関数の結果は文字列のシーケンスです。一般的なマピングノレールによると、ソースシーケンス内の各アイテムのために新規itemがターゲトコンポーネト内に作成されます。この結果、マピングの出力は以下のようないます。



更に詳細なサンプルに関してはくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内の tokenize String 2.mfd マンングを参照してくたい。



tokenizeString2.mfd

前例で使用したものと同じ XML ソースファイルを以下に示します。MissionKit 要素には2つの属性が存在します: Edition と ToolCodes ですがMissionKit 要素コンテント ではかません。このサンプルニ 関連しない XML コンテンソ 排削除されていることご注意し てください。

xml version="1.0" encoding="UTF-8"?
<pre><altovatools <="" pre="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></altovatools></pre>
<pre>xsi:noNamespaceSchemaLocation="AltovaTools.xsd"></pre>
<version>2010</version>
<tool code="XS" name="XMLSpy"><!-- --></tool>
<tool code="MF" name="MapForce"><!-- --></tool>
<tool code="SV" name="StyleVision"><!-- --></tool>
<tool code="UM" name="UModel"><!-- --></tool>
<tool code="DS" name="DatabaseSpy"><!-- --></tool>
<tool code="DD" name="DiffDog"><!-- --></tool>
<tool code="SA" name="SchemaAgent"><!-- --></tool>
<tool code="SW" name="SemanticWorks"><!-- --></tool>
<tool code="AU" name="Authentic"><!-- --></tool>
<missionkit edition="Enterprise Software Architects" toolcodes="XSMFSVUMDSDDSASW"></missionkit>
<missionkit edition="Professional Software Architects" toolcodes="XSMFSVUMDS"></missionkit>
<missionkit edition="Enterprise XML Developers" toolcodes="XSMFSVDDSASW"></missionkit>
<missionkit edition="Professional XML Developers" toolcodes="XSMFSV"></missionkit>

どのAltova 製品が MissionKit の各エディションに含まれているかを表すリストを生成します。

マピングのしくみ

- Select Mission Kit 入力コンポーネントはマンピングへの、デメーターとしての役割を果たします。定数からデフォルトの値を受け取ります。この場合は[Enterprise XML Developers]です。
- Equal 関数はノースXML ファイルからのEdition アイテムと共にフラメータとして提供されているエディンヨンを比較し、結果を ToolCodes フィルターのbool ノラメーターイン マします。
- ToolCodes フィルターのnode/row 入力がノースコンポーネトのToolCodes アイテムから接続されています。
 [Enterprise XML Developers] エディションの値: xsmFsvDDsasw.
- XSMFSVDDSASW 値はon-true パラメーターン やされ、更にtokenize-by-length 関数のinput パラメーターン やされます。
- tokenize-by-length 関数は値 xSMFSVDDSASW をそれそれ2文字ずつ持つ複数のチャンカニ分割します。length / ラメーターは2です。このために6個のチャンクか結果として作成されます。
- 各チャンカは総合して9個のアイテムからなるソースファイルからの2-文字 Code 値に対して比較されます。比較の結果がシィ ルター(filter)のbool / ラメーターへ渡されます。tokenize-by-length 関数から生成された全てのの要素がシィルターの node/row / ラメーターへ渡されている点に注意してくたさい。
- exists 関数はフィルターコンポーネントのon-true パラメーターにより、ひされた既存/非既存ノードをチェックします。既存するノ ードとは ToolCodes チャンクとCode 値の間で一致するものです。非既存のノードはCode 値に一致するToolCodes チャンクが存在しないものです。
- exists 関数の各 bool 結果 はif-else コンポーネント に いされます。ノード か存在する場合、おけま [Y] を、ノード か存在 しない場合 [N] をターゲナト 内で生成する。

マピングの結果は以下のようてなります

```
Tool;MissionKit for Enterprise XML Developers
XMLSpy;Y
MapForce;Y
StyleVision;Y
UModel;N
DatabaseSpy;N
```

7.10.10.13 tokenize-regexp

入力文字列を文字列のシーケンスに分割します。引数として与えられている正規表現 pattern に一致するサブ文字列はセルーターを 定義します。一致する(セルーター)文字列は関数により返された結果に含まれていません。

メモ C++、C#、おゴま Java コードを生成する際には、正規表現構文の高度な機能が異なる可能性があります。詳細に関しては 各言語の正規表現ドキュメトを確認してくたさい。

f ₈ tokenize-regexp		
l>input		
¢ pattern	result 🖗	
flags		

言語

Built-in、C++、C#、Java、XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明	
input	入力文字列。	
pattern	正規表現パターンを提供します。パターンに一致するサブ文字列は区切り文字として扱われます。正 規表現に関する更に詳しい情報については正規表現のセクションを参照くたさい。	
flags	任意の ラメーターです。使用される正規表現 <u>flags</u> を提供します。例えば、フラグ "i" はマメピング に大文字と小文字を区別して処理するようこ命令します。	

サンプル

下で紹介されているマンピングのゴールは文字列 a 、 b c、 d を各アルファベト文字がシーケンス内のアイテムである個所で文字列のシーケンスに分割することです。 冗長空白スペースませましてを削除する必要があります。



このゴールを達成するコは正規表現パターン [,]+ がtokenize-regexp 関数の、ラメーターとして提供される必要がおります。このパターンは以下を意味します

- 文字クラス [,] 内の文字の、ずれかに一致します。このために、入力文字列内でエンマませるとつきか発生すると分割が行われます。
- 量指定子 + は一致する先頭の文字クラスの1つませば複数の発生を指定します。量指定子がない場合は文字列の結果シーケンス内に各空白文字ませましてか個別のアイテムを作成します。これば期待されない結果です。

マピングの出力は以下のようてなります。

7.10.10.14 translate

文字の置換による文字を実行します。string1内に含まれている文字のためにvalueを確認し string2内の同じポジョンにあるもののために各文字を置き換えます。string2に対応する文字が無い場合、その文字に削除されます。

	J _x translate		
ę	value		
ę	string1	result 🕻	>
ę	string2		

言語

Built-in、C++、C#、Java、XQuery、XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	説明
value	入力文字列。
string1	検索文字のノストを提供します。文字列内の各文字のポジュンは重要です。
string2	置換文字のJストを提供します。各置換文字のポジションはstring1内のものに対応する必要が みます。

サンプル

文字列 [12,3] を (12.3) に変換すると仮定します。具体的には、角かっこがれかっこと置き換えられる必要があり、コンマはピリドと置き換えられる必要があります。これを達成するには translate 関数を以下のように呼び出します:



上のマンピングでは、最初の定数は処理される入力文字列を提供します。2番目と3番目の定数はそれそれ string1 と string2 とて文字のノストを提供します。

		_
otring 1	г	1
Stringi	L /	/ L

string2 (.)

string1とstring2 には同じ数の文字が存在することに注意してくたき、。string1 内の各文字のために、同じポジェンのstring2 からの等価の文字が置換として使用されます。この結果、次の置換が発生します:

- 各[は(と置き換えられます
- 各,は.と置き換えられます
- 各]は]と置き換えられます

マピングの出力は以下のようないます

(12.3)

この関数は文字列から特定の文字を削除するために使用することができます。これを達成するこは string1 / ラメーターを削除する文字に 設定し、を空の文字列を string2 に設定します。例えば下のマメピングは文字列 38ab8a7a65xkh3 からすべての小数を削除します。



マピングの出力は以下のよう」ないます

abaaxkh

7.10.11 db

db ライブラルは、データベース内にNULL フィールドが発生すると、マパングの結果を定義することのできる関数を含みます。

7.10.11.1 is-not-null

フィールドがnull の場合 false を返します。それ以外の場合はtrue を返します。

	$\mathbf{f}_{\mathbf{x}}$	is-r	not-null
ç	>fie	ld	result 🕻

言語

Built-in、C++、C#、Java。

パラメーター

名前	説明
field	データベースフィールド

7.10.11.2 is-null

フィールドがnull の場合 true を返します。それ以外の場合はfalse を返します。

∫₈ is-null ⊳field result D

言語

Built-in、C++、C#、Java。

パラメーター

名前	説明
field	データベースフィールド

7.10.11.3 set-null

データベースフィールドをNULLをセオします。この関数はデフォルト値をNULLに上書きします。データベースフィールド以外のアイテムへ接続されている場合、空のシーケンスとして振る舞います。以下の点に注意してください

- 他の関数へset-nullを接続しても、通常その関数は呼ばれません。(例えばcountのような)シーケンス関数へsetnullを接続すると、空のシーケンスが呼び出しに使用されます。
- set-null をフィルターイは接続するとif-else 条件は予想通い作動します。フィールドはNULL に設定されます。フィルターイコは [node/row] 入力を意味します。

• set-null から simpleType への接続を行っても、ターゲトコンポーネト内に要素が作成されることはありません。



言語

Built-in、C++、C#、Java。

7.10.11.4 substitute-null

NULL でおよ場合フィールド自身を返します。それ以外の場合 replace-with が返されます。

f ₈ substitute-null	
⊳field	racult N
Preplace-wit	th

言語

Built-in、C++、C#、Java。

パラメーター

名前	説明
field	データベースフィールド
replace-with	代替の値。

サンプル

substitute-null 関数が実際に使用されている例が以下に示されています。このマピングはDB-ApplicationList.mfd と呼ばれくマイドキュメント> Altova Map Force 2021 Map Force Examples ¥フォルダー内で見つれてとかできます。



[Applications] テーブルを含むAccess データベースからのデータを読み取ります。

	AppID 🔹	AppName •	Description •	Category •	URL •
1	1	Altova MapForce	The premier data mapping tool.	IDE	www.altova.com/xmlspy
2	2	Notepad	[NULL]	[NULL]	[NULL]

最初の関数により、[Applications] テーブル内の Category フィールドが NULL であるかチェックされます。このフィールドは Notepad ア プルテーションのために NULL のため、代替の値 [Misc] がテキストファイルの Category アイテム ニマップされます。

2番目の関数は Description フィールドがNULL であるかチェックします。このフィールドは Notepad アプルケーションのために NULL のため、代替の値 [No description] かターゲットファイルの Description アイテム こマップされます。

7.10.12 lang | datetime functions (日付時刻関数)

lang ライブラルらの日付と時刻関数は日付、時刻、期間を操作するために使用することができます。core ライブラルらの日付と時刻関数は異なり、これらの関数はビルトイン、Java、C#、おさはC++ 言語を選択すると使用することができます。

7.10.12.1 age

引数として与えられた生年月日から今日まで経過した年数を数値で返します。

$f_{\scriptscriptstyle R}$ age	
birthdate	
now	result

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
birthdate	xs:date	必須。xs:date 値とて生年月日を提供します。
now	xs:date	任意の、ラメーターです。デフォルトは現在のシステムの日付です。値がnow 引数 にマップされると、関数は生年月日と現在の日付の差分を年数で返します。

7.10.12.2 convert-to-utc

UTC (世界協定時刻) に引数とて提供されている時刻の値を変換します。関数は(例えば、[+5:00] などの) タイムゾーンコンポーネントを考慮します。

<mark>∫₈ convert-to-utc</mark> ≥time time ⊃

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
time	xs:dateTime	変換するxs:dateTime 値を与えます。

サンプル

入力値が2001-12-17T09:30:02+05:00の場合、関数の結果は2001-12-17T04:30:02です。

入力値が2001-12-17T09:30:02Zの場合、関数の結果は2001-12-17T09:30:02です。この場合、行末[Z]が既にの時 刻を[UTC]と同じゼロ(おけばズールー)時刻に設定しているため変換は発生しません。

7.10.12.3 date-from-datetime

引数とて与えられたxs:dateTime 値から date の部分を返します。time 部分はゼロに設定されています。タイムゾーンは変更されていません。

	$\mathbf{f}_{\mathbf{x}}$	f ₈ date-from-datetime		
ç	datetime		date 🕻	2

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	処理するxs:dateTime 値を与えます。

サンプル

入力値が2001-12-17T09:30:02+05:00の場合、関数の結果は2001-12-17T00:00:00+05:00です。

7.10.12.4 datetime-add

期間(2番目の引数)をdatetime(最初の引数)に足して得られたxs:dateTime 値を返します。

$f_{\!\scriptscriptstyle 8}$ datetime-add	
datetime	recult [
duration	result L

言語

Built-in、C++、C#、Java。

名前	型	説明	
datetime	xs:dateTime	入力とて使用されるxs:dateTime 値を与えます。	
duration	xs:duration	xs:duration 値を提供します。	
		サンプルの期間はP1Y2M3DT04H05M59S:	
		 [P] か期間の指定子で、必須です; 残りの文字は以下の順序で表示されます: 1 年、2 カ月、3 日、T (時間指定子)、04 時間、05 分、59 秒。 	
		マイナス文字が[P] 指定子の前に表示されると、これはマイナスの期間を示しています。例 -P1D.	

サンプル

入力 datetime 値が2001-12-17T09:30:02+05:00 と想定します。duration がP10D (10 日)の場合関数の結果は 2001-12-27T00:00:00+05:00 しないます。

昨日の日付を取得するコはnow 関数を datetime 入力に接続します。下のマンピングでは期間 -p1D は[マイナス1日]を意味し、マッ ピングは昨日の日付を返します。



7.10.12.5 datetime-diff

datetime2(2番目の引数)をdatetime1(壱番目の引数)から引き算し得られた期間を返します。結果は文字列ませば期間データ型

	f _x datetime-diff		
ç	>datetime1		
Ę	>datetime2		result

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明	
datetime1	xs:dateTime	最初のxs:dateTime 値を提供します。	
datetime2	xs:duration	2番目のxs:dateTime 値を提供します。	

サンプル

下に表示されるマピングでは date time - diff 関数は飛行機の離陸日時 date time 2001-12-17T09:30:02+05:00 を到着 日時 2001-12-17T19:30:02+05:00 から l き算します。 到着日時は 離陸時間 よ 比大きな 値であり、 関数の最初の入力に 接続さ れています。



マピングの出力はつの値の差分です(10時間の期間):

PT10H

7.10.12.6 datetime-from-date-and-time

xs:date 値(最初の引数)とxs:time 値(2番目の引数)かビルドされたxs:dateTime 値を返します。結果は文字列型おけ xs:dateTime データ型へマルングすることができます。

	$f_{\!8}$ datetime-from-date-and-time		
Þ	da	tevalue	datatima ľ
Ş	timevalue		Gatetime

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明	
datevalue	xs:date	型 xs:date の値を提供します。	
timevalue	xs:time	型 xs:time の値を提供します。	

サンプル

最初の引数が2012-06-29 で2番目の引数が11:59:55 の場合、関数は2012-06-29T11:59:55 を返します。

7.10.12.7 datetime-from-parts

以下のいたを組み合わせて引数としてビルドされた型 xs:dateTime の値 年、月、時間、分、秒、ミルト、タイムゾーン。この関数は自動的に提供されたい デメーターを正規化します。例えば、1月32日は自動的に2月1日に変換されます。

f ₈ datetime-from-parts		
⊳year		
⊳month]	
🗘 day]	
hour	datatima	
minute		
second]	
millisecond]	
timezone		

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
year	xs:int	年の値を与えます。
month	xs:int	月の値を与えます。
day	xs:int	月の日付を与えます。
hour	xs:int	任意。時間の値を与えます。
minute	xs:int	任意。分の値を与えます。
second	xs:int	任意。秒の値を与えます。
millisecond	xs:decimal	任意。ミノ利の値を与えます。
timezone	xs:int	任意。タイムゾーンを分数で与えます。この値は負の値であることができます。

サンプル

以下のマピングはxs:dateTime 値を定数にお提供される部分から作成します。



マピングの出力は2020-04-17T08:58:54.333-01:00 です。

7.10.12.8 day-from-datetime

引数とて与えられたxs:dateTime 値が整数の値とて日数を返します。

∫<mark>x day-from-datetime</mark> ▷ datetime day ▷

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T10:30:03+01:00 の場合、関数は17 を返します。

7.10.12.9 day-from-duration

引数とて与えられたxs:duration 値が整数の値とて日数を返します。

	$f_{\!\scriptscriptstyle R}$ day-from-duration			
Ç	>dı	uration	day 🗘	

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP1Y2M3DT10H30M の場合 day-from-datetime 関数は3 を返します。

7.10.12.10 duration-add

2つの期間を足し合わせることで得られる期間を返します。

	$f_{\! 8}$ duration-add		
Ç	>duration1	a a cult f	
ç	>duration2	result	

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明	
duration1	xs:duration	型 xs:duration の入力値を提供します。	
duration2	xs:duration	型 xs:duration の2番目の入力値を提供します。	

サンプル

最初の期間は POYOM3DTO3HOM (3日 2時間) で2番目の期間は POYOM3DTO1HOM (3日 2時間) の場合、関数は P6DT4H (6日 2時間) を返します。

7.10.12.11 duration-from-parts

引数とて提供された以下の部分を組み合わせて計算された型 xs:duration の値を返します: year.month.day.hour. minute.second.millisecond.negative。

サンプルの期間はP1Y2M3DT04H05M59S:

- [P] か期間の指定子で、必須です;
- 残りの文字は以下の順序で表示されます:1年、2カ月、3日、T(時間指定子)、04時間、05分、59秒。

マイナス文字が[P] 指定子の前に表示されると、これはマイナスの期間を示しています。例 -PID.

f ₈ duration-from-parts			
⊳year			
⊳month			
¢ day			
် <mark>hour</mark>			
՝>minute	duration		
second			
millisecond			
negative			

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明	
year	xs:int	年の値を与えます。	
month	xs:int	月の値を与えます。	
day	xs:int	月の日付を与えます。	
hour	xs:int	任意。時間の値を与えます。	
minute	xs:int	任意。分の値を与えます。	
second	xs:int	任意。秒の値を与えます。	
millisecond	xs:decimal	任意。ミノ利の値を与えます。	
negative	xs:boolean	任意。負の期間の場合 true である必要があり、それ以外の場合はfalse です。	

サンプル

以下のマピングは1年、4カ月、17日、8時間、58分、54.333秒の負の期間を生成します。



マピング出力は-P1Y4M17DT8H58M54.3335です。

7.10.12.12 duration-subtract

duration1 からduration2 を減算し取得されたxs:duration 値を返します。

サンプルの期間はP1Y2M3DT04H05M59S:

- [P] か期間の指定子で、必須です;
- 残りの文字は以下の順序で表示されます:1年、2カ月、3日、T(時間指定子)、04時間、05分、59秒。

マイナス文字が[P] 指定子の前に表示されると、これはマイナスの期間を示しています。例 -PID.

	$f_{\!\!8}$ duration-subtract	
Ç	>duration1	recult P
Ç	>duration2	result

言語

Built-in、C++、C#、Java。

サンプル

duration1 がP0Y0M0DT05H07M(5時間7分)でduration2 がPT1H(1時間)の場合、関数はPT4H7M(4時間7分)を返します。

7.10.12.13 hour-from-datetime

引数とて与えられたxs:dateTime 値が整数の値とて時間数を返します。

	$\mathbf{f}_{\mathbf{x}}$	hour-from	-datetime
ç	datetime		hour ር

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02+05:00 の場合、関数は9 を返します。

7.10.12.14 hour-from-duration

引数出て与えられたxs:duration 値から整数の値出て時間数を返します。

	$f_{\!\scriptscriptstyle R}$ hour-from-duration		
duration		uration	hour 🗘

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がPOYOMODT05H07M の場合、関数は5 を返します。

7.10.12.15 leapyear

現在のxs:dateTime クラスからるう年の場合、true を返します。それ以外の場合 false を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2020-04-17T09:30:02+02:00 の場合、2020 はうろう年のため 関数は true を返します。

7.10.12.16 millisecond-from-datetime

引数出て与えられたxs:dateTime 値が整数の値出て引刷数をxs:decimal 出て返します。

∫[∗] millisecond-from-datetime ▷ datetime millisecond ▷

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は544 を返します。

7.10.12.17 millisecond-from-duration

引数出て与えられたxs:duration 値が整数の値出て引刷数をxs:decimal 出て返します。

	$f_{\!\!8}$ millisecond-from-duration		d-from-duration
ç	>dı	uration	millisecond 🖡

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP0Y0M0DT05H07M02.227s の場合、関数は227 を返します。

7.10.12.18 minute-from-datetime

引数出て与えられているxs:dateTime 値が整数の値出て分数を返します。

∫_∗ minute-from-datetime ▷ datetime minute ▷

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は30 を返します。

7.10.12.19 minute-from-duration

引数とて与えられたxs:duration 値から整数の値とて分数を返します。

	$\mathbf{f}_{\mathbf{x}}$	minute-fro	m-duration
¢	duration		minute 🕻

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP0Y0M0DT05H07M02.227s の場合、関数は7 を返します。

7.10.12.20 month-from-datetime

引数とて与えられたxs:dateTime 値が整数の値とて月数を返します。

∫_∗ month-from-datetime ▷ datetime month ▷

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は12 を返します。

7.10.12.21 month-from-duration

引数とて与えられたxs:duration 値が整数の値とて月数を返します。

	f_8	month-fro	m-duration
Ç	duration		month 🕻

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP0Y04M0DT05H07M02.227S の場合、関数は4 を返します。

7.10.12.22 now

(タイムゾーンを含む)現在の日時をxs:dateTime 値とて返します。



言語 Built-in、C++、C#、Java。

サンプル 以下のマピングは現在の日時を出力します:出力はマピングを実行する都度変更されます。



サンプルの出力は2020-04-17T11:42:34.684+02:00 です。

1日前の値を得る方法については core | lang | datetime-add 関数を参照くたさい。

7.10.12.23 remove-timezone

(型xs:dateTimeの) time入力/デメーターからのタイムゾーンコンポーネトを削除します。

	f ₈ remove	-timezone
Ç	>time	time Þ

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
time	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

time が2001-12-17T09:30:02+05:00 の場合、関数は2001-12-17T09:30:02 を返します。

7.10.12.24 second-from-datetime

引数とて与えられたxs:dateTime 値が整数の値とて秒数を返します。

∫_∗ second-from-datetime ▷ datetime second ▷

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は2 を返します。

7.10.12.25 second-from-duration

引数とて与えられたxs:duration 値が整数の値とて秒数を返します。

	f _x	second-from-duration	
duration		iration	second

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP0Y04M0DT05H07M02.227s の場合、関数は2 を返します。

7.10.12.26 time-from-datetime

引数出て与えられたxs:dateTime 値からxs:time 値出て時間コンポーネトを返します。

∫_x time-from-datetime ⊳datetime time⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は09:31:02+05:00 を返します。

7.10.12.27 timezone

引数として与えられたxs:dateTime 値からタイムゾーンオフセットを分数で返します。UTC のために0 を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は300 を返します。

7.10.12.28 weekday

引数とて与えられたxs:dateTime 値が整数の値とて曜日を返します。関数は値1を月曜のために値2を火曜のために返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

7.10.12.29 weeknumber

引数とて与えられたxs:dateTime 値から週番号を返します。関数は値1を年度の最初の週のために値2を2番目の週のために返します。

J_x weeknumber P datetime weeknumber P

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

7.10.12.30 year-from-datetime

引数とて与えられたxs:dateTime 値から整数の値とて年数を返します。

∫_x year-from-datetime ▷ datetime year ▷

言語 Built-in、C++、C#、Java。

パラメーター

名前	型	説明
datetime	xs:dateTime	型 xs:dateTime の入力値を提供します。

サンプル

datetime が2001-12-17T09:30:02.544+05:00 の場合、関数は2001 を返します。

7.10.12.31 year-from-duration

引数とて与えられたxs:duration 値が整数の値とて年数を返します。

	f_{x} year-from-duration		
Ç	> dı	uration	year 🗘

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値を提供します。

サンプル

duration がP01Y04M0DT05H07M02.227S の場合、関数は1 を返します。

7.10.13 lang | file functions (ファイル関数)

バイナリコンテンンをファイルから読み取る、おゴコンアイルマンドナリコンテンンを書き込むためにfile 関数を lang ライブラから使用します。

7.10.13.1 read-binary-file

この関数は型 xs:base64Binary のBLOB (バイナリラージオブシンケ) とて指定されファイルのエンテンを返します。データ型が [base64Binary] と呼ばれる場合でも内部の表記はBLOB です。関数の結果を型 xs:base64Binary のXML ノードにマップする 場合のみ、ベース64 エンコード しないます。関数の結果を xs:hexBinary、データベーズブロブ、おけよプロトコール バッファー構造内のバ イナリフィールド にマップすることもできます。

	f_8 read-binary-file		
0	>filepath	contract P	l
ľ	>must-exist	content	ſ

バイナリファイルをマピングに読み込むはは前lepath 引数に対しての入力としていなを提供します。filepath か相対的な場合 MapForce はマピングと同じディレクトリ内でファイルを探します。must-exist 引数が狂意です。ファイルを開くことはできず、このパラメー ターはtrue, してより、マピングはエラーをスローします。ファイルを開くことができなく、このパラメーターがfalse の場合、からのパイナルが返されます。

言語

Built-in。

名前	型	説明
filepath	xs:string	ファイルな。
must-exist	xs:boolean	任意の ラメーターです。 ファイルを開くことが できなく、この ラメーターがtrue の場合、 マピングはエラーをスローします。 ファイルを

名前	型	説明
		開くことができなく、このパラメーターがfalse の場合、からのパイナルが返されます。
		デフォルトの値は、true です。

サンプル

例バイナリファイルの読み取りを参照してくたさい。

7.10.13.2 write-binary-file

この関数は、イナリニンテンを指定されたファイル、マに書き込み、書き込まれたファイルの、マを返します。、イイナリファイルが唯一の希望する 出力の場合 単純型出力 コンポーネントに関数の結果を接続します。出力がマンピング内で使用される都度この関数はファイルに書き込む ため、間に他の処理を使用することなく、関数の結果を直接ターゲトコンポーネントに接続することが奨励されます。

f ₈ write-binary-file	
┝ filepath	filenath
content	niepath D

バイナリファイルを書き込むコはfilepath 引数に対しての入力として、文を提供します。filepath が相対的な場合 MapForce はマピングと同じディレケーリ内でファイルを生成します。The content 引数は実際の、イナリニンテンツ(例えば、データベース内のBLOB フィールド)に接続されている必要があります。

MapForce 内でマピングをプレビューすると、ディスタにファイルを直接書き込む代わりに関数は一時的なファイルをデフォルトで生成します。

一時的なファイルをディスクニ保存するコよ、最初に出力 タブをクリックし適用可能な場合 生成された出力を保存 🕒 おっよ全ての 生成済みの出力 🚰 ツール いーボシンをクリックします。

MapForce が出力を一時的なファイルではなくファイナルファイルに直接書き込むように構成するには[ツール|オプション メニューコマンド] を選択し [全般]をクトックして [最終出力ファイルに直接書き込む] オプションを選択します。 このオプションは同じ名前を持つ既存のファイ ルを上書きすることに注意してくたさい。

この関数はファイルカディスクに保存されていたい場合(マッピングをプレビューし [最終出力ファイルに直接書き込む] オプションカ無効化されている場合)でも常に最終(一時的ではない)ファイル名を返します。

マピングが自身の出力ファイルを読み取ることはサポートされていないてとご注意してくたさい。

言語

Built-in。

名前	型	説明
filepath	xs:string	入力ファイル 🎗

名前	型	説明
content	xs:base64Binary	型 xs:base64Binary のイナロンテ ンソ

サンプル

例バイナノファイルの書き込みを参照してくたさい。

7.10.14 lang | generator functions (ジェネレーター関数)

lang ライブラルらのジェネレーター関数は植を生成する関数です(現在 create-guid はこの種類の唯一の関数です)。

7.10.14.1 create-guid

グロー・シレー意識別子(GUID)を16進数にエンコードされた文字列として作成します。この関数はデータベースフィールド、おけま他のコンポ ーネント型のためにマンピングから直接一意の値を生成するために使用されます。関数 format-guid-string を参照してくたさい。

∫₈ create-guid result k

言語

Built-in、C++、C#、Java。

7.10.15 lang | logical functions (論理関数)

lang ライブラルらの関数にはブール値理論を使用する多種の値を評価する関数が含まれています。

7.10.15.1 logical-xor

value1 がvalue2 と異なる場合 true を返し、それ以外の場合はfalse を返します。

	f ₈ logical-xor		
C	>value1	re cult l	L
Ç	>value2	result	ſ

言語

 $Built-in, C++, C\#, Java_{\circ}$

パラメーター

名前	型	説明
value1	xs:boolean	最初の入力値。
value2	xs:boolean	2番目の入力値。

7.10.15.2 negative

入力値が資の値(ゼロより小さ)値)の場合 true を返し、それ以外の場合は false を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:decimal	入力値。

7.10.15.3 numeric

入力値が数値、おけま文字として解析することができる文字列である場合 true を返し、それ以外の場合は false を返します。



言語

Built-in、C++、C#、Java。

名前	型	説明
value	xs:decimal	入力值。

サンプル

入力値が文字列 "4.33" の場合、 関数は true を返します。 入力値が文字列 "4.33 USD" の場合、 関数は false を返します。

7.10.15.4 positive

入力値が正の値(ゼロより大きい値)の場合 true を返し、それ以外の場合は false を返します。

∫₈ positive ⊳value result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:decimal	入力値。

7.10.16 lang | math functions (数学関数)

lang ライブラからの数学関数をマピング内で異なる数学オペレーションを実行するために使用することができます。

7.10.16.1 abs

引数とて提供された数値の絶対値を返します。引数が資の値ではない場合、引数が返されます。引数が資の値の場合、引数のネゲーションが返されます。



言語

Built-in、C++、C#、Java。

名前	型	説明
value	xs:decimal	入力値。

7.10.16.2 acos

-pi/2 からpi/2 までの範囲で value のアークコサインを返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.3 asin

-pi/2 からpi/2 までの範囲で value のアークサインを返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.4 atan

-pi/2 からpi/2 までの範囲で value のアークタンジェントを返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.5 cos

value により与えらえた角度の三角コサインのを返します。値のユニナトはラジアンです。

	f ₈ cos	
C	>value	result 🗘

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.6 degrees

ラジアンで測定された角度を度数で測定されたまま等価の角度に変換します。



言語

Built-in、C++、C#、Java。

名前	型	説明
value	xs:double	入力值。

7.10.16.7 divide-integer

value1 を value2 て割オ-整数結果を返します。

$f_{\!\!8}$ divide-integer		
¢value1	racult N	
¢value2	result	

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value1	xs:decimal	最初の入力値。
value2	xs:decimal	2番目の入力値。

サンプル

最初の引数が15 で2番目の値が2 の場合、関数は7 を返します。

7.10.16.8 exp

value に累乗したオイラー数 e が返されます。



言語

Built-in、C++、C#、Java。

名前	型	説明
value	xs:double	入力值。

7.10.16.9 log

value の(底がeの)自然対数を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.10 log10

value の(底が10 の) 10 進数対数を返します。

∫_x log10 ⊳value result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.11 max

引数とて提供された数値の最大値の数値を返します。デフォルトでは、この関数には2つの、ラメータのみ存在しますが、更に、ラメーターを追加することができます。 **パラメーターの追加(回)** おけよ**パラメーターの消除(図)** をクリックしてパラメーターを追加まけば削除します。 関数の引数の追加、まけば削除も参照してくたさい。

result

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value1	xs:decimal	最初の入力値。
value2	xs:decimal	2番目の入力値。
valueN	xs:decimal	N 番目の入力値。

7.10.16.12 min

引数とて提供された数値の最小値の数値を返します。デフォルトでは、この関数には2つの、ラメータのみ存在しますが、更に、ラメーターを 追加することができます。パラメーターの追加(回)ませはパラメーターの削除(図)をクリックして、ラメーターを追加ませせ消除します。 関数の引数の追加、ませず削除も参照してくたさい。



言語

Built-in、C++、C#、Java。

名前	型	説明
value1	xs:decimal	最初の入力値。
value2	xs:decimal	2番目の入力値。
valueN	xs:decimal	N 番目の入力値。
7.10.16.13 pi

数理定数 pi の値を返します。



言語

Built-in、C++、C#、Java。

7.10.16.14 pow

bに累乗されたaの値を返します。

	$\mathbf{f}_{\mathbf{x}}$	pow	
Ç	>a	recult [l
ç	>b	result	ſ

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
а	xs:double	値 a (底)を返します。
b	xs:double	値 b (累乗)を返します。

7.10.16.15 radians

度数で則定された角度をラジアンで則定されたまで等価の角度に変換します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.16 random

0.0 と等しいませまたき、、そして 1.0 よりた小さい値を正号と共に返します。戻り値は範囲内からにまう 一様な分布を使用して疑似的に選択されます。



言語

Built-in、C++、C#、Java。

7.10.16.17 sin

value のサインを返します。

∫_x sin ⊳value result ⊳

言語

Built-in、C++、C#、Java。

名前	型	説明
value	xs:double	入力值。

7.10.16.18 sqrt

value の正確に端数処理された平方根を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.19 tan

value のタンジェントを返します。

∫_x tan ⊳value result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.16.20 unary-minus

サインされた入力値の負の値を返します。



Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:decimal	入力値。

サンプル

入力文字列が3の場合、関数は-3を返します。入力文字列が-3の場合、関数は3を返します。

7.10.17 lang | QName functions (QName 関数)

lang ライブラルらのQname 関数は修飾名(QName)値を文字列にします。おけな文字列を修飾名値に変換します。core ライブラ いからの関数とは異なり、これらの関数はビルトイン、Java、C#、おけはC++ 言語でのみ使用することができます。

7.10.17.1 QName-as-string

引数として提供された Qname 値の文字列表記を返します。

f ₈ QName-as-string	
QName	result 🖓

言語

Built-in、C++、C#、Java。

パラメーター

名前	説明
QName	入力 xs:QName 值。

7.10.17.2 string-as-QName

QName の文字列表記をQName に変換します。

f ₈ string-as-QName		
់string result 🕻		

Built-in、C++、C#、Java。

名前	説明
string	入力 string 值。

7.10.18 lang | string functions (文字列関数)

lang ライブラルらの文字列関数により文字列を処理することが可能になります(例えば、トリミング、パディング、置換、文字列を大文字またはオリン文字に置き換えるなどの操作)。

7.10.18.1 capitalize

各単語の頭文字が大文字に変換されている入力文字列 value てを返します。

∫₈ capitalize ≥value result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:string	入力值。

サンプル

入力文字列が The quick brown fox の場合、関数はThe Quick Brown Fox を返します。

7.10.18.2 charset-decode

charset-decode 関数は、入力文字列デーを取り、Base64 テキストとしてデコードします。指定された文字セル(例えば "utf-8")に従いデータをデコードし、結果する文字列の値を設定します。Base64 テキストとして以前にエンコードされている/ ゾナリデータをデコー ドするコよ <u>charset-encode</u> 関数を使用します。

$f_{\rm s}$ charset-decode	
binary	
encoding	string 🕻
error-abort	

Built-in。

パラメーター

名前	型	説明
binary-data	xs:base64Binary	Base64 テキストとての ゙イナリデータ。
encoding	<pre>xs:string</pre>	エンコードのかかっ使用される文字セオ (例、 "utf-8")。
error-abort	xs:boolean	エラーが発生した場合処理がどのように継続されるかを指定する任意の引数。有効な値: ・ true - 無効な文字の例外により処理を終了します。 ・ false - 処理を続行し、無効な 文字を置換文字 ♥ と置き換 えます。
		デフォルトの値は true です。

サンプル

ソースXML ファイルカシダナナデーゲティードすると仮定します。Message 要素はBase64 テキストとしてエンコードされている イナナノデーダを含むことに注意してくたさい。

```
<?xml version="1.0" encoding="UTF-8"?>
<message xsi:noNamespaceSchemaLocation="message.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">TG9yZW0gaXBzdW0=</message>
```

スキーマで示されるとおり、 message 要素のデータ型はxs:base64Binary です:

上記のメッセージをデコードするマッピングは以下の通りです



このサンプルのマピングはデキスト "Lorem ipsum" を出力します。

MapForce シリアルビエンポーネントを使用して、マピングはBase64 データとしてエンコードされているテキスト おけはXML ファイルを処 理することもできます。例えば、下に示さるマピングリゴは、Base64 テキスト デーダを期待する input パラメーター か存在します。 charsetencode サンプルで示されるように XML ファイルから Base64 データが作成されていると仮定し、基の XML ファイルを下のマピングで示さ れるように再作成することができます。



このマピング内では error-abort 引数は、booleann ビルドイン関数を使用して作成された false の値を取得します。これにより、 無効な文字が発生しても、処理が継続されることが保証されます。 関数の文字列の結果は、XML ファイルニ変換される XML 解析エンポ ーネント につ マされます。 XML 解析を可能にするために、XSD スキーマファイルが必要にないます。 詳細に関しては、文字列の解析とノア ノルを参照してくたさい。

7.10.18.3 charset-encode

charset-encode 関数は 入力文字列デーを取り Base64 テキスト とてエンコードします。指定された文字列セナ(例えば ~utf-8″)内でエンコードされたデータよ xs:base64Binary 型として返されます。Base64 テキストとして以前にエンコードされてしるバ イナリデータをデコードするコよ charset-decode 関数を使用します。

	$f_{\!8}$ charset-encode	
0	> string	
0	encoding	binary 🕻
ľ	substitute	

言語

Built-in。

パラメーター

名前	型	説明
string-data	xs:string	エンコードされる文字列データ
encoding	xs:string	エンコードのかかっ使用される文字セナ (例、 "utf-8")。
substitute	xs:string	無効な文字が発生した場合、置換の文字 を指定する任意の引数。非-Unicode エ ンコードを使用する場合、この引数を適用 できます。Unicode エンコードのかかの、 置換文字は ? です。

サンプル

UTF-8 文字セナを使用して、テキスト "Lorem ipsum"をBase64 デーダをエンコードしターゲナ XML ファイルに書き込むと仮定します。ターゲナ XML ファイルコは、xs:base64Binary 型のmessage 要素がスキーマによし示されているようご存在します。

Base64 エノコードを実行するマノピングは以下の通りです。



このマッピングは下のような XML 出力を生成します (スキーマ参照とXML 宣言はスキップされています):

<message>TG9yZW0gaXBzdW0=</message>

MapForce シアル化コンポーネトを使用して、serializationテキストおけまXML ファイルを Base64 としてエンコードすることができます。例、下で示されるマピングは ソースXML ファイルを文字列にシノアル化します。結果する文字列は charset-encode 関数の引数として提供されます。最後に、関数の結果は、単純型出カコンポーネトを使用してマピングの出力として返されます。 文字列の値を返すを参照してくたさい。シノアル化の情報に関しては、文字列の解析とノノアル化を参照してくたさい。



7.10.18.4 count-substring

string 内で発生する substr の回数を表した整数値が返されます。

$f_{\rm R}$ count-substring		
I <mark>⊳st</mark> r	ing	recult P
¢su	bstr	result

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列。
substr	xs:string	テストするサブ文字列。

サンプル

次のマンングは2を返します。これは入力文字列 id | name | email 内で発生する イプセ レーターの発生する回数です。



7.10.18.5 empty

入力文字列が空の場合 true を返します。それ以外の場合は false を返します。



言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:string	入力值。

7.10.18.6 find-substring

string 内のsubstr の最初の発生の位置を返します。デフォルトではポジュン(インデックス)1 を持ち、最初の文字から検索を開始す る関数は最初の文字列からの検索を無視しますが、開始インデックスを任意で指定することができます。substr か見つからない場合関数 は0 を返します。

f_{s} find-substring	
l> string	
¢ substr	result 🛱
startindex	

言語

Built-in、C++、C#、Java。

名前	型	説明
string	xs:string	入力文字列。
substr	xs:string	検索するサブ文字列。
startindex	xs:int	任意。検索の開始ポジョン(インデックス) を指定します。ノラメーターが指定されてい ない場合、検索はポジンョン」で開始されま す。

サンプル

次のマピングは入力文字列 id | name | email 内の パプ文字の最初の発生のポジョンである3 を生成します。



4 を開始インデックスと指定する場合、関数の検索は4番目の文字から検索します。この結果、下のマンピングは4番目の文字以降からの検索した2、イプ文字の最初の発生である8を出力します。



7.10.18.7 format-guid-string

通常データベースのフィーリドで使用するオメクス正しくフォーマナされたグローノシレー意識別子(GUID)がformatted_guidから返されます。create-guid 関数の項目も参照くたとい。

∫_∗ format-guid-string ▷unformatted_guid formatted_guid ▷

言語

Built-in、C++、C#、Java。

名前	型	説明
unformatted_guid	xs:string	入力16進エンコード文字列のみかウォー マナンされます。

7.10.18.8 left

入力文字列の最初のnumber 文字を含む文字列を返します。

	$f_{\scriptscriptstyle R}$ left		
0	> string	racult [l
0	>number	result	ĺ

言語

 $Built-in, C++, C#, Java_{\circ}$

パラメーター

名前	型	説明
string	xs:string	入力文字列。
number	xs:int	文字列先頭から開始して返される文字列 の数値を指定します。

サンプル

入力文字列がThis is a sentence で、数値が4の場合、関数はThis を返します。

7.10.18.9 left-trim

全ての先頭空白文字が削除されている入力文字列を返します。

∫_x left-trim ▷string result ▷

言語

Built-in、C++、C#、Java。

名前	型	説明
string	xs:string	入力文字列。

7.10.18.10 lowercase

入力 string を小文字に変換します。Unicode の文字の場合、(Unicode コンノーシアムにお定義された)対応する小文字が使用されます。

f_x lowercase

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列。

7.10.18.11 match-pattern

pattern によに定義された正規表現が入力文字列に一致する場合、ブール値 true を返し、それ以外の場合には false を返します。 正規表現も参照してくたさい。

メモ C++、C#、おゴま Java コードを生成する際コよ 正規表現構文の高度な機能が異なる可能性があります。詳細に関しては 各言語の正規表現ドキュメトを確認してくたさい。

f_{8} match-pattern			
0	> string	re cult l	L
0	>pattern	result L	ſ

言語

Built-in、C++、C#、Java。

名前	型	説明
string	xs:string	入力文字列。
pattern	xs:string	一致する正規表現。

サンプル

以下のマピングは多種の個人のタイトルを検証します。具体的にはマピングは以下のタイトルに対してtrueを出力する必要があります: Mr、Mrs、Mx、Ms、Miss。



入力文字列が上記以外のタイトルの場合、マノピングはfalse を返します。

7.10.18.12 pad-string-left

単一の特定文字に
以
広
に
い
%
ド
さ
れ
た
文
字
列
を
返
し
ま
す
。
希
望
す
る
文
字
列
の
長
さ
と
、
ッ
デ
ィ
ン
グ
文
字
l
お
l
数
と
、
て
与
え
ら
れ
ま
す
。

	f ₈ pad-string-left	
0	> string	
0	desired-length	result 🛱
0	>padding-char	

言語

Built-in、C++、C#、Java。

名前	型	説明
string	xs:string	入力文字列を指定する。
desired-length	xs:int	パッディングの後の文字列の希望する長さを 定義する。
padding-char	xs:string	パッディング文字として使用する文字を定義 する。

7.10.18.13 pad-string-right

単一の特定文字により店にいったされた文字列を返します。希望する文字列の長さとッディング文字は別とて与えられます。

f _x pad-string-right	
string 🖓	
desired-length	result C
padding-char	

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列を指定する。
desired-length	xs:int	パッディングの後の文字列の希望する長さを 定義する。
padding-char	xs:string	パッディング文字として使用する文字を定義 する。

7.10.18.14 repeat-string

引数 n 回与えられた文字列を繰り返します。 count 引数は、文字列の繰り返しの回数を指定します。



言語

Built-in、C++、C#、Java。

名前	型	説明
string	xs:string	入力文字列。
count	xs:int	文字列を繰り返す回数。

7.10.18.15 replace

value によ得られた入力文字列の方、oldstring の文字列に一致する各文字列がnewstring の文字列に置き換えられます。

$f_{\!\!8}$ replace	
¢ value	
loldstring	result 🖡
newstring	

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
value	xs:string	入力值。
oldstring	xs:string	置き換えられる古い文字列。
newstring	xs:string	置換としての役割を果たす新規の文字 列。

サンプル 特殊文字の置換を参照してください。

7.10.18.16 reversefind-substring

string 内のsubstr の最後の発生の位置を返します。デフォルトではポジション(インデックス)1 を持ち、最後の文字で検索を終了する 関数は最初の文字列からの検索を無視しますが、終了インデックスを任意で指定することができます。substr か見つからない場合関数は 0 を返します。

$f_{\rm s}$ reversefind-substring	
l> string	
⊳ substr	result 🖗
endindex	

言語 Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列。
substr	xs:string	検索するサブ文字列。
endindex	xs:int	任意。検索の終了ポジョン(インデックス) を指定します。ノラメーターが指定されてい ない場合、検索はstring内の最後の文 字で終了します。

サンプル

次のマピングは入力文字列 id | name | email 内の パプ文字の最後の発生のポジョンである8 を生成します。



4を終了インデックスと指定する場合、関数の検索は4番目の文字まで検索します。この結果、下のマメピングは3を出力します。



7.10.18.17 right

入力文字列の最後のnumber 文字を含む文字列を返します。

$f_{\!8}$ right	
⊳ string	racult [
¢ <mark>number</mark>	result

言語 Built-in、C++、C#、Java。

サンプル

入力文字列がThe brown red fox で、数値が3の場合、関数はfoxを返します。

7.10.18.18 right-trim

全ての行末空白文字が削除されている入力文字列を返します。

∫₈ right-trim ⊳string result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列。

7.10.18.19 string-compare

大文字と小文字を考慮してstring1 とstring2 の文字列の比較の結果を返します。両方の文字列が等価の場合 0 が返されます。 string1 がstring2 より小さ 場合 -1 が返されます。string1 がstring2 より大き 場合 1 が返されます。

f ₈ string-compare	
⊳string1	racult [
⊳string2	result

言語

Built-in、C++、C#、Java。

名前	型	説明
string1	xs:string	最初の入力文字列。
string2	xs:string	2番目の入力文字列。

7.10.18.20 string-compare-ignore-case

大文字と小文字を無視した string1 と string2 の比較文字列の結果が返されます。両方の文字列が等価の場合 0 が返されます。 string1 が string2 より小さ、場合 -1 が返されます。 string1 が string2 より大き、場合 1 が返されます。

f _x string-compare-ignore-case	
⊳string1	
⊳string2	result

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string1	xs:string	最初の入力文字列。
string2	<pre>xs:string</pre>	2番目の入力文字列。

7.10.18.21 uppercase

入力 string を大文字に変換します。Unicode の文字の場合、(Unicode コンノーシアムに比定義された)対応する大文字が使用されます。

∫₈ uppercase ⊳string result ⊳

言語

Built-in、C++、C#、Java。

パラメーター

名前	型	説明
string	xs:string	入力文字列。

7.10.19 xpath2 | accessors

xpath2 | accessors サブライブラカシの関数はXMLノードませオアイテムに関する情報を抽出します。これらは関数はXSLT2 また はXQuery 言語が選択されている状態で利用することができます。

7.10.19.1 base-uri

base-uri 関数はノードを入力とて取り、ノードを含むXMLリソースのURIを返します。出力は型xs:string型になります。



言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
node	mf:node	入カ ノー ド。

7.10.19.2 node-name

node-name 関数は入力/デメーターとしてノードを受け取り、そのQName を返します。QName か文字列として表示され、ノードにプレ フィックスかある場合はprefix:localname 形式の入力を受け取り、ノードにプレフィックスか無い場合はlocalname 形式の入力が 受け取られます。ノードの名前空間 URI を取得するゴよ(qname-1式関連したfunctions 以下にある)<u>namespace-uri-from-</u> QName 関数を使用してくたさい。

∫₈ node-name ⊳node result ⊳

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
node	mf:node	入カ ノー ド。

7.10.19.3 string

string 関数は、xs:string のようご作動し、入力された値をxs:string 型に変換します。

入力が、例えばxs:decimalの原子型である場合、原子型の値がxs:string型の値に変換されます。入力引数が一下の場合、 ノードの文字列値が抽出されます。(ノードの文字列値とはノードの子孫ノードから得られた値を連結したものになります)。 ∫<mark>x string</mark> ⊳item result Þ

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
item	mf:item	入力値。

7.10.20 xpath2 | anyURI functions (anyURI 関数)

xpath2 | anyURI サブライブラルはresolve-uri 関数を含んでいます。この関数はXSLT2 おけはXQuery 言語が選択されている状態で利用することができます。

7.10.20.1 resolve-uri

resolve-uri 関数は最初の、ラメーターとして相対的な URI を取り、2番目の引数内のベース URI に対して解決します。結果は型xs:string になります。関数の実装は両方の入力を文字列として扱います。これらの URI によし識別されるリノースが実在するかチェック されません。

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
relative	xs:string	ベースコンれて解決される相対的なURI。
base	xs:string	ベースURI。

サンプル

下で表示されているマピング内では、最初の引数は相対的なURI MyFile.html を提供しており、2番目の引数はベースURI file:///C:/Dir/を提供しています。解決されたURI は2つを連結したもので file:///C:/Dir/MyFile.html になります。



7.10.21 xpath2 | boolean functions (ブール値関数)

boolean 関数 true ならびに false 関数 は 関数 と します。 定数の boolean 値が 要な場所で使用することができます。

7.10.21.1 false

boolean 値のfalse を挿入します。

∫₈ false false Þ

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

7.10.21.2 true

boolean 値のtrue を挿入します。



言語 XQuery、XSLT 2.0、XSLT 3.0。

7.10.22 xpath2 | constructors

XPath 2.0 ライブラルの[constructors] サブライブラル内の関数は入力テキストから特定のデータ型をエンストラクトします。以下のテーブル は使用可能なすべてのエンストラクタ関数をリストしてします。

xs:ENTITY	xs:double	xs:nonPositiveInteger
-----------	-----------	------------------------------

xs:ID	xs:duration	xs:normalizedString
xs:IDREF	xs:float	x s:positiveInteger
xs:NCName	xs:gDay	xs:short
xs:NMTOKEN	xs:gMonth	xs:string
xs:Name	xs:gMonthDay	xs:time
xs:QName	xs:gYear	xs:token
xs:anyURI	xs:gYearMonth	xs:unsignedByte
xs:base64Binary	xs:hexBinary	xs:unsignedInt
xs:boolean	xs:int	xs:unsignedLong
xs:byte	xs:integer	xs:unsignedShort
xs:date	xs:language	xs:untypedAtomic
xs:dateTime	xs:long	xs:yearMonthDuration
xs:dayTimeDuration	xs:negativeInteger	
xs:decimal	<pre>xs:nonNegativeInteger</pre>	

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

サンプル

通常、入力されたテキストはエンストライトされるデータ型に対応した書式で記述されている必要があります。そうでは、場合は変換に失敗します。例えば、xs:dateTime 値をxs:dateTime エンストラクタ関数を使用してエンストライトする場合、入力テキストは xs:dateTime データ型の書式である YYYY-MM-DDTHH:mm:ss の形式で記述されている必要があります。



上で説明されているマピング内では("2020-04-28T00:00:00") は関数の入力引数を提供するために使用されています。入力はノ ーズドコメント内のアイテムから取得することできます。 xs:dateTime 関数は型 xs:dateTime の値 2020-04-28T00:00:00 を返します。

(関数引数のデータ型を含む)マッピングアイテムの期待されるデータ型を確認するコはマウスカーノルを対応する入力ませま出力コネクタロポ 个トします。

7.10.23 xpath2 | context functions (コンテキスト関数)

xpath2 ライブラルらのコンテキスト関数は現在の日時のその他の精報、以前に使用されたデンォルトの照合順序、現在のシーケンスのサイズ、現在のノードのポジノコンを提供します。

7.10.23.1 current-date

システムクロックから得られた現在の日付(xs:date)を返します。

f₈ current-date result P

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

7.10.23.2 current-dateTime

システムクロックから得られた現在の日時(xs:dateTime)を返します。

<u>f</u>⁸ current-dateTime result P

言語 XQuery、XSLT 2.0、XSLT 3.0。

7.10.23.3 current-time

システムクロックから得られた現在の時刻(xs:time)を返します。

f_x current-time result P

言語 XQuery、XSLT 2.0、XSLT 3.0。

7.10.23.4 default-collation

default-collation 関数は入力パラメーターを取らず、デフォルトの照合、つまい指定することができる関数に対して特定の照合か指定されなかった場合に使用される照合を返します。

max-string とmin-string 関数を含む比較はデフォルの照会順序をベースしています。



言語 XQuery、XSLT 2.0、XSLT 3.0。

7.10.23.5 implicit-timezone

評価コンテキストからの[implicit timezone] プロ・ティの値を返します。

f₈ implicit-timezone result P

言語

XQuery <code>XSLT 2.0</code> <code>XSLT 3.0</code> $_{\circ}$

7.10.23.6 last

現在処理されているアイテムのシーケンス内のアイテムの数量を返します。下のサンプルで説明される通り、重要な点はアイテムのシーケンスは現在のマンピングエンテキストにより決定されます。



言語 XQuery、XSLT 2.0、XSLT 3.0。

サンプル

次のノースXML ファイルが存在すると仮定します:

目的はデーダを異なるスキーマを持つXML ファイルニピーすることです。また、全てのアイテムのカウトトはターゲル XML ファイルニ保存される必要があります。これは以下のようなマッピング内で達成することができます:



上のサンプルではlast 関数は現在の親コンテキスト内と最後のノードの位置を返し、値3を持つ count 属性を生成します。

値3はArticleとitemsの接続によれたマンングコンテキスト内の最後のアイテムすなわちすべてのアイテムのカウト)のポジ ションです。この接続が存在してい場合でもアイテムはターゲットにコピーされますが反復する親コンテキストが存在していすっかりast 関数は 値1を間違って返します。(具体的には、両方のコンポーネントのルートアイテム間で作成され、期待される通り3個ではなく1個のアイテムの シーケンスを作成する)デフォルトの明示的なコンテキストを使用します。

一般的にはlast 関数の代わりに core からの count 関数を使用することが奨励されます。これは前者がマンビングコンテキストを明示的に変更することのできる parent-context 引数を持っているからです。

7.10.24 xpath2 | durations, date and time functions (期間、日付、および時刻関数)

xpath2 ライブラルらの期間、日付、およい時刻関数により日付と時刻の値内でタイムゾーンを調整することができ、特定のコンポーネントを日付、時刻、および期間値から抽出し、日付時刻の値を減算します。

<Articles> <Article>

</Article>
<Article>

</Article>
<Article>

</Article>

<Name>T-Shirt</Name>

<Name>Socks</Name>

<Name>Jacket</Name>

<SinglePrice>25</SinglePrice>

<SinglePrice>2.30</SinglePrice>

<SinglePrice>57.50</SinglePrice>

タイムゾーンの調整

日付および時刻の値内でタイムゾーンを調節するコよ、以下の関数を使用することができます

- adjust-date-to-timezone
- adjust-date-to-timezone(タイムゾーン引数を持つ)
- adjust-dateTime-to-timezone
- adjust-dateTime-to-timezone(タイムゾーン引数を持つ)
- adjust-time-to-timezone
- adjust-time-to-timezone (タイムゾーン引数を持つ)

これら関数はそれそれxs:date、xs:time、おけはxs:dateTime 値を最初の引数とて取り、(存在する場合)2番目の ラメータ ーから得られた値により、タイムゾーンの削除や編集を行います。

次のシチュエーションは最初の引数にタイムゾーンか含まれたい場合のみ可能です(例えば日付2020-01 まけは時刻14:00:00)。

- timezone 引数が存在する場合、結果にはお番目の引数により指定されたタイムゾーンが含まれています。2番目の引数内のタイムゾーンが追加されます。
- timezone 引数が不在の場合、結果にコンステムのタイムゾーンである明示的なタイムゾーンが含まれます。システムのタイムゾーンが追加されます。
- timezone 引数が不在の場合、結果にはタイムゾーンは含まれません。

次のンチュエーションは最初の引数にタイムジーンか含まれたい場合のみ可能です(例えば、日付 2020-01-01+01:00 おけまう) 14:00:00+01:00)。

- timezone 引数が存在する場合、結果にコお番目の引数にお指定されたタイムゾーンが含まれています。オバンナルのタイムゾーンがを番目の引数のタイムゾーンに置き換えられます。
- timezone 引数が不在の場合、結果にコエンステムのタイムゾーンである明示的なタイムゾーンか含まれます。オドジナルのタイムゾ ーンパシステムのタイムゾーンに置き換えられます。
- timezone 引数が不在の場合、結果にはタイムゾーンは含まれません。

日付と時刻のコンポーネントの抽出

時刻、分数、日数、月数などの数値を日付と時刻の値から抽出するコよ、以下の関数を使用することができます。

- day-from-date
- day-from-dateTime
- hours-from-dateTime
- hours-from-time
- minutes-from-dateTime
- minutes-from-time
- month-from-date
- month-from-dateTime
- seconds-from-dateTime
- seconds-from-time
- timezone-from-date
- timezone-from-dateTime
- timezone-from-time
- year-from-date
- year-from-dateTime

892 関数

これらの関数はそれそれxs:date、xs:time、xs:dateTime、およびxs:duration 値から特定のひたやホイを抽出します。 結果はxs:integer おはxs:decimal です。

期間からコンポーネントを抽出する

時刻コンポーネートを期間から抽出するコよ、以下の関数を使用することができます。

- days-from-duration
- hours-from-duration
- minutes-from-duration
- months-from-duration
- seconds-from-duration
- years-from-duration

期間は年数と月数を抽出するけるの xs:yearMonthDuration おけま日数、時間数、分数、秒数を抽出するけるの xs:dayTimeDuration とて指定される必要が別ます。xs:decimal を返す seconds-from-duration 関数の例外と共に型 xs:integer の結果を返します。

日付と時刻の値の減算

日付と時刻の値を減算するコよ、以下の関数を使用することができます

- subtract-dateTimes
- subtract-dates
- subtract-times

それそれの減算関数により、時間に関する値を他の値から差し引き、期間の値を得ることができます。

7.10.24.1 adjust-date-to-timezone

xs:date 値を評価コンテキスト(システムのタイムゾーン)内で明示的なタイムゾーンに調節します。

∫<mark>∦ adjust-date-to-timezone</mark> P date result P

言語

XQuery、XSLT 2.0、XSLT 3.0。

ノデメーター

名前	型	説明
date	xs:date	型 xs:date の入力值。

サンプル

次のマピングはxs:date から文字列をコンストライトし adjust-date-to-timezone 関数の引数とて提供します。



XSLT 2.0 マッピング

システムのタイムゾーンが+02:00のエンピューターでマンピングが作動する場合、関数は日付の値がシステムのタイムゾーンを含むように調整します。この結果、マンピングの出力は2020-04-30+02:00 にないます。

7.10.24.2 adjust-date-to-timezone

xs:date 値を特定のタイムゾーンまけおイムゾーン無しに調整します。timezone 引数が空のシーケンスの場合、関数はタイムゾーン 無しでxs:dateTime を返します。それ以外の場合、タイムゾーン付きでxs:date を返します。

f ₈ adjust-date-	to-timezone
🗟 date	
timezone	result

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
date	xs:date	型 xs:date の入力值。
timezone	xs:dayTimeDuration	タイムゾーンはxs:dayTimeDuration 値とて表示されます。この値は負の値であ ることができます。例えば-5 時間のタイムゾ ーンの値は-PT5H と表記することができま す。

サンプル

XPath 2 <u>constructor</u> 関数を使用して次のマピングはadjust-date-to-timezone 関数への両方の デメーターを文字列から コンストラオトします。このマピングの目的はタイムゾーンを -5 時間に調整することです。このタイムゾーンは -PT5H と表記することができま す。



XSLT 2.0 マッピング

この関数は引数とて与えられたタイムゾーンに日付の植を調整します。この結果、マルピングの出力は2020-04-30-05:00 になります。

7.10.24.3 adjust-dateTime-to-timezone

xs:dateTime 値を評価コンテキスト(システムのタイムゾーン)内で明示的なタイムゾーンに調節します。

	<pre>f₈ adjust-dateTime-to-timezone</pre>			
ç	>da	ateTime	result 🖓	,

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。

7.10.24.4 adjust-dateTime-to-timezone

xs:dateTime 値を特定のタイムゾーンまけおタイムゾーン無しに調整します。timezone 引数が空のシーケンスの場合、関数はタイム ゾーン無しでxs:dateTime を返します。それ以外の場合、タイムゾーン付きでxs:dateTime を返します。

f _x adjust-dateTim	e-to-timezone
dateTime	
timezone	result

言語 XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。
timezone	xs:dayTimeDuration	タイムゾーンはxs:dayTimeDuration 値とて表示されます。この値は負の値であ ることができます。例えば-5 時間のタイムゾ ーンの値は - PT5H と表記することができま す。

7.10.24.5 adjust-time-to-timezone

xs:time 値を評価コンテキスト(システムのタイムゾーン)内で明示的なタイムゾーンに調節します。

∫₈ adjust-time-to-timezone ⊳time result ⊳

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。

7.10.24.6 adjust-time-to-timezone

xs:time 値を特定のタイムゾーンまけ」おイムゾーン無しに調整します。timezone 引数が空のシーケンスの場合、関数はタイムゾーン 無しでxs:time を返します。それ以外の場合、タイムゾーン付きでxs:time を返します。

J ₈ adjust-time-to-timezone	
⊳time	consult P
timezone	result

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。
timezone	xs:dayTimeDuration	タイムゾーンはxs:dayTimeDuration 値と、て表示されます。この値は負の値であ ることができます。例えば-5時間のタイムゾ ーンの値は-PT5H と表記することができま す。

7.10.24.7 day-from-date

xs:date 值の日付の部分を表す xs:integer を返します。

∫_x day-from-date ▷ date result ▷

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
date	xs:date	型 xs:date の入力值。

サンプル

xs:date コンストラクタ関数を使用して次のマンピングは文字列をxs:date に変換します。day-from-date、month-fromdate、およびyear-from-date 関数は日付の関連する部分を抽出しターゲル XML ファイル内の個別のアイテムに書き込みます。



XQuery 1.0 マルング

マピングの出力は以下のようてなります。

7.10.24.8 day-from-dateTime

xs:dateTime 値の日付の部分を表す xs:integer を返します。

∫₈ day-from-dateTime ▷ dateTime result ▷

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。

7.10.24.9 days-from-duration

引数とて与えられた期間の値の正規表記の日付コンポーネトの部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

サンプル

下で表示されるマピングは P2DT1H (2 日と1 時間) のxs:dayTimeDuration をエンハーラトし、入力とて days-fromduration 関数に提供します。結果は2 です。



XSLT 2.0 マッピング

メモ 期間がP1DT24H(1日と24時間)の場合、関数は1ではなく2を返します。これはP1DT24Hの正規表現が実際 P2D (2日)であるからです。

7.10.24.10 hours-from-dateTime

xs:dateTime 値の時間の部分を表す xs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。

7.10.24.11 hours-from-duration

引数とて与えられた期間の値の正規表記の時間コンポーネトの部分を表す xs: integer を返します。

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

サンプル

期間がPT1H60M(1時間と60分)の場合、関数は1ではなくを2返します。これはPT1H60Mの正規表現が実際にはPT2H(2時間であるからです。

7.10.24.12 hours-from-time

xs:time 值の時間の部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。

7.10.24.13 minutes-from-dateTime

xs:dateTime 値の分数の部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。

7.10.24.14 minutes-from-duration

引数とて与えられた期間の値の正規表記の分数コンポーネトの部分を表すxs:integer を返します。

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

サンプル

期間がPT1M60S (1 分 60 秒) の場合 1 ではなく 2 を返します。これは PT1M60S の正規表現は PT2M (2 分)たからです。

7.10.24.15 minutes-from-time

xs:time 值の分数の部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。

7.10.24.16 month-from-date

xs:date 值の月数の部分を表す xs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
date	xs:date	型 xs:date の入力値。

7.10.24.17 month-from-dateTime

xs:dateTime 値の月数の部分を表すxs:integer を返します。

言語

XQuery XSLT 2.0 $XSLT 3.0_{\circ}$

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。
7.10.24.18 months-from-duration

引数として与えられた期間の値の正規表記の月数コンポーネトの部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

7.10.24.19 seconds-from-dateTime

date Time のローカライズされた値内の秒コンポーネントを表す xs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
dateTime	xs:dateTime	

7.10.24.20 seconds-from-duration

引数とて与えられた期間の値の正規表記の秒数コンポーネトの部分を表すxs:integer を返します。

言語

XQuery XSLT 2.0 $XSLT 3.0_{\circ}$

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

7.10.24.21 seconds-from-time

引数として提供されたxs:time 値の分数の部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。

7.10.24.22 subtract-dateTimes

dateTime1の正規化された値とdateTime2の正規化された値間の差分に対応するxs:dayTimeDurationを返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
dateTime1	xs:dateTime	最初の入力値。
dateTime2	xs:dateTime	2番目の入力値。

7.10.24.23 subtract-dates

date1の正規化された値とdate2の正規化された値間の差分に対応するxs:dayTimeDurationを返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

名前	型	説明
date1	xs:date	最初の入力値。
date2	xs:date	2番目の入力値。

サンプル

下に示されているマメピングは2つの日付の減算を表しています(2020-10-22から2020-09-22を差し引く減算)。結果は30日を表す型xs:dayTimeDurationの値 P30D です。



7.10.24.24 subtract-times

time1の正規化された値とtime2の正規化された値間の差分に対応するxs:dayTimeDurationを返します。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
time1	xs:time	最初の入力値。
time2	xs:time	2番目の入力値。

7.10.24.25 timezone-from-date

引数とて提供された日付のタイムゾーンコンポーネントを返します。結果はUTC からの偏差を示す xs:dayTimeDuration 。この植は+14:00 から-14:00 時間の両方の値が含まれる範囲です。

言語

XQuery、XSLT 2.0、XSLT 3.0。

名前	型	説明
date	xs:date	型 xs:date の入力値。

7.10.24.26 timezone-from-dateTime

引数とて提供されたxs:dateTime 値のタイムゾーンマンポーネントを返します。結果はUTC からの偏差を示す xs:dayTimeDuration。この値は+14:00 から-14:00 時間の両方の値が含まれる範囲です。

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
dateTime	xs:dateTime	型xs:dateTime の入力値。

7.10.24.27 timezone-from-time

引数とて提供されたxs:time 値のタイムゾーンエンポーネントを返します。結果はUTC からの偏差を示す xs:dayTimeDuration。この値は+14:00 から-14:00 時間の両方の値が含まれる範囲です。

言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
time	xs:time	型 xs:time の入力値。

7.10.24.28 year-from-date

xs:date 値の年数の部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

名前	型	説明
date	xs:date	型 xs:date の入力值。

7.10.24.29 year-from-dateTime

xs:dateTime 値の年数の部分を表す xs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
dateTime	xs:dateTime	型 xs:dateTime の入力値。

7.10.24.30 years-from-duration

引数とて与えられた期間の値の正規構文表記の年数コンポーネトの部分を表すxs:integer を返します。

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
duration	xs:duration	型 xs:duration の入力値。

7.10.25 xpath2 | node functions (ノード関数)

xpath2 ライブラカらのノード関数はマッピングコンポーネント上のノード (アイテム)の情報を与えます。

lang 関数は([en] などの)言語コードを識別する文字列の引数を取ります。関数はエンテキストノードに関数の引数に一致する値を持つxml:lang 属性があるがに従いtrue ませまfalse を返します。

local-name、name、および namespace-uri 関数はそれぞれローカル名、名前、入カノードの名前空間 URI を返します。例え ば、ノード altova: Products のローカル名は Products、名前は altova: Products、名前空間 URI は altova: prefix が、 ウイ される名前空間の URI です(<u>local-name</u>関数のナダのサンプルを参照してくたき、)。これら3個の関数はは2つの パアント が存 在します:

- 引数無し: 関数はコンテキストノードに適用されます(コンテキストノードのサンプルのためはは<u>lang</u> 関数のために与えられたサンプルを参照してくたさい)。
- ノードである必要がある引数: 関数は接続済みのノードに適用されます。

number 関数はノードを入力として取り、ノードを自動化し、すなわち、コンテンンを抽出し)、およい植を小数に変換し、変換された値を返します。number 関数の2つの・リアントが存在します:

- 引数無し: 関数はエンテキストノードに適用されます(コンテキストノードのサンプルのためは1111mg 関数のために与えられたサンプルを参照してくたさい)。
- ノードである必要がある引数: 関数は接続済みのノードに適用されます。

7.10.25.1 lang

コンテキストノードがtestlang 引数に完全一致する、おけよリブセルである値を持つxml:lang 属性を持つ場合 true を返します。それ以外の場合関数はfalse を返します。

∫_∗ lang ≥testlang result Þ

言語

XQuery、XSLT 2.0、XSLT 3.0_{\circ}

パラメーター

名前	型	説明
testlang	xs:string	チェックする言語コードは、例えば[en] で す。

サンプル

以下のXML はxml:lang 属性のために異なる植を持つpara 要素を含んでいます。

下に示されてるマピングはlang 関数を使用して国名のリアトに関わらずドイン語の段落のみをフィルターします。



XSLT 2.0 マッピング

上のマピング内では、ソース内の各 para 内でitem はターゲト内に条件付けられて作成されます。1ang 関数がtrue を返す箇所の ノードのみをターゲナトに、なするフィルターにより条件は提供されています。すなわち、xml:lang 属性を[de]([de]のサブセナ)に設定し ているこれらのノードはフィルターの条件を満たします。この結果、マピングの出力は以下の通りです:



para とitem の間の親接続のためlang 関数は各 para のコンテキスト内で操作されます。マンピングコンテキストを参照してくたさい。

7.10.25.2 local-name

引数とて提供されたコンテキストノードの名前のローカルの部分をxs:stringとて返します。これはlocal-name 関数の ラメーター ー無しの リアントです。ノードを明示的に指定するコは入力ノードを ラメーターとして取る local-name 関数を使用してくたさい。



言語

XQuery、XSLT 2.0、XSLT 3.0 $_{\circ}$

7.10.25.3 local-name

node の名前のローカルの部分をxs:string とて返します。



言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
node	node()	入カノード。

サンプル

次のXML ファイル内ではp:product 要素の名前はプレフィックスされた装飾名 (QName) です。プレフィックス [p] は名前空間 http://mycompany.com にマップされています。

次のマピングはローカル名、名前、およびシードの名前空間 URI を抽出し、これらの値をターゲトファイルに書き込みます:



XSLT 2.0 マッピング

マピングの出力は以下のように表示します:各 col アイテムはlocal-name、name、および namespace-uri 関数の結果をリストしています。

7.10.25.4 name

コンテキストノードの名前を返します。これはname 関数のパラメーター無しのパリアントです。ノードを明示的に指定するコは入力ノードを引数としてとる name 関数を使用してください。



言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

7.10.25.5 name

ノードの名前を返します。

∫₈ name ⊳node result ⊳

言語

XQuery、XSLT 2.0。

パラメーター

名前	型	説明
node	node()	入カ ノー ド。

サンプル

local-name 関数のために与えられたサンプルを参照してくたさい。

7.10.25.6 namespace-uri

コンテキストノードのQNameの名前空間URIをxs:stringとて返します。コンテキストノードがマピング内の接続により決定される 個所のnamespace-uri 関数の ラメーター 無しの リアント です。ノードを明示的に指定する ゴは 入力ノードを ラメーターとて取る namespace-uri 関数を使用してくたさい。



言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

7.10.25.7 namespace-uri

xs:string とてのnode のQName 名前空間 URI を返します。

<mark>∫₈ namespace-uri</mark> ⊳node result ⊳

言語

XQuery、XSLT 2.0。

パラメーター

名前	型	説明
node	node()	入力ノーギ。

サンプル

local-name 関数のために与えられたサンプルを参照してくたさい。

7.10.25.8 number

xs:doubleに変換されたコンテキストノードの値を返します。これはnumber 関数の、ラメーター無しの、リアントです。ノードを明示的に 指定するコよ入力ノードを引数として取るnumber 関数を使用してくたさい。

数値に変換可能な方はBoolean、数値文字列、他の数値型のみです。(非数値文字列などの)非数値入力値は数字ではな) NaN内に結果します。



言語 XQuery、XSLT 2.0、XSLT 3.0。

7.10.25.9 number

xs:double に変換された node の値を返します。数値に変換可能な方は Boolean、数値文字列、他の数値型のみです。(非数値 文字列などの非数値入力値は数字ではない) NaN 内に結果します。



言語

XQuery, XSLT 2.0, XSLT 3.0_{\circ}

パラメーター

名前	型	説明
node	mf:atomic	入カノーギ。

サンプル

次のXML は型 string のアイテムを含んでします:



下で説明されているマッピングはこれらすべての文字列を数値に変換しターゲット XML ファイルに書き込もうとします。ターゲット XML コンポーネント内のitem のデータ型はxs:integer ですが、ソースitem はxs:string データノースです。変換に成功しない場合、アイテムはスキップされなければならず、ターゲットファイルにコピーされません。



XSLT 2.0 マッピング

マピングの目的を達成するためにフィルターが使用されています。equal 関数は変換の結果が[NaN] であるかをチェックします。これが false の場合、これは変換の成功を示し、アイテムはターゲットにコピーされます。マピングの出力は以下のようてなります:

```
<items>
<item>1</item>
<item>2</item>
</items>
```

7.10.26 xpath2 | numeric functions (数値関数)

xpath2 ライブラルの数値関数にはabs とround-half-to-even 関数か含まれます。

7.10.26.1 abs

引数とて提供された絶対値を返します。例えば入力引数が-2 ませま2 の場合、関数は2 を返します。



言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
value	xs:decimal	入力值。

7.10.26.2 round-half-to-even

round-half-to-even 関数は任意の2番目の引数内で提供された数値(最初の引数)を精度(小数の数)の度数に提供します。 例えば、最初の引数が2.141567 で2番目の引数が3 の場合、最初の引数(数値)は3番目の小数で端数処理され、結果は2.141 しています。精度(2番目パラメーター)が指定されていない場合、番号は小数が無いようご満数処理され与えられた値は整数にないます。

与えられている数値内の数字がつつ値の中間にある場合関数の名前内の低数に数値を低数にご満数処理することを指しています。例えば round-half-to-even (3.475, 2) は 3.48 とう値を返します。

	f ₈ round-half-to-even	
Ç	>value	
ţ	precision	

言語 XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
value	xs:decimal	端数処理される入力値を提供する必須の 引数。
precision	xs:integer	端数処理する小数点の数値を指定する任 意の引数。デフォルトの値は0です。

7.10.27 xpath2 | string functions (文字列関数)

xpath2 ライブラルの文字列関数により文字列を処理することができます(これには文字列の比較、文字列を大文字ませるい文字への変換、文字列ませるその他からサブ文字列の抽出などか含まれます)。

7.10.27.1 codepoints-to-string

Unicode コードポイントのシーケンスから文字列を作成します。この関数はstring-to-codepoints 関数の反対です。

 f_x
 codepoints-to-string

 > codepoints
 result >

言語

XQuery <code>XSLT 2.0</code> <code>XSLT 3.0</code> $_{\circ}$

パラメーター

名前	型	説明
codepoints	ZeroOrMore xs:integer	この入力は各整数がUnicode コードポイントを指定する箇所で整数型のアイテムのシーケンスは接続されている必要があります。

サンプル

次のXML はUnicode コードポイント値を保管する複数のitem 要素を含んでます。

```
<item>99</item>
<item>101</item>
</items>
```

下で示されているマンピングはcodepoint-to-string 関数への引数とてアイテムのシーケンスを提供します。



XSLT 2.0 マッピング

マピング出力はMapForce です。

7.10.27.2 compare

Compare 関数は2つの文字列を引数とてとし、等価とアルファベナ順に比較します。string1 がstring2 よルアルファベナ順に比較して小さ、場合(例えば 2つの文字列が[A] と[B] の場合) 関数は-1 を返します。2つの文字列が等価の場合(例えば[A] と[A]) 関数は0 を返します。string1 がstring2 よル大き、場合(例えば[B] と[A]) 関数は1 を返します。

関数のこの・リアントはUnicode であるデンォルトの照会順序を使用しています。この関数の他のvariant は照会順序を引数として提供可能な箇所で存在しています。

f ₈ compare		
⊳string1		
⊳string2	result	

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string1	xs:string	最初の入力文字列。
string2	xs:string	2番目の入力文字列。

7.10.27.3 compare

引数とて提供された照会順序を使用して compare 関数は2つの文字列を引数とてとり、等価とアルファベナ順に比較します。 string1 がstring2 よりアルファベナ順に比較して小さ、場合(例えば 2つの文字列が[A] と[B] の場合) 関数は-1 を返します。 2つの文字列が等価の場合(例えば[A] と[A])関数は0 を返します。string1 がstring2 よ児大き、場合(例えば[B] と[A]) 関数 は1 を返します。

 fx
 compare

 > string1

 > string2

 result P

 > collation

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string1	xs:string	最初の入力文字列。
string2	xs:string	2番目の入力文字列。
collation	xs:string	文字列比較のかめご使用する照会順序を 指定します。この入力は <u>default-</u> <u>collation</u> 関数の出力を元にする可能 性があります、おけよ http://www.w3.org/2005/xpat h-functions/collation/html- ascii-case-insensitive などの 照会順序の場合かあります。

サンプル

以下のマピングは文字列 [A] と[a] を定数には提供される大文字と小文字を区別した照会順序

http://www.w3.org/2005/xpath-functions/collation/html-ascii-case-insensitive を使用して比較し ています。



XSLT 2.0 マッピング

上のマルングの結果は0です。これは両方の文字列が同様に扱われるこを意味します。default-collation 関数により与えられ ている照会順序を置き換えると照会順序はデフォルトのUnicode コードポイント照会順序に変更され、マルングの結果は-1 になりま す。([A] は[a] よりトアルファベナル順でおいさめらです)。



7.10.27.4 ends-with

string がsubstr で終わる場合はtrue を返し、それ以外の場合はfalse を返します。戻り値はxs:boolean です。

関数のこの リアトは Unicode であるデフォルの照会順序を使用しています。この関数の他の variant は照会順序を引数とて提供可能な箇所で存在しています。



言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string	xs:string	入力文字列(すなオち[haystack])。
substr	<pre>xs:string</pre>	サブ文字列(すなわち[needle])。

7.10.27.5 ends-with

string がsubstr で終わる場合はtrue を返し、それ以外の場合はfalse を返します。戻り値はxs:boolean です。

$f_{\rm x}$ ends-with		
⊳ string		
⊳substr	result 🛱	>
Collation		

言語 XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string	xs:string	入力文字列(すなわち[haystack])。
substr	xs:string	サブ文字列(すなわち[needle])。
collation	xs:string	文字列比較のかめご使用する照会順序を 指定します。この入力はdefault- collation 関数の出力を元にする可能 性があります、おけよ http://www.w3.org/2005/xpat h-functions/collation/html- ascii-case-insensitive などの 照会順序の場合があります。

7.10.27.6 lower-case

全ての文字を対応する小文字に変換後 string の値を返します。

<mark>∫₈ lower-case</sub> ⊳string result ⊳</mark>

言語

XQuery、XSLT 2.0_{\circ}

パラメーター

名前	型	説明
string	xs:string	入力值。

7.10.27.7 matches

matches 関数により、(input)ラメーターにより与えられた文字列が、(pattern)ラメーターから得られる)正規表現にマッチするかが チェックされます。正規表現の構文は、XML スキーマのpattern ファセナトにて定義する必要があります。文字列が正規表現にマッチすればtrue が返され、それ以外の場合はfalse が返されます。

	f _x matches		
ç	>input		
ç	>pattern	result 🕻	
ł	>flags		

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明	
input	xs:string	入力文字列。	
pattern	xs:string	一致する正規表現に関しては正規表現を参照していたい。	
flags	xs:string	一致に影響を与える任意の引数。この引数は以下のアナグの任意の組み合わせを与え る場合かあります: i、m、s、x。例えばimx のように、複数のアラブを使用することもで きます。 フラグが使用されていない場合、4 つのアラグのデフォルト値が使用されます。4 種類あるフラグは以下の通りです:	
		立 大文字と小文字を区別するモードを使用します。デフォルトでは大文字と 小文字を区別します。	
		m 複数行モードを使用して、改行文字(x0a)によ複数行に分けられてい る入力文字列を、1 つの文字列として扱います。メタ文字のへとまが、そ れぞれ行頭と行末を表します。デフォルトは文字列モードになっており、個々 の文字列がへから始まりまで終わります。	
		s dotall モドを使用します。デフォルト ではメタ文字は新規のライン文字 (x0a)以外のすべての文字に一致する not-dot-all モドです。dot-all mode ではボットはすべての新規ライン文字に一致します。	
		× 空白文字を無視する。デフォルトでは空白文字は無視されません。	

7.10.27.8 normalize-unicode

指定されている正規化書式のルールゴ従い正規化されている string の値 (2番目の引数)を返します。Unicode 正規化は関する詳細は 象.2 of https://www.w3.org/TR/charmod-norm/ を参照してくたい。

f ₈ normalize-unicode			
ç	> sti	ring	
ţ	> no	ormalizationForm	result

言語 XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string	xs:string	正規化される文字列の値
normalizationForm	<pre>xs:string</pre>	正規化フォームを提供するオプションの別 数。デオオルトはUnicode 正規化形式 C (NFC) です。 フォームNFC、NFD、NFKC、および NFKD の正規化は、サポートされていま す。

7.10.27.9 replace

この関数は入力文字列、正規表現、置換文字列を引数とてとます。置換文字列を使用して入力文字列内の正規表現のすべての一致を置き換えます。入力文字列内で正規表現が2つのオーバーラップする文字列に一致する場合、最初の一致のみが置き換えられます。

f ₈ replace	
l>input	
▷ pattern	
replacement	result
flags	

言語

XQuery、XSLT 2.0、XSLT 3.0。

名前	型	説明
input	xs:string	入力文字列。
pattern	xs:string	ー致する正規表現に関しては正規表現 を参照してくたさい。
replacement	<pre>xs:string</pre>	置換文字列。
flags	<pre>xs:string</pre>	一致に影響を与える任意の引数。この引 数は <u>matches</u> 関数内のflags 引数と 同じ方法で使用されます。

7.10.27.10 starts-with

string がsubstr で始まる場合はtrue を返し、それ以外の場合はfalse を返します。戻り値はxs:boolean です。文字列の比較は指定されている照会順序に従い行われます。

$f_{\rm x}$ starts-with	
⊳ string	
⊳substr	result 🕻
Collation	

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string	xs:string	入力文字列(すなわち[haystack])。
substr	xs:string	サブ文字列(すなわち[needle])。
collation	xs:string	文字列比較のかめご使用する照会順序を 指定します。この入力は <u>default-</u> <u>collation</u> 関数の出力を元にする可能 性かあります、おけよ http://www.w3.org/2005/xpat h-functions/collation/html- ascii-case-insensitive などの 照会順序の場合があります。

サンプル

デフォルトのUnicode 照会順序が使用されるため入力文字列 [MapForce] はサブ文字列 [Map] から開始するため次のマピングは 値 true を返します。



7.10.27.11 string-to-codepoints

引数とて提供された文字列を構成するUnicode コードポイント(整数値) のシーケンスを返します。この関数は codepoints-tostring 関数の反対です。

f ₈ string-to	-codepoints
>input	result 🖓

言語

XQuery、XSLT 2.0、XSLT 3.0 $_{\circ}$

パラメーター

名前	型	説明
input	xs:string	入力文字列。

7.10.27.12 substring-after

文字列 arg2 の後に発生する文字列 arg1 の部分を返します。

$f_{\rm R}$ substring-after		
≥arg1		
₽arg2	result 🗟	
Collation]	

言語

XQuery、XSLT 2.0、XSLT 3.0。

名前	型	説明
arg1	xs:string	入力文字列(すなわち[haystack])。
arg2	xs:string	サブ文字列(すなわち[needle])。
collation	xs:string	文字列比較のつめご使用する照会順序を 指定します。この入力は <u>default-</u> <u>collation</u> 関数の出力を元にする可能 性があります、おけよ http://www.w3.org/2005/xpat h-functions/collation/html-

名前	型	説明
		ascii-case-insensitive などの 照会順序の場合がみます。

サンプル

arg1 が[MapForce]、arg2 が[Map] の場合、collation は<u>default-collation</u> で、関数は[Force] を返します。

7.10.27.13 substring-before

文字列 arg2 の前に発生する文字列 arg1 の部分を返します。

f ₈ substring-before	
>arg1	
⊳arg2	result 🖗
collation	

言語

XQuery、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
arg1	xs:string	入力文字列(すなわち[haystack])。
arg2	xs:string	サブ文字列(すなわち[needle])。
collation	xs:string	文字列比較のかめご使用する照会順序を 指定します。この入力は <u>default-</u> <u>collation</u> 関数の出力を元にする可能 性かあります、おけよ http://www.w3.org/2005/xpat h-functions/collation/html- ascii-case-insensitive などの 照会順序の場合かあります。

サンプル

arg1 が[MapForce]、arg2 が[Force] の場合 collation は default-collation で、関数は[Map] を返します。

7.10.27.14 upper-case

全ての文字を対応する大文字に変換後 string の値を返します。



言語

XQuery、XSLT 2.0。

パラメーター

名前	型	説明
string	xs:string	入力文字列。

7.10.28 xpath3 | external information functions

xpath3 ライブラルの外部情報関数はXSLT 実行環境に関する情報の取得、おけおト部リノースからのデータの抽出を可能にします。

7.10.28.1 available-environment-variables

environment-variable 関数に適切な環境変数名のリストを文字列の空である可能性のあるシーケンスで返します。

∫₈ available-environment-variables names ₽

言語 XSLT 3.0。

7.10.28.2 environment-variable

存在する場合システム環境関数の値を返します。戻り型はxs:stringです。

$f_{\rm x}$ environm	ent-variable
⊳name	result 🖓

言語

 $XSLT 3.0_{\circ}$

パラメーター

名前	型	説明
name	xs:string	環境変数の名前

7.10.28.3 unparsed-text

外部リノース(例えばファイル)を読み取り、リノースの文字列表記を返します。

f ₈ unparsed-text	
<mark>>hre</mark> f	recult N
encoding	result

言語

XSLT 3.0。

パラメーター

名前	型	説明
href	xs:string	URI レフォレンスのフォーム内の文字列
encoding	xs:string	オプションの引数。エシコードの名前を指定 します。例えば "UTF-8"、""UTF-16" な ど。エシコードが自動的に決定されない場 合 UTF-8 か想定されます。

7.10.28.4 unparsed-text-available

特定の引数を持つunparsed-text への呼び出しか成功するかを決定します。戻り型はxs:boolean です。

f ₈ unparsed-text-available		
hre	f	
encoding		result

言語 XSLT 3.0。

パラメーター

名前	型	説明
href	xs:string	URI レフォレンスのフォーム内の文字列
encoding	<pre>xs:string</pre>	オプションの引数。エンコードの名前を指定 します。例えば "UTF-8"、""UTF-16" な ど。エンコードが自動的に決定されない場 合 UTF-8 か想定されます。

7.10.28.5 unparsed-text-lines

外部リノース(例えば、ファイル)を読み取り、そのエレテンンを文字列のシーケンスとして返します。リノースの文字列表記がテキストのそれぞれのライノニンを1個表示されます。

f ₈ unparsed-text-lines	
l> href	lin oc l
encoding	linesu

言語

 $\rm XSLT \ 3.0_{\circ}$

パラメーター

名前	型	説明
href	xs:string	URI レファレンスのフォーム内の文字列
encoding	<pre>xs:string</pre>	オプションの引数。エコードの名前を指定 します。例えば "UTF-8"、""UTF-16" な ど。エンコードか自動的に決定されない場 合 UTF-8 か想定されます。

7.10.29 xpath3 | formatting functions

xpath3 ライブラを使用可能な書式設定関数は日付、時刻、および整数の値を書式設定するために使用されます。

7.10.29.1 format-date

表示のためにxs:date 値を含む文字列を返します。

f ₈ format-date	
⊳value	
▷ picture	
language]result 🖗
calendar	
place	

言語

 $\rm XSLT~3.0_{\circ}$

名前	型	説明
value	xs:date	書式設定する入力 xs:date 値です。 必須の ラメーターです。
picture	xs:string	必須の ラメーターです。 "XPath とXquery 関数と寅算子 3.1" W 3C 推奨のセクション 9.8.4.1 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
language	xs:string	任意の ラメーターです。 "XPath とXquery 関数と演算子 3.1" W 3C 推奨のセンション 9.8.4.8 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
カレンダー	xs:string	上記と同様です。
place	xs:string	上記と同様です。

7.10.29.2 format-dateTime

表示のためこxs:dateTime 値を含む文字列を返します。

f ₈ format-dateTime	
Value	
¢ picture]
language] result 🛱
calendar]
place]

言語

 $\rm XSLT~3.0_{\circ}$

名前	型	説明
value	xs:dateTime	書式設定する入力 xs:dateTime 値で す。
picture	xs:string	必須の ラメーターです。 "XPath とXquery 関数と寅算子 3.1" W 3C 推奨のセクション 9.8.4.1 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
language	xs:string	任意の デメーターです。 "XPath とXquery 関数と演算子 3.1" W 3C 推奨のセンション 9.8.4.8 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
カレンダー	xs:string	上記と同様です。
place	xs:string	上記と同様です。

7.10.29.3 format-integer

指定されている場合与えられた自然言語の変換を使用して与えられた文字列に従い整数を書式設定します。

$f_{\rm s}$ format-integer	
¢ value	
picture	result 🛱
language	

言語

XSLT 3.0_{\circ}

名前	型	説明
value	xs:integer	書式設定する入力整数値
picture	xs:string	必須の ラメーターです。 ["] XPath とXquery 関数と演算子 3.1" W 3C 推奨のセクション 4.6.1 を参照して くたさい (https://www.w3.org/TR/xpath- functions-31)。
language	xs:string	任意の ラメーターです。 値か書式設定するかがご従う自然言語を 設定します。指定される場合、この値は空 の文字列まれは "拡張マークアップ言語 (XML) 1.0 W 3C 奨励 (https://www.w3.org/TR/xml) に従っ たxml:lang 属性のためご許可される値 しています。 according to the.

7.10.29.4 format-time

表示のためにxs:time 値を含む文字列を返します。

f ₈ format-time	
value	
⊳ picture	
language]result 🖗
calendar	
place	

言語

XSLT 3.0_{\circ}

パラメーター

名前	型	説明
value	xs:time	書式設定する入力xs:time値です。
picture	xs:string	必須の デメーターです。 "XPath とXquery 関数と寅算子 3.1" W 3C 推奨のセクション 9.8.4.1 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
language	<pre>xs:string</pre>	任意の デメーターです。 ["] XPath とXquery 関数と演算子 3.1" W 3C 推奨のセンション 9.8.4.8 を参照し てくたさい (https://www.w3.org/TR/xpath- functions-31)。
カレンダー	xs:string	上記と同様です。
place	xs:string	上記と同様です。

7.10.30 xpath3 | math functions

xpath3 ライブラルの数学関数は三角および他の数学計算を行うために使用されます。

7.10.30.1 acos

0 からpi までの範囲で角度のアークコサインを返します。



言語

XSLT 3.0°

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.2 asin

-pi/2 からpi/2 までの範囲で角度のアークサインを返します。

∫₈ asin ⊳value result ⊳

言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.3 atan

-pi/2 から pi/2 までの範囲で角度のアークタンジェントを返します。



言語

XSLT 3.0_{\circ}

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.4 atan2

座標(x, y)と正 x-軸を持つ平面乗の原点に対しての角度をラジアンで返します。

	f_8	atan2	
Ç	>y	recult [ĺ
Ç	>x	result L	ſ

言語

 $\rm XSLT 3.0_{\circ}$

パラメーター

名前	型	説明
у	xs:double	x 座標。
x	xs:double	y 座標。

7.10.30.5 cos

value により与えらえた角度の三角コサインを返します。値のユニナトはデジアンです。



言語 XSLT 3.0。

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.6 exp

value に累乗したオイラー数 e か返されます。

∫_x exp ⊳value result ⊳

言語

XSLT 3.0_{\circ}

パラメーター

名前	型	説明
value	xs:double	入力値。

7.10.30.7 exp10

10の値乗を返します。

∫_x exp10 ⊳value result ⊳

言語

 $\rm XSLT~3.0_{\circ}$

名前	型	説明
value	xs:double	入力值。

7.10.30.8 log

value の(底がe の) 自然対数を返します。



言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.9 log10

value の(底が10 の) 10 十進対数を返します。

<mark>∫₈ log10</mark> ≥value result Þ

言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.30.10 pi

数理定数 pi の近似値を返します。



言語

 $\rm XSLT~3.0_{\circ}$

7.10.30.11 pow

bに累乗されたaの値を返します。



言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
а	xs:double	入力值 a。
b	xs:double	入力值 b。

7.10.30.12 sin

value により与えらえた角度の三角サインを返します。値のユニントはラジアンです。



言語 XSIT3

 $\rm XSLT~3.0_{\circ}$

名前	型	説明
value	xs:double	入力值。

7.10.30.13 sqrt

引数の負以外平方根を返します。

∫₈ sqrt ⊳value result ⊳

言語

XSLT 3.0_{\circ}

パラメーター

名前	型	説明
value	xs:double	入力値。

7.10.30.14 tan

value により与えらえた角度の三角タンジェントを返します。値のユニットはラジアンです。



言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
value	xs:double	入力值。

7.10.31 xpath3 | URI functions

xpath3 ライブラ」内のURI 関数は、URI 内での使用を目的とたエノコード、エスケープおよし変換を行います。

7.10.31.1 encode-for-uri

URI の なセグメント 内で使用される予定の文字列内の予約された文字をエノコードします。この 関数の詳細については "XPath と Xquery 関数と演算子 3.1" W 3C 推奨のセグション 6.2 を参照してくたえい (<u>https://www.w3.org/TR/xpath-functions-31</u>)。



言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
uri-part	xs:string	エンコードする入力 URI 値。

7.10.31.2 escape-html-uri

HTML ユーザーエージェントがURI を含むことを期待される属性値を処理する方法と同じ方法でURI をエスケープします。この関数の詳細 については "XPath とXquery 関数と演算子 3.1" W 3C 推奨のセクション 6.4 を参照してくたさい (https://www.w3.org/TR/xpath-functions-31)。

f ₈ escape-html-uri	
l>uri	result 🖓

言語

 $\rm XSLT~3.0_{\circ}$

パラメーター

名前	型	説明
uri	xs:string	エスケープする入力 URI 値。

7.10.31.3 iri-to-uri

IRI (国際リノース識別子)を含む文字列をURI (統一資源識別子)に変換します。この関数の詳細については "XPath とXquery 関数と演算子 3.1" W 3C 推奨のセクション 6.3 を参照してくたさい (https://www.w3.org/TR/xpath-functions-31)。
	$\mathbf{f}_{\mathbf{x}}$	iri-to-uri uri C		
Ģ	>iri			ł

言語

XSLT 3.0°

パラメーター

名前	型	説明
iri	xs:string	入力 IRI 值。

7.10.32 xslt | xpath functions (xpath 関数)

このサブグループ内の関数はマルングアイテム(おけよノード)に関する情報を抽出するXPath 1.0 関数です。関数の多くはノードを引数として取り、そのノードに関する情報を返します。The last とposition 関数はマルング上の接続により決定される現在のマルングコン テキスト 内で作動します。

メモ その他のXPath 1.0 関数を core 関数 ライブラ に確認することできます。

7.10.32.1 lang

コンテキストノードがstring 引数に完全一致する、まけはサブセットである値を持つxml:lang 属性を持つ場合 true を返します。それ 以外の場合関数はfalse を返します。

∫₈ lang ⊳string result D

言語

XSLT 1.0°

パラメーター

名前	型	説明
string	xs:string	チェックする言語コードは、例えば[en] で す。

サンプル

xpath2 ライブラルのlang 関数のために与えられたサンプルを参照してくたさい。

7.10.32.2 last

処理されたノードリスト内の最後のノードのポジノョン番号を返します。



言語

 $XSLT \ 1.0_{\circ}$

サンプル

xpath2 ライブラノのlast 関数のために与えられたサンプルを参照してくたさい。

7.10.32.3 local-name

引数とて提供されたノードの名前のローカルの部分を返します。

<mark>∫₈ local-name</mark> ⊳node result ⊳

言語

XSLT 1.0, XSLT 2.0_{\circ}

パラメーター

名前	型	説明
node	node()	入力ノーギ。

サンプル

xpath2 ライブラルのlocal-name 関数のために与えられたサンプルを参照してくたさい。

7.10.32.4 name

引数とて提供されたノードの名前を返します。



言語

XSLT 1.0、XSLT 2.0。

パラメーター

名前	型	説明
node	node()	入カノ ー ド。

サンプル

xpath2 ライブラルのlocal-name 関数のために与えられたサンプルを参照してくたさい。

7.10.32.5 namespace-uri

引数とて提供されたノードの名前空間 URI を返します。

∫_x namespace-uri ⊳node result ⊳

言語

XSLT 1.0、XSLT 2.0。

パラメーター

名前	型	説明
node	node()	入力ノ ー ド。

サンプル

xpath2 ライブラルのlocal-name 関数のために与えられたサンプルを参照してくたさい。

7.10.32.6 position

現在処理されているノードセナ内の現在のノードのポジョンを返します。



言語

 $\rm XSLT~1.0_{\circ}$

7.10.33 xslt | xslt functions (xslt 関数)

このグループ内の関数はその他のXSLT 1.0 関数です。

7.10.33.1 current

current 関数は引数を取らず、現在のノードを返します。



言語

 $\rm XSLT~1.0_{\circ}$

7.10.33.2 document

外部 XML ドキュメント からノード にアクセスします。 関数の結果は、出力ドキュメント内にあるノードへ返されます。



言語

XSLT 1.0。

パラメーター

名前	型	説明
uri	xs:string	必須。XMLドキュメトへの、文を指定します。XMLドキュメントは有効で解析可能であ必要があます。
nodeset	node()	任意。ノード、および相対的である場合最 初の引数として与えられているURIを解 決するため使用されたベースURIを指定 します。

7.10.33.3 element-available

element-available 関数は関数の文字列引数とて入力されている要素がXSLT プロセッサーによりサポートされているかをテストします。引数文字列はQName とて評価されます。これらはマッピングのために生成される基になる XSLT 内のこれらの名前空間のために 宣言されているプレフィックスのため、XSLT 要素はxs1: プレフィックスを持つ必要があり、 XML スキーマ要素はxs: プレフィックスを持つ 必要があります。関数はブール値を返します。

∫_x element-available ⊳element result ⊳

言語

XSLT 1.0。

パラメーター

名前	型	説明
element	xs:string	要素名。

7.10.33.4 function-available

function-available 関数はelement-available 関数に類似しており、関数の別数と、て提供されている関数名がXSLT プロセッサーイことサポートされているかテストします。入力文字列はQName とて評価されます。関数はブール値を返します。

∫₈ function-available ⊳function result ⊳

言語

XSLT 1.0。

パラメーター

名前	型	説明
function	xs:string	関数名。

7.10.33.5 generate-id

generate-id 関数は任意の入力引数により識別されるノードセト内の最初のノードを識別する一意の文字列を生成します。入力ハラ メーターが与えられない場合、コンテキストノード上のID が生成されます。出力ドキュメントにある任意のノードに対して出力結果を接続する ことができます。 ∫₈ generate-id ≥nodeset result Þ

言語

XSLT 1.0, XSLT 2.0, XSLT 3.0。

パラメーター

名前	型	説明
nodeset	node()	入力ノードを提供するオプションの引数。

7.10.33.6 system-property

system-property 関数はXSLT プロセッサー(システムのプロ、ティを返します。入力された文字列はQName とて評価されるため XSLT スタイルシート内でXSLT 名前空間に関連付けられている xsl:prefix を使用する必要があます。これらはxsl:version、 xsl:vendor、とxsl:vendor-url です。基づる XSLT スタイルシート内のXSLT 名前空間とルフィックスが関連付けられるよめ Qname とて入力文字列は評価され xsl: プレフィックスを持つ必要があます。

∫_x system-property ⊳string result ⊳

言語 XSLT 1.0、XSLT 2.0、XSLT 3.0。

パラメーター

名前	型	説明
string	<pre>xs:string</pre>	以下のずれかであるプロ ディ名を指定し ます: xsl:version、xsl:vendor、 xsl:vendor-url。

7.10.33.7 unparsed-entity-uri

関数への入力文字列はDTD 内部にて宣言された(パースされていない)エンティティーにマッチする必要が砂ます。パースされていないエン ティティ(例えばイメージ)はパースされていないエンティティをロケートするURI を持ちます。関数の入力文字列はDTD 内で宣言されている パースされていないエンティティの名前と一致する必要が砂ます。関数は出力ドキュメント内のノードに移動されるパースされていないエンティ ティのURI を返します。例えば、href ノードなど。

f ₈ unparsed-entity-uri	
string	result 🗅

言語

XSLT 1.0_{\circ}

パラメーター

名前	型	説明
string	xs:string	URI が抽出される解析されていないエンティティの名前。

関数

8 マッピングの自動化と MapForce

MapForce を使用してデザインされたマッピングは Linux とmacOS サーバーを含む)サーバー環境内でサーバーレベルの ウォーマンス を使用して次のAltova 変換エンジノニて実行することができます (にれらのエンジノオ個別にライセンスが供与される必要が好)ます):

- RaptorXML Server: マルングの変換言語がXSLT 1.0、XSLT 2.0、XSLT 3.0 おけよXQuery の場合、マルングを このエンジンを使用して実行することが最適です。RaptorXML Server を使用して自動化するを参照してくたさい。
- MapForce Server (または MapForce Server Advanced Edition)。変換言語がBUILT-IN*の場合このエンジンの 使用が最適です。BUILT-IN 言語は MapForce 内のマナビング機能の多数をサポートしますが、MapForce Server (そし で特に、MapForce Server Advanced Editionin)はマナビングを実行するために最高の、ウォーマンスを提供します。 MapForce Server を使用して自動化するを参照してくたさい

*BUILT-IN 変換言語は、MapForce Professional おけまEnterprise Edition を必要とます。

更に、MapForce はよ、XSLT、XQuery、C#、C++、および Java コードをコマイ・ラインインターフェイスから自動的に生成する機能が搭載されています。これはよMapForce Server 実行を想定したサーバー実行ファイル(.mfx)をコンパイルする機能も含まれています。詳細は関しては、MapForce コマイドラインインターフェイスを参照してください。

8.1 RaptorXML Server を使用して自動化する

RaptorXML Server (今後は、略して RaptorXML)は、Altova 第3世代の超高速な XML とXBRL のよめのプロセッサーです。最新の標準と、ラレルコンピューティング環境のために最適化されています。クロスプラ・トフォームに対応可能で、エンジンは今日のマルチコアコンピューティングを有効に利用し、XML とXBRL データの高速処理を提供します。

RaptorXML はよ Altova ダウルロードページ(<u>https://www.altova.com/ja/download-trial-server.html</u>) からダウルロードしてイン ストールすることのできる複数のエディションかあります:

- RaptorXML Server は、XML、XML スキーマ、XSLT、XPath、XQuery、などへのサポートを搭載した高速な XML 処 理エンジンです。このエディションは、FlowForce Server インストール シケージの一部です。
- RaptorXML+XBRL Server は、XBRL の一連の標準を処理および検証する追加機能を使用して、RaptorXML Server の全ての機能をサポートします。

制約事項

- XML 署名はサポトされていません
- COM インターフェイスを介してグローノ ジレノノースはサポートされていません
- ODBC とADO データベース接続はWindows のみでサポートされています。他のオペレーティングシステムでは、JDBC を使用する必要があります。

XSLT おけまin XQuery、でコード生成するけま、MapForce は、生成の際に選択する出力フォルダーは保存される DoTransform.bat と呼ばれる、シチファイルを作成します。バッチファイルの実行は、RaptorXML Server を呼び出し、サーバー上で XSLT (おけまXQuery) 変換を行います、

XSLT/XQuery 変換を実行します。

サードー上の他の出力のために、MapForce マンングを実行おけは自動化するには、Altova MapForce Server を使用して自動 化するを参照してくたい。

メモ 内蔵のエンジンを使用して、XSLTとXQueryコードをプレビューすることができます。

8.2 MapForce Server を使用して自動化する

MapForce Server は、Windows、LinuxとmacOS オペーティングシステムのオダのエンタープライズサー・シノストウェアノリューショ ンです。MapForce Server の役割は(非-Windows プラナフォームを含む)サード・環境で、サードーレベルのマピングを実行する ことです。MapForce マピング内でBUILT-IN カダーゲナト実行言語である箇所でサード・実行をおこなうことができます(次も参照 変 換言語の選択)。MapForce Server は、(コマイ・ライン、おけよAPI から呼び出される)スタイドアロンで、おけまFlowForce Server の管理下で作動します。

MapForce Server がタイアロン製品とて使用されている場合、MapForce マンングは コン イルされ MapForce Server が 実行されているマンノニピーされる必要があります。MapForce Server コマイ・ランコマイ run を使用してマンとングは実行されます。 MapForce Server API の run メンドを呼び出してマンングを呼び出すこともできます。詳しい情報に関しては、以下を参照してくたさい、MapForce Server 実行ファイルへマンとノグをコンパイルする。

MapForce Server がFlowForce Server 管理下で実行されている場合、マンングをHTTP(おけよSSL/HTTPS) 接続を使用 して MapForce から、直接ターゲットマンノニデブロイすることができます。サーバー上では、マンングは、トリガーされた、おけよ予定された ジョブとして実行されます。おけよ、FlowForce Server 管理インターフェイスから定義されたサービス呼び出しを介して実行されることができ ます。詳しい情報に関しては、以下を参照してくたさい、<u>FlowForce Server にマンピングをデブロイする</u>。

MapForce Server には以下の2つのエディションがあります:

- MapForce Server
- MapForce Server Advanced Edition (このエディンコンはFlowForce Server インストール シケージの一部です)

MapForce Server Advanced Edition は MapForce Server と同じ機能が搭載されており、追加して、最適化を行うことのでき るマピングに対して、マピングを最適化する機能が含まれています。これは、多量のデータをジョイン、おけよフィルターするマピングの場合、 おけは多量のデータをフィルターする場合、実行アピードを向上するためにジョインの最適化が適用されます。MapForce Server とは異な り、MapForce Server Advanced Edition はノード関数が存在する箇所でマピングを実行することができます。次を参照してくたさい、 デフォルトンード関数。

制約事項

- XML 署名はサポートされていません
- ADO、ADO.NET、およびODBC データベース接続はWindows でのみりポートされており(その他のオペレーティングシステムに関してはサーバー実行のためにマンピングを準備を参照してください)。

MapForce Server に関する詳細は 付随するドキュメトを参照してたさい(https://www.altova.com/ja/documentation).

8.3 サーバーでの実行のためにマッピングを準備する

MapForce を使用してデザインされ、プレビューされるマンピングは、現在使用中のマシンと(データベースなどの) オペレーティングシステムの 外部にあるりノースを参照することができます。 更に、 MapForce 内では、全てのマンピングの マカデフォルト でWindows スタイルの規則 に従ってします。 封二、 MapForce Server が作動するサーバーでは、マンピングがデザインされたマシンと同じデータベースをサポートしない 場合にあります。 このため、サーバー環境でマンピングを実行するこは、 お二、ターゲットマシンがリースマンンと異なる場合には特に、若干の準備 が必要とされます。

メモ ソースマンともう用語は MapForce がインストールされているマンを指し、ターゲトマンノは、MapForce Server、おこ は、FlowForce Server がインストールされているマンを指します。最もシンプルなシナルオは、ソースマンンとターゲトマシンが 同じコンピューターの場合を指します。最も複雑なシナルガは、MapForce がWindows マシンで作動し、MapForce Server、お」は、FlowForce Server がLinux、お」は、macOS マンンで作動する場合を指します。

FlowForce Server にデプロイする前にMapForce 内でマピングの検証を常に行うことが奨励されます。おけよ MapForce Server 実行ファイルにエン イルすることが奨励されます (マピングの検証を参照してください)。

MapForce Server がスタイアロンで(FlowForce Server 無しで作動する場合、必要とれるライセンスは、以下のとおしです:

- ソースマシン上で、MapForce Enterprise、おは、Professional Edition がマピングをデザイン、サード実行ファイル(.mfx) につい イルするはないの要とされます。<u>MapForce Server 実行可能ファイルにマピングをコン イルする</u>を参照してくたさい。
- ターゲオマシン上で、MapForce Server、おけよ、MapForce Server Advanced Edition がマピングを実行するため に必要とされます。

MapForce Server がFlowForce Server 管理下で作動する場合、次の条件が適用されます:

- ソースマシン上で、MapForce Enterprise、おは、Professional Edition がマピングをデザイルて、ターゲトマシンに デプロイするためこ必要です。, FlowForce Server にマピングをデプロイするを参照してくたさい。
- MapForce Server とFlowForce Server かターゲトマンノニインストールされている必要かあります。MapForce Server の役割は、マピングの実行であり、FlowForceの役割はマピングを、スケジュールませま、オンデマドの実行、としての実行、 エラーの処理、条件付きの処理、電子メールの通知などの機能を活用することのできるジョブとして使用できるようにすることです。
- FlowForce Server は構成されているやトワークアドレスとポートで作動している必要があります。すなわち、「FlowForce Web Server」サービスが開始され、構成されており、HTTP(おけよ、構成されている場合はHTTPS)クライアーナからの接続をファイアウォールにブロックされることない受け入れる必要があります。「FlowForce Server」サービスも開始され、指定されたアドレスとポートで作動されている必要があります。
- コンテナーの1つへの、トミッションか与えられている Flow Force Server ユーザーアカウトを所有している必要が決ます(デフォ ルトでは、/public コンテナーは認証済みのユーザーであればアクセスすることができます)。

全般的な注意点

- スタイドアロン MapForce Server で作動するターゲナマンとのマピングを作動するコは、マピングには参照される全ての 入力ファイルをターゲナマンノこもピーする必要があります。MapForce Server がFlowForce Server 管理化で作動する 場合、ファイルを手動でピーする必要にありません。この場合、インスタンスとスキーマファイルはターゲナマンノにデプロイされた。シ ケージ内に含まれます。, see マピングをFlowForce Server にデプロイする。
- マピングに特定のデータベースドライバーを必要とするデータベースエンポーネトか含まれる場合、このようボドライバーはターゲト マシンノ きインストールされている必要があります。例えば、マピングがMicrosoft Access データベースからデータを読み取る場合、Microsoft Access、ませま、Microsoft Access Runtime がターゲトマシン きインストールされている必要があります(<u>https://www.microsoft.com/en-us/download/details.aspx?id=50040</u>)。
- マピングを非Windows プラナフォームニデプロイする場合、ADO、ADO.NET とODBC データベース接続は自動的に JDBC に変更されます。ネイティブ SQLite とネイティブ PostgreSQL 接続は保存され、追加の構成を必要としません。「データベース接続」も参照してください。
- (例えば、.dll、ませま、.class ファイルなどのカスタム関数呼び出しをマパングか管む場合、このような依存は、ランタイム前に

既知ではないため、マピングと共にデプロイされません。この場合、これを手動でターゲナマシノニピーすることができます。サーバー上の.dll おけて.class ファイルの なは MapForce 内の「ライブラの管理」ウイドウと同じである必要があます。例:

Manage Libraries	×	
🕞 🛃 New Design1		
User-Defined Functions		
Own Library Imports Add		
🖃 😋 Global Library Imports 🛛 Add		
🖓 🔂 C:\Libraries\Converters.dll		
- 🕀 Functions		I
f ₈ Converters, Version=1.0.0.0,	Show All Open [<u>O</u> ocuments
f _x Converters, Version=1.0.0.0, 🗸	Show <u>F</u> ile Paths	
Own Library Imports		
4]

- マンピングの一部は、ワイルドカード・パンを使用して複数の入力ファイルを読み取ります(<u>複数の入力ませ」は出力ファイルを動的に</u> <u>処理する</u>を参照してくたさい。この場合、入力ファイル名はランタイム前に既知ではないたっか、デプロイされません。マンピングの実 行に成功するコな、入力ファイルがターゲットマンン上に存在する必要があります。
- マッピングの出力、ペニディノトリが含まれている場合、これらのディノトリはターゲトマシン上に存在している必要があります。それ以外の場合、マメピングを実行するとエラーが生成されます。この振る舞いは、「一時的なファイルに出力を生成する」オプションが有効化されている場合既存ではおいディレクトリが自動的に生成されるMapForce とは異なります(MapForce オプションの変更を参照してくたさい)。
- マピングがクライアト証明書を使用するHTTPS 認証を必要とするWeb サービスを呼び出す場合、証明書をターゲトマン ンに手動で移動する必要があます。次も参照してくださいデジタル署名の管理。
- Microsoft Access とSQLite などのファイルをベースコンナデータベースコマンピングが接続される場合、データベースファイルは 手動でターゲナトマンノコ移動、お当よ参照することのできる共有されるディレオーリコ保存される必要があります。次を参照「ファ イルをベースコントデータベース」。

パマをポーダブルにする方法

サーバー上でマピングを実行する場合、マピングが適用することのできる変換パスを使用していること、および、サポートされるデータベース接続を使用していることを確認してください。

ノンを非Windows オーレーティングシステムへ移動可能にするためコよ、MapForce内でマーピングをデザインする際に絶対ノシの代わり に相対ノンを使用してくたさい。

- 1. 希望するマピングデザインファイル(.mfd)をWindows 上のMapForce で開きます。
- 2. 「ファイル」メニューでマッピングの設定を選択し、チェックされている場合「生成されたコード内でいてを絶対いてにする」 チェックボックスをクリアします。
- 3. 各マピングコンポーネントのケッションプロ・ティダイアログボックスを開き、(例えば、コンポーネントのタイトル・ーをダブルクトックして) 全てのファイル マを絶対・マから相対・マスご変更します。「MFD ファイルニ対して全ての・マを相対的に保存する」チェック ボックスを選択します。便宜上、マッピングと同様全ての入力ファイルとスキーマを同じファイルニニピーし、ファイル名で参照することが できます。

マピングのデザイン時に相対および絶対パマを処理するコは、相対、および、絶対パマの使用を参照してくたきい。

更に MapForce Server と FlowForce Server は、すてへの相対パマか解決されるしたから「作業ディレクトリ」をサポートします。 多種の実行環境内のパマを参照してくたさい。作業ディレクトリは以下のとおりマッピングのランタイムに指定することができます。

• FlowForce Server では、ジョブの「作業ディレクトリ」、ラメーターを編集します。

- MapForce Server API では、COM と.NET API のWorkingDirectory プロ ディーには、おけよ Java API の setWorkingDirectory メノバーにようことができます。
- MapForce Server コマバラインでは作業デルケリはコマイシェルの現在のデルケリです。

データベース接続

ADO、ADO.NET、およびODBC 接続はLinux とmacOS マンンではサポートされていないてとご注意して下さい。このため、ターゲットマシンがLinux おけまmacOS の場合、このような接続は、マメピングがFlowForce にデプロイされると、おけま、MapForce Server 実行ファイルにマメピングをエッ イルすると、JDBC に変換されます。この場合、マメピングをデプロイ、おけま、サーバー実行ファイル にマメピングをエッ イルする前に以下のオプションを使用することができます。

- MapForce 内では データベースへのJDBC 接続を作成します(JDBC 接続のセルアップを参照してくたき、)
- MapForce 内では、データベースコンポーネトの「JDBC 固有の設定」セクション内のJDBC データベース接続の詳細に入 カしてください、(データベースコンポーネト設定を参照してください)。

マルングがPostgreSQL 封まSQLite データベースへのオイティブな接続を使用する場合、ネイティブな接続が保存され、JDBC 変換は発生しません。
<u>多数の実行環境内のデータベースマルング</u>を参照してくたさい。マルングがMicrosoft Access とSQLite などのファイルベースのデータベース」接続される場合、追加構成が必要とされます。下記の「ファイルベースのデータベース」を参照してくたさい。

JDBC 接続を使用してマピングを作動することは、Java Runtime Environment おとしま Java Development Kit がサーバーマン ンニインストールされていることを必要とします。Oracle JDK おとしのenJDK などのオープンノースビルドであることができます。

- JAVA HOME 環境変数はJDK インストールディレクト」を指している必要があます。
- Windows 上では、Windows レジストリ内で検出された Java Virtual Machine パスは JAVA_HOME 変数 よ に優先順位を有します。
- JDK プラオフォーム(64-ビット、32-ビット)はMapForce Server を同じてある必要がおます。それ以外の場合は、 理由かあるエラーが挙げたれる可能性がおります。「JVM へのアクセスが不可能です。」

Linux またはmacOS 上でのJDBC 接続のセットアップ

- 1. データベースペンダーには提供される JDBC ドライバーをダウンロードして、オペレーティングシステムにインストールします。オペレー ティングシステムが82ビットで作動する場合は32ビット版バージョンを、オペレーティングシステムが64ビットで作動する場合は 64ビット版バージョンを選択してください。
- 2. 環境変数をJDBC ドライバーがインストールされている場所に設定してください。通常、CLASSPATH 変数と、必要があれば その他複数を設定します。どの特定の環境変数が構成されるかを検出するコス、JDBC ドライバーゴ付随するドキュメントを確 認してください。
- メモ macOS では、システムはJDBC ライブラが/Library/Java/Extensions ディレクトリニあることを必要とします。ですから、 JDBC ドライバーをこの場所でアンパンクすることが奨励されます。それ以外の場合、システムを構成し、JDBC ライブラが JDBC ドライバーをインストーリルたい ひを検索するようにする必要があります。

macOS 上でのOracle Instant Client 接続

macOS 上で、Oracle Database Instant Client を介して Oracle データベース 接続する場合これらの命令を適用することができます。必要条件:

- Java 8.0 おけよ 以降がインストールされている必要がおけます。Mac マシンが Java 8 以前の Java バージョンで作動作動している場合、「JDBC Thin for All Platforms」 ライブラ を介して接続し、以下の命令を無視することも可能です。
- Oracle Instant Client がインストールされている必要があます。Oracle Instant Client をOracle の公式ダウロードページからダウロードすることができます。Oracle ダウンロードページでは複数のInstant Client パッケージがあることできます。さい、Oracle Call Interface (OCI) サポトのある、シケージが選択されていることを確認してください(例えばInstant)

Client Basic)。また、オペレーティングシステムか82ビットで作動する場合は22ビット版バージョンを、オペレーティングシステムが 64ビットで作動する場合は64ビット版バージョンを選択してくたさい。

ダウンロードした後、Oracle Instant Client をアンパンし、インストーラと共にパンクされているプロン ディリストファイル(.plist)を編集すると、以下の環境変数が対応するドライバーンズをポイントします。例えば

変数	サンプル値
CLASSPATH	/opt/oracle/instantclient_11_2/ojdbc6.jar:/opt/oracle/instantcli ent_11_2/ojdbc5.jar
TNS_ADMIN	/opt/oracle/NETWORK_ADMIN
ORACLE_HOME	/opt/oracle/instantclient_11_2
DYLD_LIBRARY_PATH	/opt/oracle/instantclient_11_2
パス	<pre>\$PATH:/opt/oracle/instantclient_11_2</pre>

メモ 使用中のオペレーティングシステムで Oracle Instant Client ファイルがインストールされている場所で上記のサンプル値を編集してくたさい。

ファイルをベースユナデータベース

Microsoft Access とSQLite などのファイルをベースはたデータベースは、FlowForce Server にデプロイされる シケージ内に含まれません。まけは、エンパルされた MapForce Server 実行ファイル さ含まれません。このため、ソースとターゲトマンンが同じでは内場合、次のステップに従ってくたさい

- 1. MapForce 内では、マピングを右クリックして、「生成されたコード内でいてを絶対いてにする」のチェックボックスのチェックを解除します(マピング設定の変更を参照してくたさい。
- 2. マメング上のデータベースコンポーネントを右クリックし、相対パマを使用してデータベースファイルへの接続を追加します。ファイルを <u>ベースコンナデータベースへのパマを設定</u>を参照してくたさい。マメビングデザイン(.mfd ファイル)をデータベースファイルと同じデルク トリニ保存し、ファイル名別(相対パマを使用して)にマメピングから参照することによりパマゴ関連した問題を回避することができま す。
- 3. データベースファイルをターゲオマンス内のディレオリニピーします("作業ディレオリビビ呼ぶことします)。以下に表示されるよう にマッピングを実行する際必要とされるため、このディレオを忘れないでください。

このようなマピングをサーバー上で実行するコお以下を行います

- FlowForce Server コトロール下で作動するMapForce Server によりマピングが実行される場合、FlowForce Server ジョブが前に作成された作業ディレケリをポイントするように構成します。データベースファイルは作業ディレケリ内に存在する必要があります。
- コマボラインでスタボアロン MapForce Server によりマピンが実行される場合、MapForce Server 上でrun コマボ を呼び出す前に、現在のディレカリを作業ディレカリー変更します(例えば、cd path\to\working\directory)。
- MapForce Server API によりマナングが実行される場合、作業ディレケーをプログラング的にマナングを実行する前に設定します。これを効果的に行うために、COM と.NET API 内のMapForce Server オブジェケトのためにプロ・ディ WorkingDirectoryを使用することができます。Java API 内では、メノンド setWorkingDirectoryを使用することができます。 できます。

ローカルネ・トワーク上のWindows マンン上にノースとターゲットマンンが存在する場合、共有されるディレクトリからデータベースファイルを読み取るナムウニマッピングを以下のように構成することができます

- 1. ソースとターゲオマシンによりアクセスすることのできる共有されるディレクトリにデータベースファイルを保管します。
- 2. マッピング上のデータベースエンポーネントを右クリックし、絶対パンをしようしてデータベースファイルは接続を追加します(ファイルをベ <u>ースにしたデータベースへのいたを設定</u>)。

グロー・ジレノノース

マピングが直接の、マまけはデータベース接続の代わりにグロー・シリハノースへの参照を含む場合、サー・ドー側でもグロー・シリハノースを使用 することができます。MapForce Server 実行可能ファイル(.mfx) にマピングをエン・ペルオる場合、グロー・シリハノースへの参照は変更 されないので、マピングランタイムにサー・ドー側に提供することができます。FlowForce Server にマピングをデプロイする場合、リノースを サー・ドー上で使用するかを任意で選択することができます。

マルング(おけま FlowForce Server の場合マルンク関数)の実行に成功するために、グロー・ドルソースとして提供する実際のファイル、フォルダー、おけまデータベース接続の詳細はサーバー環境で互換性を持つ必要があります。例えば、マルングがLinux サーバーで作動する場合、パマのためにLinux 変換を使用する必要があります。同様に、データベース接続として定義されているグロー・ドルノースはサーバーマンンで利用が可能である必要があります。

詳細に関しては<u>MapForce Server 内のグロー・シリンースとFlowForce Server 内のグロー・シリンース</u>を参照してくたさい。

8.4 MapForce Server 実行可能ファイルにマッピングをコンパイルする

MapForce 内で作成されるマピングのターゲト言語がBUILT-IN に設定されている場合 MapForce のみでなく、MapForce Server もがマピングを実行することができます、次を参照してくたさい、<u>MapForce Server についつ</u>。MapForce Server を使用して2通りの実行を行うことができます。

- MapForce Server がタメアロンモードで作動している場合、(つまり、FlowForce Server がインネールされていない状態で)、マメニングは、下に表示されるとおり、サードー実行ファイル(.mfx)にコンパイルされる必要があります。その後、.mfx ファイルをコマメドラインでコマンド run を使用して実行します。run メンメを MapForce Server API から呼び出して、マメニングを実行することもできます。詳細に関しては、次を参照してください、MapForce Server ドキュメント ((https://www.altova.com/ja/documentation)。
- おは、MapForce Server がFlowForce Server 管理下で作動している場合、マンピングは、MapForce Server と FlowForce Server の双方が作動しているマンメニデプロイオることができます。このシナリオの詳細に関しては、次を参照してく ださい、マンピングをFlowForce Server にデプロイオる。

必要条件 サーバーでの実行のためこマパングを準備するを参照してくたさい。

MapForce Server Execution (.mfx) ファイルレマッピングをコンパイルする:

- 1. MapForce 内でマピングを開きます(例えば myMapping.mfd)。
- 2. 「ファイル」メニューから「MapForce Server 実行ファイルニンパル」をクルクします。
- 3. .mfx ファイルを保存するフォルダーを選択し、必要であれば名前を変更します。
- 4. 「保存」をクリックします。MapForce Server 実行ファイルmyMapping.mfx は、選択されたフォルダー内に生成されます。

MapForce Server 実行可能ファイル(.mfx) ファイルにマッピングをコマンドラインを使用してコンパイルする

• コマボラインで MapForce を実行し、マピングファイルと / COMPILE コマボラインオプロンを指定します。

例えば次のコマイは、マピング<マイドキュメント> Altova Map Force 2021 Map Force Examples Simple Total.mfdをMap Force Server 実行ファイルにエン イルレ、ターゲオ出力ディレオリ C:\Users\altova\Desktop に作成されます。

"C:\Program Files (x86)\Altova\MapForce2021\MapForce.exe" "C: \Users\altova\Documents\Altova\MapForce2021\MapForceExamples" /COMPILE "C: \Users\altova\Desktop"

以下も参照してくたさい MapForce コマ・ドラインインターフェイス。

.mfx ファイルに含まれている内容

.mfx ファイルコンのデーの含まれています:

- 他のマピングからインポートされたユーザー定義関数(UDF)を全て含むマピングアルゴリズム。
- コンポーネントから参照されている入力と出力ファイル名。マンピング設定に従い、アレン維対的ませば相対的です。多種の実行環境内の、文を参照してくたさい。
- マピングがXMLコンポーネントを含む場合、マピングの実行に必要なXMLスキーオニ関する情報はマピングアルゴリズムコエンコードされる必要があります。
- マピングがデータベース接続を含む場合にはデータベース接続の詳細。パマワードは暗号化されています。

マピングによ使用された入力インスタンスファイル(XML、CSV、テキスト)コンパルされた..mfx ファイルコは含まれません。Access おけよSQLite などのファイルベースのデータベースに関しても同様に適用されます。詳細に関してはサ<u>ーバー実行のためにマンピング参準</u> 備するを参照してくたさい。

特定のMapForce Server バージョンのためにマンピングをコンパイルする

MapForce Server がMapForce お先古いシンションの場合、前者は、新機能が搭載されていない可能性があるさめ、新しいシンションのMapForce で作成された、mfx ファイルを実行できない可能性があります。このような場合には、.mfx ファイルを特定のMapForce Server バージョンのために、以下の方法でコンパイルすることができます。

- 1. 「ツール」メニューから、「オプション」をクトックして、「生成」をクトックします。
- 2. 「サーバー実行ファイル」の下の「MapForce Server バージョンの生成」横から、必要とする MapForce Server バージョンをドロップダウノノストから選択します。

-サーバー実行ファイル ■ すべての ADO と ODBC データベース接続を JDBC へ変換(J) ▼ デジタル署名を無視 (MapForce Server でサポートされません)(G) 次の MapForce Server バージョンのために生成:	最も最近の ▼	
	ОК	キャンセル

新しいMapForce Server バージョンを取得すると このオプションを必要に応じて変更してください。MapForce Server の特定の、ー ジョンのナンリニエン・ゲルオる特定の理由が存在してい場合は、 "最新" オプションを選択します (これはデフォルトのオプションです)。 このオプ ションが選択されていると、mfx ファイルは、MapForce Server の最新の、ージョンのナンリニエン・ゲルされるナン、前の、ージョンでは使 用することのできたい最新の機能を活用することができます。

ターゲナ MapForce Server バージョンをコマイドラインで指定するコよ /MFXVERSION スイッチを利用して /COMPILE コマイドを実行します。例

```
"C:\Program Files (x86)\Altova\MapForce2021\MapForce.exe" /COMPILE /MFXVERSION:2021
```

以下も参照してくたさい MapForce コマボラインインターフェイス。

他のオプション

MapForce Server 実行可能ファイルのエンパールは、次のオプションによっても影響されます。

<i>すべてのADO とODBC データ ベースの接続を JDBC に変換 する</i>	オプションが有効化されていると、ADO とODBC データベースの接続は、JDBC ドライバーとデ ータベースコンポーネント 設定ダイアログボックス内で定義された。データベースURLを使用して、 JDBC に変換されます(次を参照してください、データベースコンポーネントの設定)。 JDBC 接続はターゲットマンンがLinux おけまmacOS サーイトーである場合明示的に使用され ます。
(MapForce Server によりサ	このオプタンはMapForce Enterprise にのみ適用することができます。 おっ、 デフォルトで有効
ポートされていない)デジタル署名	化されています。 マメングがXML デンタル署名を使用する場合、 MapForce Server が
を無視する	XML デンタル署名をサポートしてい サム、 デンタル署名に関する情報をスキップします。

これらのオプノョンをビュー、おけな更することができます

• 「ツール」メニューから、「オプション」をクリックして、「生成」をクリックします。

これらのオプションはコマボラインインターフェイスからも使用することができます。以下も参照してくたさい MapForce コマボラインインターフェイス。

8.5 マッピングを FlowForce Server にデプロイする

FlowForce Server へのマピングのデプロイはMapForce か特定のマピングによ使用されるリノースを整理し HTTP (おけ構成されている場合 HTTPS)を介して FlowForce Server が作動するマンノン マオることを意味します。 メモ「ソースマンン」 とう用語は、 MapForce がインストールされているエピューターを指し、「ターゲルマンン」とう用語は、 FlowForce Server がインストールされてい るコピューターを指します。 マピングがデプロイされると、完全機能を搭載した FlowForce Server ジョブを作成することができ、(例えば、 ジョブのケーカニンスタムトリガーを定義する、ジョブをサービスとして公開するなど)ジョブ固有の機能全てを活用することができます。

メモ 「ソースマシン」とう用語は、MapForce がインストールされているエピューターを指しており、「ターゲト・マシン」はFlowForce Server がインストールされているマシンを指しています。最もシンプルなシナルは、ソースマシンとターゲト・マシンが同じエピュータ ーの場合を指します。最も複雑なシナルオは、MapForceがWindows マシンで作動し、FlowForce Server がLinux、 およ、macOS マシンで作動する場合を指します。

FlowForce にデプロイされる シケージ お以下の通りです:

• マピング自身ユポーネトは入力引数となり、ターゲトコポーネトはこの関数の出力引数となります。全てのノースコンポーネ ントは入力引数になり、全てのターゲトコポーネトはこの関数の出力関数になります。

Execute fund	tion /public/Comple	etePO.man	nin	a	
Parameters:	Customers:	(input)	e	altova://packagedfile/C:/Users/	/MapForceExamples/Customers.xm
	Articles:	(input)	뎚	altova://packagedfile/C:/Users/	/MapForceExamples/Articles.xml
	ShortPO:	(input)	뎥	altova://packagedfile/C:/Users/	/MapForceExamples/ShortPO.xml
	CompletePO:	(output)	욚	CompletePO.xml	
	Working-directory	:		+	

• マピングによ使用される入力インスタンスファイル(XML、CSV、テキスト)。

必要条件

サーバー実行のためにマンピングを準備するを参照してくたさい。

Flow Force Server にマンプをデプロイする

- 1. MapForce を作動し、Built-in デジン・ノレトーボタンをクトック、おけま出力 | Built-in 実行エンジン メニューコマンドを選択して、変換言語がBUILT-IN に設定されていることを確認してくたさい。
- 2. 「ファイル」メニューから「FlowForce Server にデプロイ」を選択します。マメピングのデプロイダイアログボックスか開かれます。

😢 Deploy Mappir	ng	×
Enter the host name mapping.	e and port of a FlowForce Administration Interface to deploy the cur	rent
Server:	localhost V Port: 8082	
<u>U</u> ser:	root Use SS	L
Pass <u>w</u> ord:	••••	
Login:	Directly	
Global Resources	on the Server	
Use <u>R</u> esource	S	
Resource Path:	/public/	
	The path must start with a slash character.	
Deploy As		
<u>P</u> ath:	/public/CompletePO.mapping Browse	
	The path must start with a slash character.	
S <u>a</u> ve mapping) before deploying	
🗹 Open web bro	owser to create new job	
	OK Cance	:I

3. (下で説明される通りデプロイ設定を入力し [OK]を押します。「新しいジョブ作成のためにウェブブラウザを開く」 チェックボック スが選択されている場合、FlowForce Server 管理インターフェイスが開かれ、すくに FlowForce Server ジョブの作成を開 始することができます。

以下のテーブルは、マパングのデプロイダイアログボックスで使用することのできるマパングデプロイ設定をリストしています。

設定	説明
サーバー、ポート、SSL の使用	FlowForce Server のサードホスト名(おけまI.P. アドレス)とポートを入力します。これ はFlowForce Server がデフォルトのポートと同じマンで実行されている場合、 localhost と8082 である可能性があります。疑問がある場合は、FlowForce Server Web 管理インターフェイス ロングインして、Web ブラウザーのアドレスバーに表示される I.P. アドレスとポートを確認してくたさい。
	接続エラーか生じた場合、FlowForce Server か実行されているマシンか入力接続を許可するように構成されていることを確認してくたさい。
	SSL 暗号化済み接続を介してマンングをデプロイする場合 SSL の使用 チェックボックスを 選択しますこれは、FlowForce Server が既にSSL 接続を受け入れるようご構成されて いるものの想定しています。詳細に関しては、FlowForce Server ドキュント (<u>https://www.altova.com/ja/documentation</u>) を参照してくたさい。

設定	説明
ᠴ᠆ᡃᡃᠮ᠆ᡃᢣᡃᢂᠺᡃ᠋ᡗ᠆ᡟ	ユーザー名と マワードは、ログインドロップダウンリストの値(次のオプション参照)により入力されます。ログインドロップダウンリストが >Default> おけま Directly に設定されている場合、FlowForce Server ユーザー名と マワードを入力します。それ以外の場合ドメインユ ーザー名と マワードを入力し、ドメイン名をログインドロップダウンリストから選択します。
ログイン	FlowForce Server 内でDirectory Service が有効化されているとこのドロップダウン リストカンドメイン名を選択し、ユーザーと スワードフィールド にドメイン資格情報を入力します (前のオプノョンを参照してくたさい)。
リノースの使用、リノース/ SL	マメピンク関数がサーバーにデプロイ後に <u>リソース</u> を使用する場合リソースの使用 チェックボックスを選択しますチェックボックスを選択すると、「リソースパス」テキストボックスとの対応するリ ソースのパスを入力する必要があります。リソースを選択し、「省略記号」ボタンをクトックしま す。
	サーイ・上に選択するハノースが存在しない場合、「グローイジレノソースのデプロイ」をクトックして、必要とされるグローイジレノノースをサーイトーにデプロイします。詳細に関しては、 FlowForce Server にコーノスをデプロイするを参照してくたさい。
	「リソースの使用」チェックボックスを選択しない場合、グローイ シリトノースは現在選択されて しる構成をベースに解決されます。サーインー上では、マッピング関数はグローイ シリトノースを必 要とせず、代わりに解決された値を使用します。
/次	「参照」をクリックして、FlowForce Server エンテナーの階層内でマッピング関数が保存 される ひを選択します。デフォルトで ひは FlowForce Server の/public エンテナーに 設定されています。
	必要な FlowForce Server パーミッションと特権が与えられている場合、既存のコンテナー とマッピングを削除できることもできます。
デプロイ前ニマッピングを保存する	保存されていないマンビングをデプロイする時にこのオブションを使用することができます。デプロイ 前にマッピングを保存する場合このチェックボックスを選択します。
新しいジョブを作成するけっかコブラウリを開く	このチェックボックスを選択した場合、FlowForce Server Web 管理インターフェイスかデプ ロイ後開かれ、FlowForce Server ジョブの作成の開始をすることができます。

トラブルシューティング

次のテーブルはマピングをデプロイする際に発生する一般的な問題をリストしています。

問題	ソリューション
マピングのデプロイは以下のエラーを返します: ファイル上の I/O オペレーションに失敗しました。 I/O エラー 28: 〈ser ver〉ポー 8082 への接続に失 敗しました。タイムアウト システムエラー 10060: 接続されたパーティが一定の時間内に 正確!ユンスポイレなかった。または、接続されたホストがレスポイド に失敗し確立された接続が失敗したため接続の試みは失敗しまし た。	ターゲトマシン上で Flow Force Web Server サービスが作動し ており、指定されているポート(デフォルトで 8082)で接続のイングコノッ スンするよう 1構成されていることを確認してくたさい。 オ・ファイア・カー いかこのポートを介して受信される接続をブロックしたい ようしてくたさ い。 デプロイを可能にするオングニ Flow Force Server サービスも作動し ている必要かあります。
マンピングのデプロイは以下のエラーを返します	デプロイされるマッピングの入力ファイルがFlowForce Server により

問題	ソリューション
ファイル上の I / O オペレーションに失敗しました。 I / O エラー 413: ペイロードが大きすぎます	許可されているHTTP リクエストの最大量(大体 100 MB)を超える とエラーが発生する場合があります。「ミットを max_request_body_size オプションを(in bytes) flowforceweb.ini とflowforce.ini ファイル内で設定して増量 することができます。詳細に関しては FlowForce Server ドキュメン トを参照してくたさい。

サーバーバージョンの選択方法 (Windows のみ)

マルングをデプロイするサー・ーで FlowForce Server 管理下で、複数の ージョンの MapForce Server を作動している場合、 (Windows サー トーのみに適用することができます) このマルングを実行する MapForce Server の ージョンを指定するようにプロンプ トされます。

Select MapForce	Server			Х
Multiple versions of MapForce Server were found which can all execute this mapping.				
Select the most appropriate version automatically				
O Choose version <u>m</u> anually:				
Version:	2017r3	\sim		
			OK Cance	4

メモ FlowForce Server インホールディレケルに、tool ファイルは含まれて、る場合、FlowForce Server の管理下で実行され る各 MapForce Server の・ジュンのために、ダイアログボックスの表示されます。デフォルトでは、MapForce Server を FlowForce Server の一部とてインホールすると、MapForce Server tool ファイルは、このディンケルに自動的に追加 されます。FlowForce 内でファイルは保管される、ペロジェアのとおしです。C:\Program Files\Altova\FlowForceServer2021\tools。FlowForce Server 管理下での他のMapForce Server バージョン を作動する場合、、tool ファイルを手動で上記のディントリニピーする必要があります。MapForce Server の、tool ファイル は以下に保管されて、ます: C:\Program Files\Altova\MapForceServer2021。

8.6 MapForce コマンドラインインターフェイス

コマボラインでのMapForce コマボの一般的な構文:

MapForce.exe <filename> [/{target} [[<outputdir>] [/options]]]

レジェンド

コマイドライン構文を示すためこ次の表記が使用されます

表記	説明
かって無しのテキスト	アイテムは表示されるとおり入力されなければなりませんです。
〈山かこ内のテキスト〉	値を提供する必要のあるプレースやいダです。
[角かっこ内のテキスト]	任意のアイテムです。
{かこ内のテキスト}	必要とされるアイテムを設定します。1つ選択します。
垂直線()	相互的に排除するアイテムのためのセルーターです。1つ選択します。
省略記号()	繰り返すことのできるアイテム。

<filename>

コードが生成されるマピングデザイン(.mfd)、ませまマピングプロジェクト(.mfp)ファイルです。プロジェクト全体のためにコードを生成する 場合、ターゲト /GENERATE を設定して、<filename> としてプロジェクトパンを入力します(下の説明を参照してくたさい)。例えば MapForceExamples.mfp。

/{target}

コードが生成されるターゲナ言語ませま環境を指定します。次のコード生成ターゲナがサポートされます。

ターゲット	説明		
/COMPILE[:compileoptions]	MapForce Server 実行可能ファイル(.mfx) にマッピングをコンパルします。任意で、次のコンマで区切られたオプションか与えられます:		
	JDBC	JDBC ドライバーとデータベースコンポーネント設定 ダイアログボックス内で定義されたデータベースURL を 使用するJDBC への全てのデータベース接続を変換 します。 <u>データベースコンポーネント設定</u> も参照してくだ さい。	
	NOXMLSIGNATURES	MapForce Server 実行ファイル内の電子署名の生成を抑制します(MapForce Server は自動署名を サポートしなことに注意してくたさい。	
/generate	プロジェクトファイル内に全ての 定を使用して生成します。フ	アマピングのオメのプロジェクトコードを現在のフォルダー設 ロジェクトフォルダーの管理を参照してくたさい。このターゲッ	

ターゲット	説明	
	トを選択すると、MapForce プロジェクト (.mfp <filename> とて提供してくたさい。</filename>	ファイル)プロジェクトを
/XSLT	XSLT 1.0 コードを生成します。	
/XSLT2		
/XSLT3	XSLT 3.0 コードを生成します。	
/XQuery	XQuery コードを生成します。	
/JAVA	Java コードを生成します。	
/CS	C# コードを生成します。このコマンドは任意で特定のコード生成オプションの設定を許可します、具体的には	
	/CS[:{VS2010 VS2013 VS2015 VS201	.7 VS2019}]
	VS2010 Visual Studio 20	10
	VS2013 Visual Studio 20	13
	VS2015 Visual Studio 20	15
	VS2017 Visual Studio 207	17
	VS2019 Visual Studio 20	19
	Visual Studio バージョンか指定されていないと Visual Studio バージョンを使用してコードが生」 ションを参照してくたさい。	コード生成オプション内で定義された 或されます。 <u>コードジェネレーターオプ</u>
/CPP	C++ コードを生成します。このコマンドはオプション を許可します。具体的には	で特定のコード生成オプタンの設定
	/CPP[:{VS2010 VS2013 VS2015 VS20 {LIB DLL},{MFC NoMFC}])17},{MSXML XERCES3},
	最初のオプショングループはターゲット Visual Stu 値	idio バージョンを設定します。有効な
	VS2010 Visual Studio 20	10
	VS2013 Visual Studio 20	13
	VS2015 Visual Studio 207	15
	VS2017 Visual Studio 20	17
	VS2019 Visual Studio 20	19
	2番目のオプショングループは生成されたコードによ 指定します。有効な値	りターゲナとされる XML ライブラ を

ターゲット	説明	
	MSXML XERCES3	MSXML 6.0 のためにトードを生成します
	3番目のオプショングループ かを指定します。有効な値	は動的なライブラルとな対に静的なライブラル生成される
	LIB	静的な LIB ライブラ を生成します
	DLL	DLL ライブラ を生成します
	4番目のオプショングループ 指定します。有効な値	はMFC サポート有り、おけましでコードが生成されるかを
	MFC	MFC サポートを有効化します
	NoMFC	MFC サポートを無効化します
	上のオプロンか指定されて るVisual Studio バージョ 参照してくたさい。	Ω ✔21 場合、コードはコード生成オプション内で指定されてい シを使用して生成されます。 <u>コードジェネレーターオプション</u> を

<outputdir>

出力ディレケーを指定する任意の、ラメーターです。出力、マか与えられていない場合、現在の作業ディレケーが使用されます。相対的なファイルレマは、現在の作業ディレケーリニオして相対的であることに注意してください。

/GENERATE かターゲオで、<outputdir> ノラメーターか設定されていない場合、コード生成言語と各マピングの出力/マは、プロジェケトの各フォルダーの方法に定義されている設定にお提供されます。プロジェケトフォルダーの管理を参照してください。

/GENERATE かターゲオで、<outputdir> ノラメーターか設定されている場合、<outputdir> 値はコマイ・ラインで与えられ、ルー とプロジェクトレベルで定義されている出力ディレクトリよりも、優先順位からえられます。しかしなから、プロジェクト内の各フォルダーで定義 されているコード生成設定よりは高い優先順位からえられません。

/options

/options は相互排他的ではかほせん。1つ、おは複数のオブシンを設定することができます。

オプション	説明
[/MFXVERSION[: <version>]</version>	ターゲトが/COMPILE の場合このオプランを適用することができます。特定の MapForce Server の・デジョンのナジロこMapForce Server Execution (.mfx) ファイルをコン・イルします。2013r2 から現在のMapForce ノ・デジョンまでの MapForce Server の・デジョンを値として与えることができます。次も参照してくたさい :MapForce Server 実行可能ファイルニマノピングをコン・イルオる。
/GLOBALRESOURCEFILE <filename></filename>	マノピングが入力ファイル、出力ファイル、フォルダー・ス、おゴはデータベースを解決するかの にグロー・シルリノースを使用する場合、このオプションを適用することができます。詳細に関

オプション	説明
	しては、Altova グロー・ジレンースを参照してくたさい。
	オプション /GLOBALRESOURCEFILE 12プロー・シリリノース.xml ファイルへの 文を指定します。/GLOBALRESOURCEFILE カ設定されて る場合、/GLOBALRESOURCECONFIG か設定されなければよりません。
/GLOBALRESOURCECONFIG <config></config>	このオプランはグロー・ シリノース構成の名前を指定します(前のオプランも参照してく たさい。/GLOBALRESOURCEFILE か設定されている場 合、/GLOBALRESOURCECONFIG か設定されなければないません。
/LIBRARY <libname> ()</libname>	コード生成ターゲト言語と共に使用して追加関数ライブラを指定します。このオプション を、複数のライブラをロードするなっとったすることができます。 関数ライブラルの 管理も参照してくたさい。
/LOG <logfilename></logfilename>	指定された、次でログファイルを生成します。くlogfilename> はフル やとて使用することができます。例えば、ディノオ・リとファイル名の両方を含むことができます。したしながら、フル やか与えられると、ログファイルを生成するためにエグファイル約存在する必要かあります。ファイル名のみか指定されると、ファイルはWindows コマ・ドプロンプトの現在のディレノオ・リ内に保存されます。

リマーク

- 相対、以はMapForce を呼び出すアプリケーションの現在のディレケトリである作業ディレケトリニ対して相対的です。これ は、.mfd ファイル名、.mfp ファイル名、出力ディレケトリ、ログファイル名、およびグローイ ジレリノースファイル名の マレニ対して適用されます。
- コマボラインでは シクスラッシュと終わり 同符を最後に使用しないでください(例えば、 "C: \My directory \Left)。これら2つの文字はコマボラインパーサーにはに重引用符マークとして解釈されます。引用符が必要な場合、("c: \My Directory \Left")、二重の円記号 \Left を使用します。ませま スペースの使用を回避してください。

サンプル

1) MapForce を開始して、マピング <filename>.mfd を開くには、次を使用します:

MapForce.exe <filename>.mfd

2) XSLT 2.0 コードを生成し、<logfilename> とう名前のエグファイルを作成するコよ、次を使用します:

MapForce.exe <filename>.mfd /XSLT2 <outputdir> /LOG <logfilename>

3) グローイ ジレトノースファイル <grfilename> からのグローイ ジレトノース構成 <grconfigname> を考慮して XSLT 2.0 コードを生成 するコよ 次を使用します:

Mapforce.exe <filename>.mfd /XSLT2 <outputdir> /GLOBALRESOURCEFILE
<grfilename> /GLOBALRESOURCECONFIG <grconfigname>

4) Visual Studio 2015 のためこC# アプリケーションを生成し、ログファイルを出力する場合、次を使用します。

MapForce.exe <filename>.mfd /CS:VS2015 <outputdir> /LOG <logfilename>

5)「ツール | オプション」内で定義されているコード生成設定を使用し C++ アプリケーションを生成し、ログファイルを作成するコよ、次を 使用します:

MapForce.exe <filename>.mfd /CPP <outputdir> /LOG <logfilename>

6) Visual Studio 2015、MSXML、のためのC++ アプリケーションを静的なライブラノ、MFC サポート、を使用してログファイル無しで 生成するコよ、次を使用します:

MapForce.exe <filename>.mfd /CPP:VS2015,MSXML,LIB,MFC

7) Visual Studio 2015、Xerces、のかめのC++ アプリケーションを動的なライブラを使用してMFC サポート無しでログファイルと共 に生成するコよ、次を使用します:

MapForce.exe <filename>.mfd /CPP:VS2015,XERCES,DLL,NoMFC <outputdir> /LOG <logfilename>

8) Java アプリケーションを生成し、ログファイルを出力する場合、次を使用します。

MapForce.exe <filename>.mfd /JAVA <outputdir> /LOG <logfilename>

9) (ロジェクト内部の各フォルダーの)フォルダー設定内で定義されている言語と出力ディレクトリを使用してプロジェクト内に全てのマッピングの オーダのコードを生成するコス 次を使用します:

MapForce.exe <filename>.mfp /GENERATE /LOG <logfilename>

10) プロジェクトファイル内の全てのマッピングのために Java コードを生成する コよ 次を使用します:

MapForce.exe <filename>.mfp /JAVA /LOG <logfilename>

フォルダー設定内で定義されているコード生成言語は無視され、全てのマンピングのためにJava が使用されることに注意してくたさい。

11) 以前にコンパルされた Java マピングのためにコマボラインで入力と出力ファイルを提供するには、次を使用します。

java -jar <mappingfile>.jar /InputFileName <inputfilename> /OutputFileName <outputfilename>

/InputFileName と/OutputFileName パラメーターはコマボラインの実行内で、ラメーターの使用を許可するMapForce マッピング内の特別な入力コンポーネントの名前です(マピングロンラメーターを与えるを参照してくたさい)。

12) MapForce Server バージョン 2021 のために MapForce Server 実行ファイルにマンピングをエンパイルし、XML 署名を抑制するコよ 次を使用します:

MapForce.exe <filename>.mfd /COMPILE:NOXMLSIGNATURES <outputdir> /MFXVERSION:2021 /LOG
<logfilename>

9 Altova グローバルリソース

Altova グロー・シリソースはファイル、フォルダー、まけはデータベースへのポータブルなレフォレンスです。グロー・シリソースとして保管されると、 ハマとデータベース接続の詳細は再利用できるようこなり Altova アプリケーション全体で使用可能にないます。例えば、複数の Altova デス クトップアプリケーション内で同じファイルを頻繁に開く必要かある場合、グロー・シリソノースとして定義すると便利な場合かあります。このようこ すると、「ファイルを開く」ダイアログボックスから対応するグロー・シリノノースを開くことかできるけっな、ファイルル 文を覚えてい る必要かありません。こ の方法はファイルの 文か変更されると、ハマの一部を変更するのみという利点かあります。

グロー・シリソースの一般的な使用法は、データベース接続を一度定義し、グロー・シリソースをサポートする全てのAltova アプリケーション で再利用することです。例えば、MapForce マメピングがデザインされるマシンでデータベース接続を作成し、MapForce Server がマメピ ングを実行するマシン上で同じ接続を再利用することができます(これは、一部の場合、両方のマシントコロじデータベースクライアント・ソストウェ アがインストールされている場合があります)。

任意で、(「構成」として既知の同じグロー・シリンノースの複数の・シエーションを作成することができます。これによりファイル、おけるフォルダ ー・マ、(おけまデータベース)を必要に応じて変更することができます。例えば、「開発」と「生産」の構成を持つ「データベース」リンースを作成 することができます。MapForceを例に挙げると、マピングをプレビューする前に、希望する構成をリストから選択して、開発、おけな生産デ ータベースからデータを簡単に抽出することができます。

次のAltova デスケップアプリケーションからグロー・シリレノーズを作成することができます:Altova Authentic、DatabaseSpy、 MobileTogether Designer、MapForce、StyleVision、およびXMLSpy。サーバー製品では、次のAltova サーバーアプリケー ションでグロー・シリノノーズを作成することができます:FlowForce Server、MapForce Server、RaptorXML Server、 RaptorXML+XBRL Server。

グロー・ジリノース は 多種のシナけのためにMapForce 内で使用することができます:In MapForce 内では、グロー・ジリノーズファ イル、フォルダー、おさよ データベース参照などを多種のシナリオのために使用することができます。例

- マッピングの入力とて構成することのできるファイルレンを宛てる場合。次を参照してくたさい。
 <u>例:変数入力ファイルを持つマッピン</u>
 <u>グを実行する</u>。
- マピングの出力を構成することのできる。マリコダイレクトする詳細に関しては、次を参照してくたさい<u>例、出力を変数フォルダに出力する</u>。
- マナングで使用される場合、構成することのできる やを Style Vision Power Stylesheet (.sps) ファイルに与える場合。
 例えば、Style Vision .sps ファイルを MapForce コンポーネント 設定から参照する代わりに、グロー・ ジレリノースとして定義されている、2つの構成を持つことのできる.sps ファイル(例えば、「Website.sps」と「Print sps」)を使用することができます。
 次も参照してくたさい、Style Visionを使用してマナング出力のスタイルを指定する。
- データベース接続を再利用する場合。(Altova アプリケーション内でデータベースが既にグロー・ シリノノースとして定義されている場合、セナアップのステップをもう一度行う必要なく、接続を構築することができます。次を参照してくたさい、グロー・ シリノノースから接続を使用する。
- マッピングがデータを読み取るデータベースを切り替える、おさオマングデータを書き込むデータベースを簡単に切り替えることができます(これは、2つ以上のデータベースが同じ構成を持ち、異なるデータを含むことが前提とされます。次を参照してくたさい。例 データベースの切り構え)。

• MapForce Basic Edition は グロー・ ジリノースとして定義されるデータベース接続をサポートしません。

9.1 グローバルリソースの作成

グロー・シリノースはファイル、おけはフォルダー・シス、おけはデータベース接続を表す再利用可能のレファレンスです。グロー・シリノースは一度 定義されると、複数のAltova アプリケーション全般でサポートするコンテキスト内で必要な回数再利用することができます。データベースを例 に挙げると、1つ以上のAltova アプリケーションで特定のデータベースと共に頻繁に作業する場合、グロー・シリノノースとしてデータベース接続 を追加することはは、アイデアです。このようにすると、他のAltova アプリケーションから同じデータベースは続する必要かある都度データベー ス接続ウィザードのステップを実行する必要かありません。

各グロー シリソースはサカシ「構成」を持つことができます。構成によりAltovaアプリケーションにより使用、おけお作成されるファイル、フォ ルダー、およびデータベース間を簡単にむい替えることができます。これはテストのシナリオで特に役に立ちます。例えば、異なるドライバーの型 を持つ同じデータベースは接続する以下の3つの個別の接続から構成されるデータベースリソースを作成することができます:(a)デフォルトの接 続型である ODBC。(b) JDBC、と(c) ADO.NET。このようして、特定のドライバーを持つデータベースは接続するには、グロー・シリノ ソースドロップダウノリストから対応する構成を選択するだけです。



グロー・ ジレトノース ドロップダウンリスト

ボタンクトックのみで、構成は変数フォルダーニマンピング出力を生成することができます。例えば、フォルダー・ハースを2つの構成から作成する ことができます: (a) ディレクトリ C:\Testing を指す「テスト」 (b) ディレクトリ C:\Production を指す「生産」。 グロー・ シルトノースドロッ プダウンリストから必要とする構成をマンピングの実行の前に選択し C:\Testing お日ま C:\Production フォルダー「こ出力を生成する マンピングを構成することができます。 サンプル 変数フォルダーに出力を生成する

グロー・ ジレノソースを作成する方法:

- 1. 「ツール」メニューカジグロー・バルリソース」をクリックします。(おけよ「グロー・バルリソースの管理」 🛃 ツール いちがくをクリックします。)
- 2. 「追加」をクリックして、(ファイル、フォルダー、データベースなどの作成するリノースの型を選択します。
- 3. 「リソースエイリアス」テキストボックス内に詳細名を入力します(例えば、 "InputFile"、"OutputFolder"、"DatabaseConnection").
- 4. 「Default」構成をセオアップします:
 - a) ファイル、おイコンオルダーの場合、デフォルトでポイントされるこのリノースへのファイル、おイコンオルダーを参照します。
 - b) データベース接続の場合、「データベースの選択」をクリングして、データベース接続ウィザードに従い、データベースは接続します(データベースへ接続ください)。(グロー・ シリノノースドロップダウノノストから他の構成が明示的に選択されている場合、または、サーバー実行内でコマイドライン・ラメーターとして提供されていたい場合以外)このデータベース接続はマンピングの実行の際にデフォリトで使用されます。
- 5. オプーコンとして、リノースに追加構成が存在する場合(例えばデータベースの場合のドライバーの種類、おけよファイルとフォルダ ーの場合代替パンなどの「構成の追加」 ポオジェをクリックして、(例えば "ProductionFolder" or
 - "JDBC_Alternative"など詳細名を入力し、次にようにセオアップします:
 - a) ファイル、ませまフォルダーの場合、前のステップ内で定義済みのデフォルトの構成の代替としてこのハースによけ代イトされる ファイル、ませまフォルダーを参照します。
 - b) データベース接続の場合、データベースは接続するためにデータベース接続ウイザードに従います。このデータベース接続はデ フォルトの代替とて使用されます。

一部の場合、デフォルトの構成のエピーとして構成を作成し、編集するほか便利な場合があります。この場合、「現在選択されている構成のコピーとして構成を追加する」 2011 ポタンをクリックします。

6. 必要とされる各追加構成のためこ前のステップを繰り返します。

9.2 グローバルリソースとしてのデータベース

データベース接続をグローイ ジルノノースとして追加する場合、接続ウィザードは接続をセットアップするステップを導きます。<u>データベース接続ウィ サードの開始</u>を参照してくたさい。ウィザードが完了すると、データベース接続ノラメーター はグローイ ジルノノースダイアログボックス上に表示され ます。

🕑 グローバル リソー:	ス				×
「エイリアス名					
	SourceDatabase				
99 = X 1197 X(A):	SourceDalabase				
構成					
1. J. ()	- 構成 "Default"の言	设定			
Detault ReleaseDatabase	見データベース	の選択			
	ポー カペーフ(p),				
) = 3/(= 7(D):				
	★ ± N2 接続文字列		Data Source=C:¥Altova¥Altova¥altova.mdb:Provider=Microsoft.1et.OLEDB.4.0		^
	ルートオブジェクト				
	データベースの種類		MS Access		=
	インポートの種類	<u></u>	ADO		
	テータベースハージ	3) 	4		
	データソース		C:¥Altova¥Altova¥altova.mdb		
	ユーザー ID				
	パスワード				
	プロバイダー		Microsoft. Jet. OLEDB. 4.0		
	セキュリティー 音報加小	子住します			•
	MapForce 固有0	D実行パラメーター <mark>(</mark> №	1)		
	∇				
	DataSource	C:#Altova#Altova#alto	ova.mdb		
	Provider	Microsoft.Jet.OLEDB.4	1.0		
	JDBCDatabaseURL	jdbc:odbc:;DRIVER=M	licrosoft Access Driver (*.mdb);		
	JDBCDriver	sun.jdbc.odbc.JdbcOd	lbcDriver		
	UserID				
	OtherProperties	Provider-Microsoft 1e	t OLEDB 4 0.Data Source-C:¥Altova¥Altova¥altova mdb:		
	Outerroperaes	Trovider			
	J				
				ОК	キャンセル

グロー・シリノースダイアログボックスでは、データベース接続ノラメーターの一部を編集することができます。上に表示されるように、ノラメーター12つのカテコリにグループ化されています:

テ ータべ ー ス	これらの デメーターは、Altova アプリケーション間で共有されます。 MapForce 内では、デザイン時に使用され、マパングがロードされると、おけは マパングをプレビューするけっかこ MapForce 内で「出力」 タブをクリックすると 共有することができます。
MapForce 固有の実行 パラメーター	これらの デメーターは プログラムコードを生成した場合、 おっよ マピングを MapForce Server 実行ファイル(.mfx) にコンパイルした場合に適用されま す。 マンピングのランタイムで以下のようご使用されます: ・ 生成された C++, C#、おっ」は Java プログラムコード内。

 マピングをMapForce Server 実行ファイルにてい イルした場合、 そして自動のJDBC 変換が実行された場合。自動のJDBC 変 換に関する詳細に関しては、次を参照してくたさい<u>多種の実行環境</u> 内のデータベースマパング。

マピングがジロー、シリソースを使用して、データベースは接続する場合、グロー、シリソースダイアログボックス内で表示されるデータベース接続の詳細がマピング上で定義されている詳細より比優先順位が高くなります。下で示される、エンポーネント設定 ダイアログボックス内では、データベース接続設定が灰色設定していていることで注し、、ダイアログボックスは、接続/ ラメーターがジロー、シリソースとして設定されていることを通知します。

コンポーネント設定	:
データベース データソース: 接続名: データベースの種類: 接続文字列:	C:¥Altova¥Altova¥altova.mdb 変更(C) SourceDatabase MS Access (ADO) Data Source=C:¥Altova¥Altova¥altova.mdb;
パラメーターはグローパ ログイン設定 ユーザー(U):	マートロントロントロートロートロートロートロートロートロートロートロートロートロートロートロート
パスワード <mark>(P)</mark> :	
JDBC 設定	
JDBCドライバー(J):	sun.jdbc.odbc.JdbcOdbcDriver
データベースURL(L):	jdbc:odbc:;DRIVER=Microsoft Access Driver (*.mdb);
ADO/OLEDB 設定	
データソース <mark>(D)</mark> :	C:¥Altova¥Altova¥altova.mdb
カタログ <mark>(A)</mark> :	altova
プロバイダー(V):	Microsoft.Jet.OLEDB.4.0
追加オプション(0):	Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:¥Altova¥
生成設定 トランザクションを テーブル名からス	·使用(T) キーマ名を除外する(S)
ステートメント実行の)タイムアウト
タイムアウト <mark>(M)</mark> :	60 秒 🕅 無限(I)
	ок キャンセル

コンポーネント設定 ダイアログボックス

データベースに直接接続するけっかに、(グロー・ ジリノノースを使用せず)コデータベースのエレポーホイを変更するには、「変更」をクリックして、データベースに接続するけっかにつんザードのステップを継続してくけざい。

9.3 グローバルリソースとしての MapForce と StyleVision 変換結果

静的なファイルをポイトする代われて、MapForce マピング 封まなStyleVision 変換よれ作成された特別なファイルを読み取るグローバ ルリノースを作成することができますこの場合、グロー ジルノースを使用する Altova アプリケーションは、最初に MapForce、封まな StyleVision を呼び出し、対応するマピング封まま変換を実行し結果ファイルを取得します。これにより、Altova アプリケーション間のデー タワークフローを定義することができます。(例えば、MapForce マピング 封まは StyleVision 変換の結果を入力として、他のマピングま たま変換についてする場合など)。例えば、グロー ジルノースを使用して、XMLSpy がMapForce マピングの結果(出力)を使用する 方法に関しては、次を参照してくたさい、例、アプリケーションワークフローを作成する。

マピングの結果(出力)をグロー・ゾリソースとして使用できるようにするためには、マピングの変換言語をBUILTHNに設定するか、またはマピングがBUILTHN言語によりサポートされるエレポーネトを含む必要があります。

上記のフークフローは同じエピューターにインストール済みのAltova デスケトップアプレケーション間で意味があります。MapForce と Style Vision 変換の結果をグロー・ブレリノースとして Altova サーバー製品とMapForce Basic Edition 内で使用することはできません。

Altova サーバーアプリケーションかどのようにグローイ ジリリノースを使用方法の詳細は、<u>MapForce Server 内のグローイ ジリノノース</u>と FlowForce Server 内のグローイ ジリノノースを参照してくたさい。

9.4 グローバルリソース XML ファイル

デフォルトでは、すべてのグロー・ジルリノースはAltova アプリケーションの作成場所に関わらず、以下の、なご保存されます: C: \Users\<username>\Documents\Altova\GlobalResources.xml。これには簡単に、ジグアップすることができ、Altova 製 品がインストールされている他のフークステーションに対してポーダブルゴンはます。GlobalResources.xml ファイルの名前を変更、おこ は、複製することができ、複数のグロー・ジルノノースファイルを作成することができます。しかしたから、Altova アプリケーション内では、つのグロ ー・ジルノースファイルのみんカアクティブ化することはできません。

他のAltova アプケーションとは異なり、FlowForce Server はグロー・シリトノースファイルを必要とません。代わりにトノースは他の FlowForce 構成データと同様に管理されます(アクセスパーミッションと共に再利用することが可能です)。詳細に関しては、 FlowForce Server 内のグロー・シリトノーズを参照してくたさい。

アクティブなグロー・バリリソースファイルをセットアップする方法:

- 1. 「ツール」メニューから「グロー・バルノソース」をクリックします。(または、グロー・バルノソース 🛃 ツール いちょうをクリックします。)
- 2. 「参照」をクリックして、必要とされるグロー・シリリノースXML ファイルを選択します。

複数のグロー・ シリノースファイルを使用している場合、現在アクティブなファイルに現在のエンテキストによい必要とされているすべてのグロー・、 ルリノースファイルが含まれていることを確認してくたさい。例えば、グロー・ シリノノースを使用する、 マカらデータを読み取るようにマンピングが構成されている場合、現在アクティブなグロー・ シリノノースファイルコはその特定のグロー・ シリノノースか含まれている必要があります。 それ以外の場合、「グロー・ シリノノースを解決する際にエラーカ発生しました」などのエラーメッセージが「メッセージ」 ウィンドウェ表示されます。

9.5 多種の実行環境内でのグローバルリソース

9.5.1 SLT、XSLT2、XQuery 内のグローバルリソース

XSLT、XSLT2、XQuery 内のグロー・ シレノノース

XSLT おけはXSLT2 コードを生成し、とマンビングがジロー・シリソースを使用する場合、生成されたXSLT スタイルシートに影響はありません。グロー・シリソースの有無に関わらず、入力と出力ファイルは永続的に割り当てられず、XSLT プロセッサー内でXSLT スタイルシートを開始すると柔軟に指定することができます。同様が生成されたXQuery コードに適用することができます。

RaptorXML 実行にお生成された DoTransform.bat ファイルのみが例外です。マメピングにお使用されたグロー・ ジリソースは、グロー・ ジリソースドロップダウンリストから現在選択されている値 (構成)を考慮して、DoTransform.bat 内の実際の マレンガして解決されます。

Default 🔹 🛃 🖕

RaptorXML ヘグローバルリソースを適用するための情報は、RaptorXMLドキュメントを参照してください (https://www.altova.com/ja/documentation.html)。

9.5.2 C++、C#、Java 内のグローバルリソース

C#、C++、おけよJava プログラムコードを生成する場合、マンピングにより使用されるグロー・シリハースは解決されます。例えば、グロー・バ リリハースとして定義されているファイルまけは、フォルダーエイリアスは実際のファイル、おけよフォルダー・シスプ変換されます。特定のグロー・シリノ ソース構成かグロー・シリハースドロップダウノノストから選択された場合、選択された構成のためにコードが生成されます。「メッセージ」ウィ ンドウよ どのようにグロー・シリハースが解決された課についての情報を提供します。例えば

Messages	×
CompletePO.mfd: Mapping validation successful - 0 error(s), 0 warning(s)	\sim
CompletePO.mfd: Code generation completed successfully 0 error(s), 0 warning(s)	
👌 🚰 CompletePO:: Resolved global resource 'altova://folder_resource/CodeGeneration/CompletePO.xml' to 'C:\codegen\CompletePO.xml'	
	\sim

特定のグロー・シリノース構成のナダのコードを生成するコよコードを生成する前に、グロー・シリノノースドロップダウノノストから選択します。 おけよコマイドラインからコードを生成する場合、コマイドラインでGLOBALRESOURCEFILEと

GLOBALRESOURCECONFIG パラメーターを与えます。(次も参照してくけさい MapForce コマ・ドライン インターフェイス)。

グロー・ シリノースを生成されたコードから切り替え、おける気はてきません(代わりに、入力、おけは出力 ファイル のを変更するためについてを変更することができます)。

メモ C# おけよ Java 内では、入力、おけは出力の なだけでは、データ型も変更することができます。次を参照してくたさい、マンピング 入力/出力(C#、Java)のデータ型の変更。

9.5.3 MapForce Server 内のグローバルリソース

MapForce Server 実行可能ファイル(.mfx) にマンピングをエン イルすると、使用されるグロー・ シレノレースレファレンスは解決されず保管 されます。これはマンピングの実行に成功するためにサー・「---側」これらのレファレンスを提供する必要があることを意味します。具体的には、 MapForce Server 内では、グロー・ シレノノースを使用する.mfx ファイルを実行するためによりに下が必要とされています:

1. グロー・ジリノースが定義されているファイル。デフォルトでは、すべてのグロー・ジリノースは、MapForce がインストールされているマンン上のDocuments\Altova フォルダー内のGlobalResources.xml とらファイル内で定義されています。
MapForce Server が作動するマンノニのファイルをエピーして、必要に応じて複数のこのようなファイルを作成することができます。グロー・バリノース XML ファイルを参照してくたさい。

2. グロー・ シリソース構成名。各グロー・ シリソースは、構成名により識別されます。 デフォルト名は、「デフォルト」ですが、必要に応じて追加の構成を作成することができます。 グロー・ シリソースの作成 て説明されているとおり「デフォルト」以外の希望する構成 名を指定する必要があります。

MapForce 内では、グロー・シレノノースファイル マとグロー・シレノノース構成名は、グラフィカルなユーザーインターフェイスから変更、および、 設定することができます。 MapForce Server 内では、これらはマッピングランタイムに設定されます。

• コマドラインインターフェイスを使用してマピングを実行すると、run コマドの後、オプタン -globalresourceconfig と--globalresourcefile を設定します。例

```
C:\Program Files (x86)\Altova\MapForceServer2021\bin\MapForceServer.exe run
SomeMapping.mfx --globalresourcefile="C:
\Users\me\Documents\Altova\GlobalResources.xml" --globalresourceconfig="Default"
```

 MapForce Server API を使用してマピングを実行する場合、Run メンドを呼び出す前に、メンド SetOptions を2回 呼び出してくたさい。最初の呼び出しはオプションとしてグロー、シリソースXML ファイルレンを提供するために必要です。2番目の呼び出しはグロー、シリソース構成名を提供するために必要です。

詳細は見してはMapForce Serverドキュメントを参照してくたさい(<u>https://www.altova.com/ja/documentation.html</u>)。

9.5.4 FlowForce Server 内のグローバルリソース

FlowForce Server の観点からよ デスケップアプケーション同様コンのXML ファイルコ保管されない ケム、グロー・シルソースはある 程度グロー・シルマはなくないます。 FlowForce 内では、各リノースファイルはコムフォルダー・ス、おコムデータベース接続詳細を含む可能性 のあるは再利用可能なオブジェクトです。 リノースをエピー、エクスポート、おコムインポートすることができ、他のFlowForce Server オブジェ クト同様にコロビューザーアクセスの対象となることができます。 つまり、 FlowForce ユーザーはマノピング関数内のすべてのノノースを使用する ことができます。

MapForce 内でグロー・ シリシースを使用するマンピングを作成すると、FlowForce Server にデプロイすることができます。 デプロイ時に デプロイメトダイアログボックス上の「リソースの使用」 チェックボックスを選択することことり、マンピングがグロー シリシノースを使用するかを選 択することができます。 チェックボックスを選択しない場合、マンピングにより使用されるグロー・ シリシノースは現在選択されている構成をベースに 解決されます。 チェックボックスを選択すると、FlowForce Server 内でもマンピング関数はシノースを必要とします。 次のマンピング関数のサ ンプルはシノースの実行を必要とする FlowForce にデプロイされます。 (最初の) デメーターがシノースファイルからデフォルトのファイル マを取 得することご注意してください):

以下に関数	ReadJ	SON.mappin	g を作	≡成 /public	
関数入力パラメー	ター				
名前: People	(入力) 🚥 型:	文字列	デフォルト:	altova://file_resource/SourceFile	0
名前: Text file	(出力) 目型:	文字列	デフォルト:	Text file.csv	
名前: Working-directory	型:	ディレクトリとしての文字列	デフォルト:		
リソース					
リソースを使用した実行器	斟数: /public/GI	obalResourcesForDocu_Default.	resources		• 🗗

上て説明されているとおりFlowForce Server 内では、ジョブではよくマルンク関数がジロー・シリソースを使用します。例えば、マルン グが、マカら読み取り、おさよ、ノマへ書き込むように構成されている場合が挙げれます。上のサンプルでは、マルンク関数はリノースから最初の入力ファイルを 読み取ります。ジョブ構成ページからの、マをオー・トライドしない限り、これにはの関数を使用するすべてのジョブが同じ ノマを使用することを意味します。

FlowForce Server にクロー・ シリソースをスタイドアロンオブジェクトととしてデプロイすることができます。 つまり、 クロー・ シリソースをデプロイ するためにマンピングを最初にデプロイする必要にありません。 詳細に関しては <u>FlowForce Server に リノースをデプロイする</u>を参照してくだ さい。

FlowForce Server 内でのハノースの使用に関する詳細におしいForce Server ドキュメントを参照してくたさい (<u>https://www.altova.com/ja/documentation.html</u>).

9.5.5 FlowForce Server にリソースをデプロイする

MapForce で作成されたグロー・シリノースをFlowForce Server にデプロイすることができます。デプロイの際には、サー・トーにデプロイされるノノースを持つ構成を選択する必要があります。サー・トートで同じグロー・シリノースのすべての構成が必要な場合、複数回デプロイし、デプロイの都度必要な構成を選択することができます。サー・トートで各グロー・シリノースの名前を変更し、サー・トートの目的コンテナーを 選択することができます。

Altova アプリケーションを作動するグロー・シリノノースはサー・、環境ではサポートされていません。詳細に関しては、グロー・シリノノース <u>とてのMapForceとStyleVision変換結果</u>を参照してくたさい。

FlowForce Server にグロー・シリレースをマッピングのデプロイ時、おけよ、個別にデプロイすることができます。

Flow Force Server にグロー・バリノノースをデプロイする方法:

- 1. 「ツール」メニューカら、「グロー・バルリソース」をクリックします。(おけよ、「グロー・バルリソースの管理」 🛃 ツール デーボタンをク リックします。)
- 2. 「サーバーにデプロイする」をクトバします。ダイアログボックスを開きます。

😢 グローバルリソース	への構成のデプロイ ン	(
グローバルリソースを 力してください。	デプロイするために、FlowForce 管理インターフェイスのホスト名とポートを入	
サーバー <mark>(</mark> S):	localhost 〜 ポート(O): 8082	
ユーザー <mark>(U)</mark> :	root	
パスワード <mark>(w)</mark> :	••••	
ログイン (L) :	<デフォルト> 〜	
「グローバルリソース・		
構成(C):	Default \checkmark	
デプロイ		
パス(P):	public/GlobalResourcesForDocu_Default.resources 参照(B)	
	パスはスラッシュから開始する必要があります。	
	OK キャンセル	

- FlowForce Server (サー、・、ポー、ユーザー、ノマワード、ログインメンボ)への接続詳細を入力します。これらの、ラメーター ーはマメピングをデプロイする際に必要とされる、ラメーターと同じです。 FlowForce Server にマメピングをデプロイする を参照してくたさい。
- 4. 「構成」リストから構成を選択します。このリストにはグロー・シリトノースファイルからのすべての構成か含まれています。

グロー・ シリノース構成を一度にデプロイすることができます。上記の通り、全ての構成がサー・、一上で必要な場合、異なる名前を持つ同じリノースを複数回デプロイすることができます。

- 5. サーバー上でハースが保存されるターゲオハマを選択してください。「参照」をクリックして、ターゲオ FlowForce コンテナーを 選択、おけお必要な場合新規を作成することのできるダイアログボックスを表示します。
- 6. 「OK」をクリックします。

FlowForce Server へのグロー・ シリノースデプロイはメッセージボックスに報告されます。

9.6 例: 変数入力ファイルを持つマッピングを実行する

入力とてXML ファイルを取るマピングを頻繁に実行すると仮定します。入力 XML を変更する必要がある場合、ソースXML コンポーネントのプロ、ディを編集し、新規の入力ファイルを参照することができます。マピングの入力 XML を変更する場合、ソースXML コンポーネントのプロ、ディを開き、新規の入力ファイルを参照します。次を参照してくたさい、<u>コンポーネント設定の変更</u>。これは1度きりのタスクの場合、簡単に達成することができます。しかしながら、マピングの入力 XML ファイルを一日に、封さは毎時に、複数回実行する場合、例えば、XML ファイルをマピングの入力として使用し、レポートを生成するためにマンピングを実行する場合、他のXML ファイルから同じレポートを毎晩作成する場合などが挙げられます。

このようは場合、グロー・シリノノースを使用することができます。マノピングを複数回編集する代わりに(おけよ、複数のコピーを保管する代わり に)グロー・シリノーズ、おけよ「ファイルエイリアス」)として定義されているファイルから読み取りマノピングを構成することができます。このサンプル で挙げられる必要条件を満たすけは、ファイルエイリアスか2つの構成を持つように構成することができます。

- 1. 「デフォルトの」-この構成は「morning」 XML ファイルをマピングの入力とて提供します。
- 2. 「Evening Reports」 この構成は「evening」 XML ファイルをマピングの入力とて提供します。

構成を正しく設定することことり、これらの入力ファイルを使用してマンピングを実行することができます。ファイルエイリアスセットアップされると、下で示されるとおり、マンピングを実行する前に、希望する構成をドロップダウンリストから選択することができます。

ステップ 1: グロー・ シレノソースを作成する

ファイルエイリアスを以下のとおり作成することができます。

- 1. 「ツール」メニューカら「グロー・バルルノース」をクリックします。(おけよ グロー・バルノソース 🛃 ツール いちかんをクリックします)。
- 2. 「追加 | ファイル」をクリックします。
- 3. 「リソースエイリアス」テキストボックスに名前を入力します(このサンプルでは、「Daily Reports」が適切な名前です)。
- 4. 「参照」をクトックして、次のファイルを選択します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\mf-ExpReport.xml.
- 5. 「ファイルの追加」 📩 をクリックして、「Evening Reports」と ら名前を与えます。
- 6. 「参照」をクリックして、今度は、次のファイルを選択します: <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\mf-ExpReport2.xml.

ステップ 2: マピング内のグロー・ ジレノノースを使用する

必要とされるグロー・ ジリノースか作成されましたが、マノピングを使用することはまだできません。前に定義されたファイルエイリアス(グロー・ バ リリノース)から読み込みマノピングを変更すること、以下を行います:

- 1. 次のマピングを開きます <マイドキュメント > ¥Altova Wap Force 2021 Wap Force Examples ¥Tutorial ¥Tut-ExpReport.mfd.
- 2. マピング上のノースコンポーネントのヘッダーを右クリックして、コンテキストメニューから「プロパティ」を選択します。
- 3. 入力 XML ファイルの横の「参照」をクルクします。
- 4. 「グローバリリノースに切り替える」をクリックして、前に定義されたファイルエイリアス「Daily Reports」を選択します。
- 5. 「開く」をクトックします。入力 XML ファイル マは、グロー・ ジルハースを使用する マを示す altova://file resource/DailyReports になりました。

🕑 コンポーネント設定	×
コンポーネント名: mf-ExpReport	
スキーマ ファイル(F)	
mf-ExpReport.xsd	参照(W) 編集(T)
入力 XML ファイル(I)	
altova://file_resource/DailyReports	参照(B) 編集(E)
出力 XML ファイル(0)	
mf-ExpReport.xml	参照(R) 編集(D)

ステップ 3: 希望する構成を使用してマメピングを実行する

マピングを実行するまえに、入力 XML ファイルを以下のように簡単に切り替えることができます。

- ファイルmf-ExpReport.xmlを入力とて使用するコは「ツール」メニューから「アクティブな構成 | デフォルト」をクリックします。
- ファイルmf-ExpReport2.xmlを入力とて使用するゴは「ツール」メニューから「アクティブな構成 | Evening Reports」をクリックします。,

おけよ、グローバリリソースドロップダウノリストから必要とされる構成を選択します。

Default 💌 🛃 🖕

マピングの結果を構成を使用してパビューするコよ「出力」タブをクリックして、生成された出力の差分を確認してくたさい。

9.7 例: 出力を変数フォルダに出力する

このサンプルはグロー・ジリノノースを使用して、異なるフォルダーニマッピングの出力をダイレクトする方法について説明しています。

マング出力をディレクトリ(例えば、C:\Testing)に生成する場合、おけ出力が他のディレクトリ(例えば、C:\Production)に生成する場合があるとます。グロー・バリノノースを使用すると、2つの構成を持つフォルダーエイリアスを作成することが可能になります。

- 1. 「デフォルトの」構成 出力を以下の場所に生成します: C:\Testing
- 2. 「Production」構成 出力を以下の場所に生成します: C:\Production.

この目的を達成するためにステップは以下のとおりです。

ステップ 1: グロー・ ジレノソースを作成する

フォルダーエイリアスを以下のとおり作成することができます。

- 1. 「ツール」メニューカら「グロー・バルルノース」をクリックします。(おさよ グロー・バルノソース 🛃 ツール いちかえをクリックします)。
- 2. 「追加 | フォルダー」をクトックします。
- 3. 「リソースエイリアス」テキストボックスに名前を入力します(このサンプルでは、「OutputDirectory」は適切な名前です)。
- 4. 「参照」をクトックして、次のフォルダーを選択します: C:\Testing。(このフォルダが使用中のオペレーティングシステムで既存であることを確認してくたさい)。
- 5. 「ファイルの追加」 🤹 をクックして、新しい構成のために名前を入力します(このサンプルでは、「ProductionDirectory」)。
- 6. 「参照」をクトックして、次のフォルダーを選択します次のフォルダー C:\Production (このフォルダが使用中のオペレーティングシ ステムで既存であることを確認してくたさい)。

ステップ 2: マッピング内のグロー・ ジレノノースを使用する

必要とされる グロー・ シリノノースか作成されました。 しかしなから、 マッピングをまだ使用することはできません。 前に定義されたフォルダーエイリ アス (グロー・ シリノノース) から使用するよう こマッピングを変更する コよ 以下を行います:

- 1. 次のマピングを開きます <マイドキュメント > ¥Altova Wap Force2021 Wap ForceExamples ¥Tutorial ¥**Tut-ExpReport.mfd**.
- 2. マシピング上のターゲナコンポーネントを右クリックし、コンテキストメニューから「プロパティ」を選択します。
- 3. 出力 XML ファイルの横の「参照」をクルクします。
- 4. 「グローバリリソースに切り替える」をクリックして、「保存」をクリックします。
- 5. 出力 XML ファイルを保存するよう コンコンプトされると、output.xml (おけま 他の出力ファイルに与える詳細なファイル名)を入 カします。出力 XML ファイル マは、グロー・ジルリンースとして定義される マを示す altova://folder resource/OutputDirectory/output.xml 」ないます。

ステップ 3: 希望する構成を使用してマメピングを実行する

マピングを実行する前に、希望するマピング出カフォルダーを簡単に切り替えることが以下のように行うことができます。

- 「ツール」メニューから「アクティブな構成 | デフォルト」をクリックします。マレングの結果をパビューするけっか」「出力」 タブを クリックします。マレング出力(下で説明されているとおり、一時、おけお泳続ファイル)がC:\Testing ディレクトリニ出力されま す。
- 「ツール」メニューからアクティブな構成 | Production Directory をクリックして、「出力」 ダブをクリックします。マノピング出力(下で説明されているとおり、一時、ませまが続ファイル)は、C:\Production ディレクトリ内に生成されます。

メモ MapForceを構成して、出力が永続ファイルに書き込まれるように設定されている以外は、マピング出力は、デフォルトで、一時ファイルとして書き込まれます。

MapForce に一時ファイルの代わりに永続ファイルを生成するように構成するコよ、以下を行います:

- 1. 「ツール」メニューから「オプション」をクリックします。
- 2. 「全般」セクションで、オプション「最終出力ファイルニ直接書き込む」を選択します。

9.8 例: データベースの切り替え

データベースからデータを読み取り、データベースにデータを書き込む場合、データベース接続をマンピングのランタイムの前に切り替えることができます。(例えば、リリースから生産データベースに、おけまその逆を行うことができます)。このサンプルは、この目的をグロー・シリハノースを使用する ことにより達成します。データベースの切り替えは、同じ構造を持つ両方のデータベースが異なるデータを含んでいることを意味します。このサン プルの目的のために、次のMicrosoft Access データベースと作業します。

- altova.mdbは次のデルクトリ内にあります: <マイドキュメント> Altova Map Force 2021 Map Force Examples ¥このデータベースはデフォルトの開発データベースの役割を果たします。
- altova.mdb は次のディレクトリ内にあります: <マイドキュメント
 >\Altova\MapForce2021\MapForceExamples\Tutorial\. このデータベースはデフォリトの生産データベースの役割を 果たします。

必要条件は、マメピングを編集することなく、ランタイムの前に、2つのデータベースのどちらかにマメピングを提供することです。この必要条件を 達成するコよ、2つの構成を持つデータベースグローィ シリノノース(データベースエイリアス)を作成します:

- 1. Default. この構成はデフォルトの開発データベースを指します。
- 2. Release. この構成はデフォルトのリースデータベースを指します。

ステップ1: グロー・ ジリノノース(データベースエイリアス)を作成する

- 1. 「ツール」メニューカら「グロー・バルノノース」をクリックします。(おさよ グロー・バルノソース 🛃 ツール いちなんをクリックします)。
- 2. 「追加 | データベース」をクリックします。
- 3. 「リソースエイリアス」テキストボックスに名前を入力します(例えば、SourceDatabase)。
- 4. 「データベースの選択」を選択し、Microsoft Access (ADO) を選択し、開発データベースを参照します(<マイドキュメント> ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥altova.mdb)。
- 5. 「ファイルの追加」 🏂 をクリックして、「ReleaseDatabase」とい名前を与えます。
- 6. 「データベースの選択」をクリンして、Microsoft Access (ADO) を選択し、生産データベースを参照します(<マイド キュメント> ¥Altova ¥Map Force2021 ¥Map ForceExamples ¥Tutorial **¥altova.mdb**)。

ステップ 2: マピング内のグロー・ ジレノソースを使用する

データベースエイリアスか再生されましたが、使用するためコよ、マンピングを変更する必要があります。

- 次のマピングを開きます: <マイドキュメント
 >\Altova\MapForce2021\MapForceExamples\Tutorial\PersonDB.mfd。
- 2. データベースコンポーネントを右クリックし、コンテキストメニューから「プロパティ」を選択します。
- 3. 「変更」をクルクして、作成されている「SourceDatabase」エイリアスを選択します。

🕑 データベースを選択	र 🗾
接続ウィザード	グローバル リソース グローバル リソース名
既存の接続	
mmo ADO接続	
DDBC 接続	
JDBC 接続	
COLUMN ROOINET 接続	
グローバル リソー ス	
PostgreSQL 接続	接続(C) 閉じる(L)

- 4. 「接続」をクルクします。
- 5. データベースオブジェクトを選択するようこプロレプトされると、デフォルトのセグションをそのままして、「OK」をクリックします。
- メモ データベース接続がウロー・ シレノンースとして設定されると、コンポーネント設定 ダイアログトの設定は、下で示されるように灰色で 表示されます。下で示される通りグロー・ シレノンース 型・ツール・・・ボタンをクリックして接続・・ラメーターを変更することができます。次も参照してくたさい、 グロー・ シレノンースとしてのデータベース。

コンポーネント設定	<u>-</u>	×
ーデータベースー		
データソース:	C:¥Altova¥Altova¥altova.mdb 変更	(C)
接続名:	SourceDatabase	
データベースの種類:	MS Access (ADO)	
接続文字列:	Data Source=C:¥Altova¥Altova¥altova.mdb; Provider=Microsoft.Jet.OLEDB.4.0	*
パラメーターはグローバ - ログイン設定	、ルリソース 'SourceDatabase' で定義されています	
ユ <i>ー</i> ザー <mark>(U):</mark>		
パスワ <i>ー</i> ド(P):		
JDBC 設定		
JDBCドライバー(J):	sun.jdbc.odbc.JdbcOdbcDriver	
データベースURL(L):	jdbc:odbc:;DRIVER=Microsoft Access Driver (*.mdb);	
ADO/OLEDB 設定		
データソース (D) :	C:¥Altova¥Altova¥altova.mdb	
カタログ(A):	altova	
プロバイダ − <mark>(∨)</mark> :	Microsoft.Jet.OLEDB.4.0	
追加オプション(0):	Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:¥Alt	ova¥
生成設定 トランザクションを テーブル名からス	· 使用(T) キーマ名を除外する(S)	
- ステートメント実行の)タイムアウト	
タイムアウト (M) :	60 秒 🗌 無限(I)	
	OK キャン	セル

ステップ 3: 希望する構成を使用してマッピングを実行する

マピングを実行する前に、希望するデータベースに簡単に切り替えることができます。

- 開発データベースからのデータを読み取るコは「ツール」メニューから「アクティブな構成 | デフォルト」をクリックします。
- 生産データベースからデータを読み取るコよ、メニューから「アクティブな構成 | ReleaseDatabase」をクルクします。

おけよ、必要とされる構成を「グロー・バリリソース」ドロップダウノリストから選択します。



構成を切り替える場合、ダイアログボックスは、ソースデータベースがまたなく再ロードされることをプロンプトします

構成の切替 - 再ロード	
次のリソースは変更されました:	
● グローバル リソース エイリアスからデータベース: SourceDatabase	A
	▼
ファイルを再ロードするか、変更を無視することができます。	
	再ロード(R) キャンセル

メモ このサンプルで使用される両方のデータベースでは、類似しているデータが含まれており、マンピングを実行した後に生成される出力 内には差異は存在しません。

9.9 例: アプリケーションワークフローを作成する

このサンプルは、グロー・シレリノースを使用した、Altova MapForce とAltova XMLSpy 間のシンプルなワークフローの作成方法について 説明しています。特に、MapForce マンピングの実行をXMLSpy から直接トリガーし、MapForce により生成されたマンピング出力を XMLSpy で開く方法について説明されています。これを可能にするコは、下て説明されるように、型「変換の結果」のブロー・シレノノースを作成します。

ステップ 1: グロー・ シレノソースを作成する

MapForce とXMLSpy の両方からこのステップを実行することかできます。

- 1. 「ツール」メニューカら「グロー・バルルノース」をクリックします。(おさよ グロー・バルノソース 🛃 ツール いちかんをクリックします)。
- 2. 「追加 | ファイル」をクルクします。
- 3. 「リソースエイリアス」テキストボックスに名前を入力します(このサンプルでは、「MappingResult」)。
- 4. オプション「MapForce 変換の結果」を選択します。
- 5. 「参照」をクトックして、マトング <マイドキュメント >\Altova\MapForce2021\MapForceExamples\Tutorial\Tut-ExpReport-multi.mfd を選択します。ダイアロ グボックス上で「入力」と「出力」セクションが作成されるまで待ってくたさい。下で示されるとおり、このマトピング には、1つの入力 とつの出力が存在します。

🕑 グローバル リソー	र हर
∈エイリアス名	
リソース エイリアス (A)・	MappingResult
77 ATT7749	
⊘構成	
+ to ×	─構成 "Default"の設定 ◎ ファイル(F)
Default	 MapForce 変換の結果(M) ◯ StyleVision 変換の結果(S)
	いたったいてんせばとれた山もたちにます。
	リソースは以下の主成された五刀を指しまり: C+¥Users》 (Documents¥Altova¥ManEorce20)
	人刀(I)
	西部 7人本 「 mf-ExpReport mf-ExpReport.xml 」
	-
	」 出力 (ラジオ ボタンにより選択) (0)
	名前 パス
	ExpReport-Target ExpReport-Target.xml
	C ExpReport-Target SecondXML.xml
	OK キャンセル

このサンプルのスコープのオーガス、フォルダー C:\temp | コンクのファイルをそれそれ生成し、デフォルトのファイル名を変更します。この目的を達成するコよ以下のとおり、希望される出力のオーガン構成を作成します:

1. 「出力」の下から、2番目の出力の横の「参照」をクルクして、C:\temp\file1.xml 目的のファイル名として入力します。これ は最初の出力ファイルがトリガーされると作成されるデフォルトの構成です。

	z 💽
エイリアス名 リソース エイリアス(A):	MappingResult
構成 ➡ <mark>也</mark>	構成 "Default" の設定 ○ファイル(F) ④ MapForce 変換の結果(M) ○ StyleVision 変換の結果(S) リソースは以下の生成された出力を指します: C:¥Users¥ i¥Documents¥Altova¥MapForce2017¥MapForceExamples¥Tutorial¥Tut-Exp 入力(f)
	名前 パス mf-ExpReport mf-ExpReport.xml 出力 (ラジオ ボタンにより選択) (0)
	ExpReport-Target C:\temp\file1.xml ExpReport-Target SecondXML.xml
	OK キャンセル

2. 「構成をコピーとして追加する」 や をクリックし、新しい構成のために名前を入力します(このサンプルでは、 "Output2")。
 3. 「出力」の下から、2番目の出力の横の「参照」をクリックして、C:\temp\file2.xml 目的のファイル名として入力します。これは、2番目の出力ファイルを生成する代替の構成です。

🕑 グローバル リソー	ス 💽
エイリアス名 リソース エイリアス(A):	MappingResult
構成 ➡	構成 "Output2"の設定 ○ファイル(F) ④ MapForce 変換の結果(M) ⑤ StyleVision 変換の結果(S) リソースは以下の生成された出力を指します: C:¥Users¥ ¥Documents¥Altova¥MapForce2017¥MapForceExamples¥Tutorial¥Tut-Exp を 入力(I) 2前 パス mf-ExpReport mf-ExpReport.xml
	出力 (ラジオ ボタンにより選択) (O) 名前 パス ExpReport-Target C\temp\file1.xml ExpReport-Target C\temp\file2.xml C M キャンセル

ステップ 2: ワークフローをトリガーする

前のステップで作成されたグローイ シリノースはXMLSpy で以下のよう 使用することができます。

- 1. XMLSpyを作動します。 2. 「ツール」メニューからグロー・バリノノースをクルクします。
- 3. 「ファイル」グループ内で、作成されている「MappingResult」グローイ ジルリノースを選択し、「表示」をクリックします。

🛃 グローバル	リソースの管理		
定義ファイル:	C:\Users\t	.∖Documents\Altov€	▶ 参照(B)
日 伯 ファイル の Mapping	Result		+ 追加(<u>A</u>)
	-/ベース URL -ス		
		-	※ 削除(<u>D</u>)
•		Þ	
		ОК	キャンセル

マンプを実行し、デフォルトのでの出力(file1.xml)を作成し、XMLSpy のメインペインコードします。ファイルは、以下の名前で保存されます: C:\temp\file1.xml.

マピングをトリガーするコよ、代替の構成を使用して以下のように実行します

1. 「ツール」メニューから「アクティブな構成 | Output2」をクトックします。

Configuration switch - Reload	
The following resources have been modified:	
Altova://file_resource/MappingResult	^
May any related the file place the designers the readification	~
r ou can reload the rile, close the document or ignore the modification	
<u>R</u> eload <u>C</u> lose Cance	

この結果、代替の出力ファイルはXMLSpy のメインペインコードされます。ファイルは以下の名前で保存されます: C: \temp\file2.xml。

10 MapForce のカスタマイズ

10.1 MapForce オプションの変更

MapForce 内の一般および他の優先順位を変更する方法は、以下の通りです:

• 「ツール」メニューカら「オプション」をクトックします。

使用することのできるオプションは、以下に表示されているとおりです。

全般

このページで使用することのできるページ内の設定は、以下のとおしです

ロゴの表示 起動時に表示	MapForce が起動されると、イメージ(スプラッシュスクリーン)を表示、おけは非表示します。
背景をグラデーションで表示	マメピングペイン内の背景をグラデーションで表示する機能を有効化、ませま無効化します。
表示する注釈をN 列に制限す る	このオプションは注釈をサポートするコンポーネントに対して適用されます(例えば、XML スキー マ、EDI)。注釈テキストに複数の行か管まれる場合、このオプションを有効化すると、コンポーネン ト上の最初のN 行のみが表示されます。N は指定することのできる値です。この設定は、コンポー ネント内で表示される SELECT ステートメントにも適用されます。
名前のエンコード	新規コンポーネントのナンリニデフォルトの文字の円コードを設定します。この設定は、各コンポーネントのナンリニ個別に変更することができます。次を参照してくナジュ、ユンポーネント設定の変更.
実行タイムアナの使用	「出力」ペイン内でマピングの結果をプレビューする際に実行のタイムアナを設定します。
ー時的なファイルの出力を生成 する	このオプションカ設定されていると、マノピングの結果をプレビューする際に生成される出力は、一時 的なファイルに書き込まれます(これはデフォルトのオプションです)。出力ファイル・マレニまだ存在し ないフォルダーか含まれる場合、MapForce がこれらのファイルを作成します。
	警告:実行のためにマンピングをサーバーにデプロイする場合、パス内のディレクトリはサ ーバートで存在する必要かあり、それ以外の場合、実行エラーか発生します。サーバー 実行のためにマンピングを準備するも参照してくたさい。
最後の出力ファイルに直接書き 込む	このオプションカ設定されていると、マノピングの結果をプレビューする際に生成される出力は、実際のファイルに書き込まれます出力ファイルレジロニまだ存在してないフォルダーカ含まれる場合、、マノピングエラーカ発生します。
	警告: このオプションは更なる確認をせず、既存の出力ファイルを上書きします。
N 文字のステップでテキストを表 示する	大きな XML とテキストファイルを生成するマピングをプレビューする際に、出力 ペイン内に表示されるテキストの最大のサイズを指定します。出力テキストがこの値を超えると、「更にロードする」 ポタンをクトックして次のチャンクをロードする必要があります。詳細に関しては、次を参照してくたさい、出力のプレビュー。

編集

このページで使用することのできる設定は、以下のとおりです

マウスのドラッグ時のユンポーネン	マケスでドラッグする際にエレポーネントおけば関数が他のエレポーネントと共に整列するかを指定します。
トの入れる	ニレポーネントの整列を参照。
スマートコンポーネントの削除	有効化されると、スマートコンポーネントの削除オプションは、削除された接続を "記憶" します。コンポーネトの削除後、接続を保持するを参照。

メッセージ

「このケッセージを表示しない」オプションにより以前に無効化されたケッセージの通知を再有効化します。

生成

プログラムのコード生成中に適用できるIDE とコンパラーの設定を定義することが有効化できます。コードジェネレーターのオプションおよび MapForce Server 実行ファイルニマンピングをコンパルするをそれそれ参照してくたさい。

データベース

DB Query ダブ内のデータベースをクエリするとき適用することのできる設定を定義することができます(次を参照してください、データベースの) 参照とクエリ)。

ネトワークプロキシ

<u>ネトワークプロキシの設定を参照してくたさい</u>

10.1.1 Java 設定

Java タブ上では、ファイルシステム上で Java VM (仮想マシン) への やを任意で入力することができます。カスタム Java VM / やの追加は常に必要なわけではありません。デフォルトで、MapForce は、Windows レジストリと JAVA_HOME 環境変数を(この順序で) 読み取ることにより、 Java VM / やを自動的に影響しようとします。このダイアログボックスに追加されるカスタム やは、自動的に検知される VM / やよ此見優先順位を与えられます。

カスタム Java VM / ひを追加する必要かある場合、例えば、インストーラを持たず、(例えば、Oracle のOpenJDK などの) レジストリ エトリを作成する必要のない Java 仮想マシンを使用している場合、カスタム Java VM / ひを追加する必要かある場合かあります。 MapForce により自動的に検知された Java VM / ひを何らかの理由でオー・・ーライドする場合、この ひを設定する必要があります。

Java	
Java VM ライブラリロケーション	
jvm.dll へのパス:	
E.g., C:¥Program Files (x86)¥Java¥jdk <some-version>¥bin¥server¥jvm.dll</some-version>	参照(B)
JVM の自動検知のためにフィールドを空のままにします。	
重要: JAVA ライブラリはアプリケーション (32-bit).と同じビットバージョンを必要とします。	
メモ: JVM が Altova アプリケーションの現在のインスタンス内で開始された場合、JVM ロケーションの変更は 再起動後に反映されます。	アプリケーションの

以下の点に注意してください

- Java VM パスはAltova デスケップと(サーバーではない)アプリケーションにお供有されています。この結果、アプリケーション内 で変更されると、他のAltova アプリケーションに自動的に適用されます。
- パロはJDK がインストールされているディレクトリー相対する \bin \server おけは \bin \client ディレクトリからの jvm.dll ファ イルを指す必要があります。
- MapForce プラオフォーム(32-ビナ、64-ビナ)がJDK と同じである必要があります。
- Java VM / CO変更後、新規の設定を反映するためこMapForce を再起動する必要があます。

Java VM ノマの変更は以下のエリア」影響を与えます:

- JDBC 接続性
- XSLT/XPath のためのJava 拡張関数

この設定はJava コード生成に影響を与えません。

10.1.2 ネットワークプロキシ設定

ネットワークプロキシ セクションでは、カスタムのプロキンの設定を構成することができます。(XML 検証の目的のナム)に)MapForce のインターネットへの接続方法に影響します。デフォルトでは、MapForce はシステムのプロキンの設定を使用します、ですから、多くの場合プロキンの設定を変更する必要におりません。必要な場合、代替のネットワークプロキシを下のオプションを使用して設定することができます。

メモ ネトワークプロキン設定は、Altova MissionKit アプケーション間で共有されています。結果、1つのアプケーション内で設定が 変更されると、自動的に他の全てのアプケーションに影響を与えます。

ネットワーク プロキシ
◎ システムのプロキシ設定を使用(U)
◎ 自動プロキシ構成(A)
☑ 自動検知の設定(D)
スクリプト URL(L)
◎ 手動のプロキシ構成(M)
HTTP プロキシ(H) ポート 0
このプロキシサーバーをすべてのプロトコールのために使用する(P)
SSL プロキシ(S) ポート 0
プロキシ無し(N)
プロキシのサーバーをローカルのアドレスのために使用しない(X)
現在のプロキシの設定(C)
URL のテスト(T)
IE 自動プロキシ構成が見つかりました。 メソッド WPAD (テスト URL http://www.example.com を使用) PAC に対して解決された DIRECT (NO PROXY). プロキシを使用していません

システムプロキシの設定の使用

システムプロキン設定を介して構成可能なインターネットエクスプローラー(IE)設定を使用します。netsh.exe winhttpを介して構成される設定が必要とされます。

自動プロキシの構成

以下のオプランを使用することができます

- 自動検知の設定: DHCP おはDNS を使用して W PAD スクリプト (http://wpad.LOCALDOMAIN/wpad.dat) を検索し、プロキシセトアップのためにのスクリプトを使用します。
- スクリプト URL: プロキンセオアップのために使用されるプロキン自動構成 (.pac) スクリプトに対する HTTP URL を指定します。
- *再ロード:* 現在の自動プロキン構成をパセナして再ロードします。このアウションゴはWindows 8 おけお以降が必要とされ、30 秒程の時間が必要です。

手動のプロキシの構成

ホスト名とポートを対応する製品のプロキンのために手動で指定します。サポートされるスキームレオスト名に含まれている場合があります(例 : http://hostname)。プロキンがスキームをサポートする場合、対応するプロトコールと同じである必要におりません。

以下のオプランを使用することができます

- このプロキシサーバーをすべてのプロトコールのために使用する:全てのプロトコルのためにHTTP プロキシの木スト名とポート使用します。
- プロキシ無し: セミコロン(;)によ)区別されているプロキンを使用しないホスト名、ドメイン名、おけよ、ホストのためのIP アドレスのJスト。IP アドレスは別時でられず、IPv6 アドレスは角かこで囲まれる必要があります(例) [2606:2800:220:1:248:1893:25c8:1946])。ドメイン名は、ドットと共に開始される必要があります(例:
 .example.com).

- プロキシのサーバーをローカルのアドレスのために使用しない:チェックされている場合、プロキシ無しリストのために
 を追加します。このオプションが選択されている場合、次の場合、プロキジメ使用されません(i) 127.0.0.1、(ii) [::1]、(iii)
 (.)ドット文字を含んでしないサスト名すべて。
- メモ プロキシサーバーが設定されており、マンピングをAltova FlowForce Server にデプロイする場合、ローカルアドレスのために プロキシサーバーを使用しないを選択する必要があります。

現在のプロキシの設定

プロキシの検知の詳細なログを提供します。URL のテスト フィールドの右の「更新」ボタンを使用して更新することができます(例、URL のテスト を変更する場合、おけよ、プロキシの設定か変更された場合)。

• URL のテスト: A test URL のテストを使用して、どのプロキンカ特定のURL ために使用されているかを確認することができます。 URL を使用して、I/O はされません。 プロキのシ自動構成が選択されている場合、このフィールドは空にしておどの要かあります(システムプロキシの設定の使用、おけよ、自動プロキシの構成を使用して)。

10.2 キーボードのショートカット

デフォルでMapForce は以下のキーボードのショートカナを提供してます:

F1	ヘルプメニュー
F2	次のブックマークへ(出力ウィドウニて)
F3	次を検索
F10	メニュー・・・をアクティベート
Num +	現在のアイテムノードを展開
Num -	アイテムノードを縮退
Num *	現在のアイテムノード以下を全て展開
CTRL + TAB	開かれているマッピング間の列い替え
CTRL + F6	開かれているウイドウをサイクル
CTRL + F4	アクティブぶマッピングドキュメントを閉じる
Alt + F4	MapForceを閉じる
Alt + F, F, 1	最後に閉じられナファイルを開く
Alt + F, T, 1	最後に閉じられたプロジェクトを開く
CTRL + N	新規ファイル
CTRL + O	ファイルを開く
CTRL + S	ファイルの保存
CTRL + P	ファイルの印刷
CTRL + A	すべて選択
CTRL + X	切り取り
CTRL + C	コピー
CTRL + V	貼り付け
CTRL + Z	元に戻す
CTRL + Y	や値し
Del	コンポーネントを削除(プロンプト付き)
Shift + Del	コンポーネントを削除(プロンプトなし)
CTRL + F	検索
F3	次を検索
Shift + F3	前を検索
矢印キー (上/下) Esc Return	コンポーネント内にあるアイテムを選択 編集の破棄 / ダイアログボックスを閉じる 選択の確定
出カウィンドウホットキー CTRL + F2 F2 SHIFT + F2 CTRL + SHIFT + F2	ブッケマークの挿入/削除 次のブッケマークへ 前のブッケマークへ 全てのブッケマークを削除
ズームホットキーhotkeys CTRL + マウスホイール前	ズームイン

CTRL + マカホイール後 ズームアナ CTRL + 0 (ゼロ) ズームのルナ

10.2.1 ショートカットのカスタム化

MapForce 内のキーボードのショナカナを以下のように定義おける変更することができます。

- 1. 「ツール」メニューから「カスタマイズ」をクトックします。
- 2. 「キーボード」タブをクリックします。

コマンドに新規ショートカットを割り当てる

- 1. 「ツール」カスタマイ」コマボを選択して、キーボードタブをクリックします。
- 2. 「カテゴリー」コンボボックスをクリックして、メニュー名を選択します。
- 3. コマイリストボックス内で新しいショートカットを割り当てるコマンドを選択します。
- 4. 「新規ショートカットキー」をクリックします:テキストボックスから、コマンドを有効化するショートカットキーを押します。

<u>カスタマイズ</u>	\times
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): 設定するアクセラレータ(F): ファイル(F) デフォルト コマンド(O): デフォルト マンド(O): 現在のキー(U): トロット(C): 割り当て(A) ド規ショートカットキー(N): 削除(R) 説明: C# のコード生成	
閉じる	

ショートカトはデキストボックス内にすぐに表示されます。ショートカトが以前に割り当てられている場合、その関数がデキストボックスの下に表示されます。

5. 「割り当」てポシンをクルクして、ショートカナを割り当てます。 現在のキーリストボックス ミュートカナカ 抜示されます。 (新規ショートカナトキーテキストボックス内のエントリをクリアする ゴよ、エトロールキーを押します CTRL、ALT まけよ SHIFT)。

ショートカットを元に戻す、または削除する

- 1. 現在のキーリストボックス内が削除するショートカットをクリックします。
- 2. 「削除」ボシをクリックします。
- 3. 「閉じる」ボタンクトックして、確認します。
- メモ 設定するアクセラレーターイコは現在関数が存在しません

10.3 メニューをカスタム化する方法

標準 MapForce メニューおよび例えば、コマイを追加、変更、および削除するオークリコンテキストメニューをカスタム化することができます。メニューのカスタム化をデフォルトの状態にした水することができます。

メニューをカスタム化する方法

• 「ツール」メニューから、「オプション」をクトックして、「メニュー」をクトックします。

Customize	×
Commands Toolbars Keyboard Menu	Options
Application Frame Menus:	Context Menus:
<u>S</u> how Menus for: Default Menu ~	Select <u>c</u> ontext menu:
<u>R</u> eset	Reset
Default application menu. Appears when no documents are open.	Hint: select the context menu, change the page to 'Commands' and drag the toolbar buttons into the menu window.
☑ Menu s <u>h</u> adows	
	Close

メニューをカスタム化する方法

[デフォルトメニュー] バーはメインウィボウ内でドキュメトが開かれていない場合に表示されます。[MapForce デザイン] メニュー・・・・ つまけは複数のマメピングが開かれていると表示されるメニュー・・ーです。各メニュー・・ーを個別にカスタム化することができ、カスタム化の変更 はお互いて影響を与えません。

メニュー・・をカスタム化するコは「メニューの表示」ドロップダウノノストから選択します。[コマンド] タブをクリックして[コマンド] リストからコマンドをメニュー・・まオコはメニューイボラッグします。

メニューからコマンドを削除しりセナする方法

メニュー内のメニューおけまコマイを全て削除するけまれ下を行います

- 1. メニューの表示ドロップダウノストから以下のしつを選択します:
 - デフォルトのメニュー(これ)おキュメントか開かれていない場合に使用可能なメニューを表示します)
 - MapForce デザイン(こればキュメノトか開かれていない場合に使用可能なメニューを表示します)

- 2. カスタム化ダイアログか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I つから削除するコマンドを選択します。
- 3. (i) メニューカらメニュー デオオオメニュコマンドカらドラッグし(ii) メニューオオオオメニューコマンドを右クトックし [削除]を選択しま す。

[メニューの表示]ドロップダウノリストから選択し[リセット]ボタンをクリックしてメニューを元の状態にした水することができます。

アプリケーションのコンテキストメニューをカスタマイズする方法

コンテキストメニューーはアプリケーションのインターフェイス内の特定のオブジェクトを右クリックすると表示されるメニューです。これらのコンテキストメニューのそれそれは以下を行ってカスタム化することができます

- 1. [コンテキストメニューの選択] ドロップダウンリストからコンテキストメニューを選択します。コンテキストメニューがポップアップされます。
- 2. [コマンド] ダブをクリックします。
- 3. [コマンド]リストボックスからコマンドをコンテキスメニューインドラッグします。
- 4. コンテキストメニューカらコマイを削除するけなに、コンテキストメニュー内でそのコマイを右クトックして「削除」を選択します。代わりに、コマイをコンテキストメニューカらドラッグします。

[コンテキストメニューの選択]ドロップダウノノスト内で選択し[リセット] ポタンをクトックしてコンテキストメニューを初期のインストールの状態にします。

メニューとメニューの影 全てのメニューに影を与えるために〔メニューの影〕 チェックボックスを選択します。

10.4 カタログファイル

MapForce はOASIS XML カタログメカニズムのサブセナをサポートします (https://www.oasis-

open.org/committees/entity/spec-2001-08-06.html)。カタログメカニズムこよりMapForce が共通に使用されるDTD と XML スキーマ(およびスタイルシートと他のファイル)をパブリックURI から解決する代わりにコーカルフォルダからを抽出することができるように なります。これは全体的な処理スピードを増やしオフライン(すなわちや・トワークに接続されていない状態)での作業を可能にし、ドキュメント のポータビリティを改善します。

カタログのしくみ

カタログは一般的にンジトクDTD おけよ通常は、ローカルファイルレジローカルURI ヘのスキーマレファレンスをリダイレクトするけっかに使用 されます。これを達成するけっかに、XML フォーマトのカタログファイルはンジトクスキーマURI とローカルURI の間のマメピングを定義しま す。MapForce がXMLドキュメトを解析する都度、カタログファイル内部のスキーマURI (おけよ、適用可能の場合、DTD の、ジトッ クまけよンステム識別子) か最初に検索されます。マメピングがサタログファイル内で見つかけた場合、そのレファレンスが使用されスキーマ1カー カルファイルから読み取られます。カタログファイル内でマンピングが見つからない場合 XMLドキュメントのURI はそのまま解決されます。

例えば、以下のXML ファイルはMapForce により処理される必要があると仮定します。

```
<?xml version="1.0" encoding="UTF-8"?>
<Articles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Articles.xsd">
<Article>
<Number>1</Number>
<Name>T-Shirt</Name>
<SinglePrice>25</SinglePrice>
</Article>
</Article>
```

(MapForce か既知の通り) catalog.xml ファイルがローカルディレクトリニ存在し、以下のラインか含まれると想定します。この点について は以下で詳しく説明されています:



XML ファイルを解析する場合、MapForce はカタログファイル内でスキーマレファレンス

http://www.w3.org/2001/XMLSchema-instance.xsd のよめの一致を検知します。この結果スキーマイよカタログファイルに 対して相対的なローカル ② files/XMLSchema-instance.xsd からロードされますカタログファイルファイル内でマメピングが見つか られ 場合、スキーマイよhttp://www.w3.org/2001/XMLSchema-instance からロードされます。

ルートカタログ

MapForce が開始されると、RootCatalog.xml と呼ばれるファイルを Program Files ディレクトル シロードします。 RootCatalog.xml はそれぞれが nextCatalog 要素内にあるカタログファイルのリストを含んでいます。これらのカタログファイルおレック アップされ内部にある URI は指定されているマノピングに従い MapForce によ 解決されます。

RootCatalog.xml

上記のノスティングで、以下のカタログがリックアップのためにノストされていることに注意してくたさい

- CustomCatalog.xml は自身のマピングを作成することができるファイルです。このファイルは以下のディレクトリ内にあります: C:\users\<name>\Documents\Altova\MapForce2021。Altova 構成済みのカタログファイルで既にカスタムスキーマのために対処されていたは、場合(次のポイントを参照してくたさい) CustomCatalog.xml にマピングを追加することができます。
- %AltovaCommonFolder%/Schemas ディレケーからの複数のcatalog.xml ファイル。各 catalog.xml ファイルは (SVG、DITA、DocBook、W SDL などの)特定のスキーマのディレケーリの内部にあり、それぞれは対応するスキーマのローカ ルで保存されたエピーを指す URI への グリックおよび、おけまたステム識別子をマップします。
- CoreCatalog.xml ははスキーマをロケートするけっかのAltova 固有のマノピングが含まれています。このファイルはMapForce [Program Files] 内にあります。

以下の点に注意してくたさい

- CustomCatalog.xmlを変更する場合 <u>サポートされる要素</u>を使用してください。エラーを引き起こす可能性があるため既存のマンピングを複製したいてください。
- %AltovaCommonFolder%\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 へのレフォレンスを含んできます。XML スキーマドキュメントをこれらのスキーマイズはしてはむりません。古 し WML スキーマ仕様はこのようなドキュメントと作業するための対応するスキーマ URI を効果的に解決する機能をMapForce に提供するためこのみ含まれていました。

上記の一部のデルノオリーマは環境関数の助けを得て表示されています。レファレンステーブルに関しては環境関数を参照してくたさい。

サポートされる要素

CustomCatalog.xml内でエトリを作成するコよ以下にリストされる要素のみを使用して下さ、OASIS XMLカタログ社様の他の要素はサポートされていません。

要素	属性	例
public	 systemId はハースの ヴトグ識別子 を指定します。 uri URI レファレンズを指定します(例え ばローカルディレケトリニズする相対的な パズ) 	<pre><public publicid="-//W3C//DTD XMLSCHEMA 200102//EN" uri="files/XMLSchema.dtd"></public></pre>
system	 systemId はハースのシステム識別子を 	<system< th=""></system<>

要素	属性	例
	指定します。 ・ uri URI レファレンスを指定します(例え ば ローカルディレクトリニシオする相対的な パズ)	<pre>systemId="http://www.w3.org/2009 /XMLSchema/datatypes.dtd" uri="files/datatypes.dtd"/></pre>
uri	 name はURI レフルシスを指定します Uri は(例えば ローカルレアイルに対する 相対的な) ななど)代替 URI レフルシス を指定します 	<uri name="http://www.w3.org/2009/XML Schema/XMLSchema.xsd" uri="files/XMLSchema.xsd"/></uri
rewriteURI	 uriStartStringは再書き込みする URIの開始部分を指定します。 Uri置換文字列を指定します(例えば、 ローカルディンケトリンマする相対的ない文) 	<pre><rewriteuri rewriteprefix="files/" uristartstring="http://www.altov a.com/schemas/svg/"></rewriteuri></pre>
rewriteSystem	 systemIdStartString は再書き込 みするシステムの識別子の開始部分を指 定します。 rewritePrefix 置換文字列を指定 します(例えば、ローカルディレクトリーン対す る相対的ない シ) 	<pre><rewritesystem rewriteprefix="files/" systemidstartstring="http://www. altova.com/schemas/svg/"></rewritesystem></pre>

public、system、およびuri 要素は相対的な URI か解決されるベース URI を指定する xml:base 属性を取ることができます。詳細 に関しては XML カタログ仕様 (<u>http://www.oasis-open.org/committees/entity/spec-2001-08-06.html</u>) を参照してくたさい。

環境変数

下のテーブルはWindows 上での多種のシステムロケーションへの なを指定する nextCatalog 要素内でサポートされるすべての環境 変数をリストしています。

%AltovaCommonFolder%	すべてのAltova プログラムで共通のファイルを保管するためご使用されているディレクトリーのフ ル ぷ。オペレーティングシステムとMapForce (32 ビット おけよ64 ビット) のプラル フォームこ より、パスはC:\Program Files\Altova\Common2021 おけはC:\Program Files (x86)\Altova\Common2021 しなります。
%DesktopFolder%	デスケップ上にファイルオブシェケを保管するかとので見合いるディンケリークのフルマ。典型的ないなよこ:\Users\Username\Desktopです。
%ProgramMenuFolder%	ユ ーザーのプログラムグループを含む自身がファイルシステムのディレオトリであるディレオトリへのフル ノな。典型的なノダはこ: \Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs です。
%StartMenuFolder%	ユーザーの起動メニューアイテムを含むディントトリへのフルトス。典型的ないなはと: \Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu です。
%StartUpFolder%	ユーザーの起動プログラムグループ 文内応するディンケトリーのフル な。典型的な なは _C : \Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup です。

%TemplateFolder%	ドキュメトテンプレートのための共通レポネリとしての役割を果たすディノオリンのフルウ。典型的ないなはc: \Users\username\AppData\Roaming\Microsoft\Windows\Templatesです。
%AdminToolsFolder%	現在のユーザーのため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ークのフル な。典型的 な ぷはこ: \Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools です。
%AppDataFolder%	アプリケーション固有のデータのナダの共通レポネト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。典型的ないなよこ:\Users\username\AppData\Roamingです。
%FavoritesFolder%	現在のユーザーのお気に入りディンオーリトフル み。典型的ない なは _C : \Users\Username\Favorites です。
%PersonalFolder%	現在のユーザーの個人ディントリーのフル な。典型的な みはC: \Users\Username\Documents です。
%SendToFolder%	宛先のメニューアイテムを含むディントリーのフル な。典型的ないなは C: \Users\username\AppData\Roaming\Microsoft\Windows\SendTo です。
%FontsFolder%	システムフォントディレクトリークのフルマ。典型的ないなはC:\Windows\Fontsです。
%ProgramFilesFolder%	プログラムファイルディノオ・リークのフル な。典型的ない なはC:\Program Files とC: \Program Files (x86) です。
%CommonFilesFolder%	共通ファイルディノオリークのフル な。典型的な なはC:\Users\Public\Public Documents です。
%WindowsFolder%	(%WINDIR% 環境変数と同様)ウイボウディレクトリ ヘクフフル な。典型的な なはC: \Windows です。
%SystemFolder%	システムフォルダーへのフル な。典型的な なは & WINDIR & \system 32 です。
%CommonAppDataFolder%	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レス。典型的ないなはC: \ProgramDataです。
%LocalAppDataFolder%	ローカル(非ローミング) アプリケーションのナ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ークのフル な。典型的な) なは c: \Users\username\AppData\Local です。
%MyPicturesFolder%	現在のユーザーのビクチャオディンオーリーのフル マ。典型的な) マはこ: \Users\Username\Pictures です。

11 Visual Studio のための MapForce プラグイン

MapForce 2021 をMicrosoft Visual Studio バージョン2010/2012/2013/2015/2017/2019 に統合することができます。これ はMapForce のマビング機能とVisual Studio の開発環境とつの世界をつなぎます。MapForce プラグインが有効化されると、マビ ングプロジェケトとファイルを直接 Visual Studio から作成することができます。MapForce のスタイドアロンの、ージョンと同様に、メニューと ツール、一を含む、MapForce オプションをカスタム化することができます。

刘 MapForcePrj1 - Microsoft Visual Studio 🗸 🖌 Quid	:k Launch (Ctrl+Q)
File Edit View Project Build Debug Insert Component Connection Function	Output MapForce Sign in 💽
Team Tools Test Analyze Window Help	
	🥌 🗏 851T 851T, 80 TABLE C++ C++ BULLT 🕺 🚜 "
- ※ \$P\$ ● 1 ● 2 社 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Altova_Hierarchical_DB.mfd 😐 🗙	▼ Solution Explorer ▼ ₽ ×
	^
톏 Altova_Hierarchical 📋 AltovaTarget	Search Solution Explorer (Ctrl++)
🔺 🖓 🗍 File: Altova_Hierarchicl	
C C Altova	MapForcePrj I' (1 project)
> ─── () PrimaryKey > > □\ PrimaryKey >	
	Altova Hierarchical DB mfd
> () ForeignKey >	
Desc Desc Desc Desc Desc	
🔁 👘 🚺 EMail 📭 🛶 👘 💷 🖬 EMail 🔯	
🖒 — () Established 🕨 — 👘 🛄 Established 🗘	
>	Overview
Name	
Phone Phone Phone P	
I> I ForeignKey I> III ForeignKey = Offil>	↓ Libraries
<u>∢</u>	🔁 core 🔺
Mapping DB Query Output	aggregate functions
Messages 🗸 🗸	× avg result
	max result
Altova Hierarchical DB.mfd: Mapping validation successful - 0 error(s), 0 warning(s)	max-string result
Altova Hierarchical DB.mfd: Execution successful - 0 error(s), 0 warning(s)	Search for function
	Add /Pemovo Librarian
· · · · · · · · · · · · · · · · · · ·	Aud/ hellove ubrailes
C Ready	🛧 Add to Source Control 🔺 🔡

MapForce Enterprise Edition プラグイン(Visual Studio 2017)

11.1 プラグインの有効化

必要条件:

- Microsoft Visual Studio 2010/2012/2013/2015/2017/2019
- MapForce (Enterprise おけよProfessional Edition)
- MapForce Integration Package を以下でダウレロードすることができます https://www.altova.com/mapforce/download。
- メモ MapForce をVisual Studio プラグインとて使用するコよ、32-ビットのMapForce とMapForce Integration Package をインストールする必要があます。Visual Studio 内での64-ビットプラグインはサポートされません。

Visual Studio のためのMapForce プラグインを有効化するコよ(MapForce と同じバージョンの) MapForce Integration Package をダンロードして、実行し、スクリーンのインストールの方法に従ってくたさい。

インストール中、「Microsoft Visual Studio をインストール」プラグインオプションが選択されていることを確認してけさい

🚼 Altova MapForce 2019 Integration Package - セットアップ	\times
Altova MapForce 2019 Integration Package オブション インストールするコンボーネントを選択してください	
このパッケージには製品の機能を他のアプリケーションへ統合するためのコンポーネント が含まれています.	
 ✓ ActiveX コントロールをインストール(必須) ✓ ActiveX コントロールの .NET アセンブリ ラッパーをインストール 	
次の開発プラットフォームに統合をインストールする:	
☑ Microsoft Visual Studio プラグインをインストール	
☑ Eclipse プラグインをインストール	
< 戻る(B) 次へ(N) > キャンセノ	V

プロンプトされると、プラグインが有効化されるべき Visual Studio バージョンを選択します。例

🛃 Altova M	lapForce 2020 Integration Package - セットアップ	\times						
	Altova MapForce 2020 Integration Package オブション インストールするコンボーネントを選択してください							
以下から使用する Visual Studio のバージョンを選択してください Altova MapForce 2020 Integration Package								
	Visual Studio 2010							
	Visual Studio 2012							
	Visual Studio 2013							
	Visual Studio 2015							
🗹 Visual Studio 2017 の全てのインスタンス								
	☑ Visual Studio 2019 の全てのインスタンス							
	< 戻る(B) 次へ(N) > キャンセル							

メモ 使用中のオペレーティングシステムにインストールされている Visual Studio の、ージョンのみか選択のために表示されます。

Integration Package がインストールされると、MapForce 機能がVisual Studio 環境内で使用することができるようプンリます。

11.2 マッピングとプロジェクトと作業

Visual Studio のためのMapForce プラグインが有効化されている場合、スタイアロン MapForce グラスイカルユーザーインターフェイス とは異なり、Visual Studio 環境に適用できる方法でマナピングとマナビングプロジェクトを作成、おけは開くことができます。例(「ファイル| 新規作成」メニューコマイを使用して) Visual Studio 内に新規のファイルを作成すると、おけよ(「ファイル|新規プロジェクト」メ ニューコマイトを使用して)新規アイテムをプロジェクトに追加すると、MapForce ファイル型として選択することができます。

project	mapforce		× -		<u>C</u> lear all
	All <u>l</u> anguages	 All platforms 	-	All project <u>t</u> ypes	
Recent project templates					
A list of your recently accessed templates will be displayed here.	Simple Ma Creates a si	<mark>pForce</mark> Project imple, empty <mark>MapForce</mark> Project that	may contain	multiple mappings	
	Web Servic Creates a N	e <mark>MapForce</mark> Project <mark>1apForce</mark> WebService Project from a	WSDL file		
		Not finding what you're l	ooking for?		
		Install more tools and	features		

新規ファイルダイアログボックス(MapForce Enterprise エディションプラグインを使用した Visual Studio 2019)

既存のマピングファイル おさまプロジェクトを開く場合、適用することのできる Visual Studio メニューを使用します (例「ファイル | 開く | ファイル」 おさは「ファイル | 開く | プロジェクト / ノリューション」)を使用して、 MapForce - ご 関連した ファイルの型を検索します。

11.3 共通メニューと関数へのアクセス

Visual Studio のためのMapForce プラグインが有効化されている場合、下で表示されるように共通メニューと関数にアクセスすることができます。これはデフォルトのセナアップですが、Visual Studio の「ツール | カスタマイズ」メニューからメニューの場所を変更することができます。

グロー・ジレノソース

MapForce <u>グローイ シリンース</u>は Visual Studio 内の「MapForce | グローイシリノースの管理」メニューで使用できます。

メモ Visual Studio 2019 内では対応するメニューは「拡張子 | Map Force | グロー・ ジレノソースの管理」です。

MapForce オプション

MapForce オプションは Visual Studio 内の「ツール | MapForce オプション」メニューで使用できます。

マピングペインのカスタム化

Visual StudioのメインペインでMapForceマナングが開かれると、「表示 | MapForce」メニューを使用することができます。 MapForceのスタイドアロンバージョンと同じオプロンカ含まれます。

ライブラリウィンドウ

MapForce ライブラリウイドウは、プラグインがインストールされた後、デフォルトではVisual Studio 内で有効化されていません。このウイン ドウで頻繁に作業する場合、「表示 | MapForce | ライブラリウィンドウ」メニューから有効化することができます(このメニューは、メイン ウイドウ内でマンピングファイルは開かれると、Visual Studio 内で使用することができます)。ライブラリウイドウが有効化されると、インター フェイスの特定の箇所に、他のVisual Studio のエンポーネトと同様ドッキングすることができます。。

Libraries	- □ ×					
🔁 core	▲					
aggregate fur	nctions					
avg	resu 🛀 vg(va					
count	result = count(
max	resu 🛄 ax(va					
max-string	resut ax(st					
min						
min-string						
string-join	resulting-					
sum	resu 💴 um(va					
conversion functions						
Search for function 🗸 🗸						
Add/Remove Libraries						

ライブラリウィンドウ(MapForce Enterprise エディション プラグインとVisual Studio 2017)

ツールドーとコマンドのカスタム化

MapForce メニューとソールレーをVisual Studio の「ツール カスタマイズ」メニューからカスタム化することができます。

ヘルプとサポート

Visual Studio の「ヘルプ | Map Force ヘルプ」メニュー で Map Force ヘルプ、サポトセター、最新情報のチェック、および Map Force についてのメニューを使用することができます。
12 Eclipse のための MapForce プラグイン

Eclipse は、プラグインのフォーム内で届けられる異なる種類のアプリケーションを統合するオープンノースのフレームワークです。MapForce Enterprise とProfessional Edition をEclipse バージョン 2020-12, 2020-09, 2020-06, 2020-03 に統合し、Eclipse から 直接 MapForce 機能にアクセスすることができます。



MapForce Enterprise Edition プラグイン(Eclipse 4.4)

以下のトピックははのインストールおよび使用にていて説明しています。

- Eclipse のためのMapForce プラグインのインストール
- MapForce パースペクティブ
- 共通メニューと関数へのアクセス
- マピングビプロジェクトとの作業
- Eclipse のためのMapForce プラグインの有効化

12.1 Eclipse のための MapForce プラグインのインストール

必要条件:

- Eclipse 2020-12, 2020-09, 2020-06, 2020-03 (<u>http://www.eclipse.org</u>)。
- the 64ゼオプラオフォームのオオの Java Runtime Environment (JRE) おは Java Development Kit (JDK)
- MapForce Enterprise おはProfessional Edition 64ビ水
- MapForce Integration Package 64-ビナを以下でダウンロードすることができます
 <u>https://www.altova.com/ja/components/download</u>。下記の通りEclipse 統合をインストールの一部とて行うことができます。

上記の必要条件は64-ビオプラオフォームを必要としています。古いEclipse 32-ビオプラオフォームとの統合はサポトされてしません。

Eclipse のための MapForce プラグインをインストールする方法

Eclipse のためのMapForce プラグインの統合を以下の方法の一つで行うことができます。

- 1. 自動的にMapForce Integration Package のインストール中に統合することができます。これは奨励されるオプションです)。
- 2. MapForce Integration Package を最初にインストール、Eclipse からプラグインを手動で統合する。

Eclipse のための MapForce プラグインを統合する方法:

1. インストールウィザードを開始するためこMapForce Integration Package を実行します。

MapForce Integration Package をインストールはオゴアンインストールする場合 Eclipse を閉じる必要がおります。

2. プロンプトされると Eclipse プラグインのインストール オブションを選択し [次へ].をクリックします。

🖟 Altova N	🖟 Altova MapForce 2021 (x64) Integration Package - Setup			×	
	Altova MapForce 2021 (x64) Please configure the compo	Integration Pation Patients to be i	ackage options nstalled		
This pack functional	age contains components tha ity into other applications.	at allow integr	ation of the produ	cťs	
🗹 In	stall the ActiveX Control (requi	red)			
	Install .NET assembly wrapp	ers for the Ac	tiveX Control		
Install integrations into the following development platforms:					
M	Microsoft Visual Studio does not support 64-bit plug-ins.				
🗹 In	☑ Install the Eclipse plug-in				
		< <u>B</u> ack	<u>N</u> ext >	Cancel	

- 3. プロンプトされると MapForce プラグインの Eclipse への統合方法を選択します。以下の1つを行ってけさい
 - プラグインインストールを自動的に完了する場合 このウィザードに Altova MapForce プラグインを Eclipse に統合する を選択し、Eclipse 実行可能ファイル(eclipse.exe) がロードされている場所を参照します。
 - Eclipse 内で後にプラブインのインストールを完了する場合、clear the このウィザード... チェックボックスをクリアします。

🖟 Altova MapForce 2021 (x64) Integration Package - Setup			×	
	Eclipse plug-in Integrate the plug-in into	Eclipse		
This wizard installed following Eclipse update sites on your system: C:\Program Files\Altova\Common2021\eclipse\UpdateSite\ C:\Program Files\Altova\MapForce2021\eclipse\UpdateSite\				
To integrate Altova MapForce start Eclipse and select menu Help Install New Software, add above folders as a new local "Sites" and install the contained plug-ins. Or,				
 ✓ Let this wizard integrate Altova MapForce plug-in into Eclipse. Please, enter installation location of your Eclipse (x64): C:\Users\altova\eclipse\java-2020-06\eclipse\ 				
		< <u>B</u> ack	<u>N</u> ext >	Cancel

4. [次] をクリックすることでインストールを完了します。自動統合を選択する場合 MapForce パースペクティブとメニューが E clipse を次回起動する際に E clipse 内で使用可能にないます。

Eclipse のための MapForce プラグインを手動で統合する方法

- 1. Eclipse にてメニューオブションからへルプ 新規ソフト ウェアのインストール を選択します。
- 2. [インストール]ダイアログボックス内で[追加]をクトックします。

🖨 Install —	
Available Software Select a site or enter the location of a site.	
Work with: type or select a site <a> Add	<u>M</u> anage
type filter text	<u>S</u> elect All
Name	<u>D</u> eselect All

3. [レポネーリの追加] ダイアログボックス内で[ローカル]をクリックします。フォルダー C:\Program Files\Altova\Common2021\eclipse\UpdateSite を参照して選択します。("Altova" などの) サイトの名前を与えま す。

🖨 Add R	lepository	—		×
<u>N</u> ame:	Altova		L <u>o</u> cal	
Location:	file:/C:/Program Files/Altova/Common2021/eclipse/	Upda	<u>A</u> rchiv	e
ОК				
?	A <u>d</u> d		Cance	ł

- 4. ステップ2から3を繰り返します、今度は、フォルダーC:\Program Files\Altova\MapForce2021\eclipse\UpdateSiteを選択し、MapForceなどの名前を与えます。
- 5. インストールダイアログボックスで -- Only Local Sites -- を選択します。次に [Altova category] フォルダーを選択し [次へ] をクリックします。

🖨 Install	_	
Available So Check the ite	ftware ms that you wish to install.	
Work with: -	-Only Local Sites ~ <u>A</u> dd	<u>M</u> anage
type filter text	:	<u>S</u> elect All
Name V III Altova category Altova Eclipse Plug-in Library MapForce Plugin		<u>D</u> eselect All

- 6. インストールされるアイテムをレビューし[次へ]をクリックして継続します。
- 7. 使用許諾契約書に合意するには対応するチェックボックスを選択します。
- 8. [完了]をクリックすることでインストールを完了します。
- メモ (不足するアイコンなど)プラグインの使用に関する問題が発生した場合 -clean プラグを使用してコマンドラインから E clipse を 起動します。

12.2 MapForce パースペクティブ

Eclipse のためのMapForce プラブインをインストールすると、Eclipse 内で新しレットースペクティブ("MapForce")を使用することがで きるよう ゴンはます。この トースペクティブのレイアナトは、MapForceのスタイドアロンエディションのインターフェイス ご類似しています。 MapForce パースペクティブご切り替える コよ 「ウィンドウーノパースペクティブを開く」その他」をクリックし、MapForce をリストから選択します。



Eclipse 内でMapForce パースペクティブを選択する

MapForce パースペクティブは Eclipse の他の ペースペクティブと同じで、必要に応じて Eclipse 内(「ウィンドウ | ナビゲーション | 次へ パースペクティブ」)で切り替えます。含まれるビューをカスタム化することもでき、 Eclipse 設定から他の複数のオブションをカスタム化します。 (MapForce パースペクティブをカスタム化するゴは、 MapForce パースペクティブ 【Window | パースペクティブを開く | その他]から 切り替え、メニューゴマンド [Window | パースペクティブのカスタム化]を選択します)。 Eclipse パースペクティブを開く | その他]から Eclipse ドキュメントを参照してください。以下のスクノーンショント MapForce パースペクティブのスイッチがオンされている Eclipse 環境を 示しています。



MapForce パースペクティブ(Eclipse のためのMapForce Enterprise Edition プラグイン)

デフォルトでは、Eclipse内でのMapForceパースペクティブは以下のよう 整理されています:

- Eclipse エディターでマンピングデザイン ウィドウを使用することができます。MapForcein のスタイドアロンエディション内と同じタ ブと機能が搭載されています。
- Eclipse ビューとして、メインマッピングエディター「こ左のライブラリウイド」 「友使用することができます。このビューか表示されていない 場合、MapForce パースペクティブに切り替えます。メニューコマナド「ウインドウ」ビューの表示 | ライブラリ」を選択します。ラ イブラビューにより、定義済みませばよカスタム定義された男数と関数ライブラレ作業できます。
- Eclipse ビューとして、メインマッピングエディター下のかッセージペインを使用することができます。メッセージビューが表示されていない 場合、MapForce パースペクティブに切り替え、メニューコマンド「ウィンドウ|ビューの表示 | メッセージ」を選択します。メッセージは、検証メッセージ、エラー、と警告を表示します。
- Eclipse ビューとして、概要ペインを使用することができます。アナラインビューが表示されていない場合は、MapForce パースペクティブに切り替え メニューコマボ「ウィンドウトビューの表示 トアナ・ライン」を選択します。このビューを使用することにより、 大きなマンピングのマンピングデザインの特定の箇所に素早く移動することができます。

MapForce マビングを開くと、Eclipse がMapForce パースペクティブに自動的に切り替えるように構成することもできます。これを行う には、メニューコマンド「ウィンドウ|設定」を開き、MapForce を選択して、「ファイルが開かれると自動的に MapForce パースペ クティブに切り替える」チェックボックスを選択します。

Preferences		
 Preferences type filter text General Ant Code Recommenders Help Install/Update Java MapForce Maven Mylyn Run/Debug Team 	MapForce MapForce Perspective Automatically switch to MapForce perspective at Preferences shared with MapForce installation Open MapForce Options Dialog	t file open
 ▷ Team Validation ▷ WindowBuilder ▷ XML 	Restore De	faults <u>A</u> pply
?	ОК	Cancel

設定 ダイアログボックス

12.3 共通メニューと関数へのアクセス

Eclipse 内では、下にリストされるEclipse 固有のパレーションを除き、スタイドアロンバージョン内と同様にMapForce 機能の同じメ ニューの多数にアクセスすることができます、これはデフォリトのセナアップですが、 Eclipse からのインターフェイスの基本設定を更にカスタマ イズする場合は、 (<u>MapForce パースペクティブ</u>を参照してください)。

メモ コンテキストが関連しない場合、Eclipse内では、MapForceメニューグループおけまコマンドの一部が無効化(おけま使用できない)場合があります。例: Eclipse内でマンピングデザインファイル(.mfd)がアクティブは場合のみ、「挿入」メニューを使用することができます。

MapForce 標準メニューイニ関する詳細は、メニューレファレンスを参照してくたさい。

一般的なMapForce コマンド

MapForcein のスタイアロンエディション内で、マメピングデザインファイルに適用することのできる「検証」、「FlowForce Server にデ プロイ」、「コードの生成」などの)コマイドは、「ファイル」メニュー内で使用することができます。Eclipse 内でこれらのコマイドは MapForce メニュー内おけよMapForce ツール・一で使用することができます。(MapForce プロジェクトファイルを含む) ファイルを開い たい閉じたりするコマイドは、Eclipse の「ファイル」メニューで使用することができることに注意してください。

```
8 🔁 😼 🔹
```

Eclipse 内のMapForce ツールドー

🕙 ツールレーポタンは MapForce ヘルプファイルを開きます。

▶ アール・・・ボタンは、 MapForce ファイル固有のコマンドを表示します。このボタンを展開すると、現在 Eclipse エディター上でア クティブなファイルにより決定される使用可能なコマンドが表示されます。例:マンピングデザイン(.mfd) ファイル固有のコマンドは、このような ファイルがでアクティブな場合、Eclipse エディター内で使用することができます。

グロー・ジレノソース

グロー・ジリノースへアクセスまたはグロー・ジリノノースの管理は、以下を行います

- MapForce ジール、オタンをクリックして展開し、グローバリノノースをクリックします。
- MapForce メニューカらグローノ シレノソースをクリックします。

MapForce プロジェクト

MapForce のスタイドアロンのエディションでは、プロジェクト メニューは、マンピング プロジェクト (.mfp) ファイルに適用することのできる複数のコマイドを含んで、ます。Eclipse 内では、これらのコマイドは以下のように存在します。

- プロジェクトを開くませるコマンドは、Eclipse「ファイル」メニューから使用することができます。
- 他のプロジェクト専用のコマイは、Eclipse内でコンテキストコマイとして使用できます。コンテキストコマイを表示、 MapForceプロジェクト(.mfp)ファイルを作成おける課人場合はプロジェクトを右クトックしてください。



標準 MapForce プロジェクト (.mfp) に加え、Eclipse 内では、「MapForce/Eclipse」の型のプロジェクトも作成することができます。 このようよプロジェクトは、2面の特質のため、MapForceコードの自動的なビルドと生成のために構成することができます。マノピングおよび プロジェクトと作業を参照してくたさい

MapForce オプション

MapForce オプションは、「ウインドウ | 優先」メニューから使用することができます。優先ダイアログボックスで、MapForce を選択し、 「MapForce オプションダイアログを開く」をクリックします。

Preferences		
Preferences type filter text General Ant Code Recommenders Help Install/Update Java MapForce Mylyn Run/Debug Team Validation WindowBuilder XML	MapForce Perspective Automatically switch to MapForce perspective at file open Preferences shared with MapForce installation Open MapForce Options Dialog Restore Defaults	□ ×
?	OK	ncel

設定ダイアログボックス

ライブラリウィンドウ

Eclipse 内で、MapForce ライブラリウイボウはビューとして使用することができます。このビューはデフォルトでは、メインエディターウイボ ウク左に配置されています。(MapForce に関連したすべてのビューは、MapForce パースペクティブ かけして切り替えられると、 Eclipse インターフェイス で表示されます <u>MapForce パースペクティブ</u>も参照)。

MapForce プラグインバージョン

Eclipse のよめのMapForce プラグインの現在インストールされているレージョンを確認するコよ、Eclipse メニューオプション「ヘルプ | Eclipse プラットフォームについて」を選択して、そして、MapForce アイコンを選択します。

へいプとサポート

MapForce ヘルプ、サポトセター、最新情報 メニューは、Eclipse の「ヘルプ | MapForce ヘルプ」メニューで使用することができます。

12.4 マッピングおよびプロジェクトと作業

When Eclipse のようのMapForce プラグインがインストールされている場合、Eclipse のEclipse プロジェクトからMapForce の スタイドアロンエディション内と同じマピング とマピングプロジェクト型 を作成することができます。マピングをデザイン、コル イル、デプロール、 マピングコードを生成するコス、新規のEclipse プロジェクトを作成するか、おけよ既存のEclipse プロジェクトを使用することができます (例 MapForce マピングを追加する Java プロジェクト)。

更に、MapForce/Eclipse プロジェクト内の MapForce プラグインをインストール後にEclipse 内で使用することのできる、特別な プロジェクト型内で全てのマメピングと作業することができます。カスタマイズを選択しない限り、MapForce/Eclipse プロジェクトはデフォル で Java ビルダーとMapForce コード生成ビルダーか割り当てられています。更に、2つのEclipse 性質が存在します: MapForce 性質 と JDT (Java 開発ツール) 性質。この結果、MapForce/Eclipse プロジェクトは、(マノピングデザインファイルなどの)リノースを保存ま 行は変更する場合、以下に記載されるように振る舞います:

- プロジェクト > 「自動的にビルド」メニューオプションが有効化されていると、マピングコードは自動的に生成されます。1つまた は複数のMapForce プロジェクトファイルMapForce/Eclipse プロジェクト内に存在する場合、コード生成言語と出力ター ゲトフォルダーは、各プロジェクトファイル内の設定により決定されます。それ以外の場合、Eclipse は、場所を選択するようこプ ロンプトします。
- エラーと出力メッセージは、メッセージと問題ビューに表示されます。

このセクションは、以下のトピックを含みます

- <u>MapForce/Eclipse プロジェクトの作成</u>
- 新しいマピングの作成
- Eclipse プロジェクト (既存のマピングをへポートする)
- 自動ビルドの構成とMapForce コードの生成

12.4.1 MapForce/Eclipse プロジェクトの作成

MapForce/Eclipse プロジェクトの作成方法:

- 1. 「ファイル」メニューから「新規作成 | その他」をクルクします。
- 2. 「MapForce/Eclipse プロジェクト」カテゴリーを選択します。

New	
Select a wizard	
<u>W</u> izards:	
type filter text	
 ▷ ➢ General ▷ ➢ CVS ▷ ➢ Git ▷ ➢ Java ▷ ➢ MapForce Files ▲ ➢ MapForce/Eclipse Project ◎ MapForce/Eclipse Project ▷ ➢ Maven ▷ ➢ Tasks ▷ ➢ WindowBuilder ▷ ➢ XML ▷ ➢ Examples 	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

3. 「次へ」をクルクします。

New MapForce Project	- • •
Specify Name and Code Language of Project	
Create new MapForce/Eclipse Project	
Project name: MapForceEclipseProject1	
✓ Use <u>d</u> efault location	
Location: C:\eclipse44\workspace\MapForceEclipseProject1	Browse
Builder for MapForce mappings and projects Image: style="text-align: center;">Image: style="text-align: center;"/>Image: s	
Additional Builders for Code Generated by MapForce © use no additional builders © use IDT builder	
 use following nature: 	
org.eclipse.jdt.core.javanature	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

- 4. プロジェクト名を入力して、プロジェクトの保存場所を選択します。「Map Force ビルダーをプロジェクトへ追加」そのままして、「JDT ビルダーを使用」オプロンをそのままします。
- 5. 「完了」をクルクします。

12.4.2 新しいマッピングの作成

以下のMapForce ファイル型をEclipse プロジェクト内で作成することができます:

- MapForce マピング
- MapForce プロジェクトファイル
- MapForce Web サービスプロジェクト (MapForce Enterprise Edition で使用することができます)

Eclipse プロジェクト内でこれらのファイル方を作成する

- 1. 新規のEclipse プロジェクトを作成、おけて既存のプロジェクトを開きます。
- 2. 「ファイル」メニューから新規作成」をクリックし、「その他」をクリックします。

e New	- • •
Select a wizard	
<u>W</u> izards:	
type filter text	
 General CVS Git Java MapForce Files MapForce Mapping MapForce Project File MapForce Web Service Project MapForce/Eclipse Project Maven Tasks WindowBuilder 	E
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

3. ウィザードダイアログボックスから必要なファイル方を選択し、「次へ」をクリックします。

MapForceMessages.NewProjectFilePageText	- • ×			
Specify Location and Name of the New MapForce Project File				
Create a New MapForce Project File				
Inter or select the parent folder:				
↓ 🔁 JavaProject1				
File name: MappingProject1.mfp				
Advanced >>				
< <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel			

4. 既存のプロジェクト内の親フォルダーを選択して、「完了」をクリックします。

12.4.3 Eclipse プロジェクトに既存のマッピングをインポートする

MapForce マッピングと依存ファイルを既存のEclipse プロジェクトにインポートする

- 1. ファイルをインポートするプロジェクトを開きます。
- 2. 「ファイル」メニューカジィンポート」をクリックします。

⊜インポート	- • ×
選択 ローカル・ファイル・システムから既存のプロジェクトへリソースをインポートします。	Ľ
インポート・ソースの選択(S):	
 → 一般 ♪ アーカイブ・ファイル ○、ファイル・システム ※ 既存プロジェクトをワークスペースへ □ 設定 > ● インストール > ● チーム > ● 実行/デパッグ 	
(P) 次へ(N) > 完了(F) #	マンセル

3. 「ファイルシステム」を選択して、「次へ」をクリックします。

⊜ Import	- • •
File system Please specify folder	
From directory: C:\Users\User\Documents\Altova\MapForce2015\MapForceExamples -	B <u>r</u> owse
▶ ▼ MapForceExamples ▼ ■ 837-Q2.x12 ▼ ▲ 837-Q2.xsd ▼ ▲ 837-Q2.xsd ▼ ▲ Accounts.mdb ▼ ▲ address.xsd ▼ ■ ADT_A28.1.hl7 ▼ ● Altova_Hierarchical_DB.mfd ▼ ● Altova_Hierarchical_Excel.mfd ▼ ● Altova_Hierarchical_FLF.mfd	d
Filter Types Select All Into folder:	Bro <u>w</u> se
Options Overwrite existing resources without warning Create top-level folder Advanced >>	
(?) < <u>Back</u> <u>N</u> ext > <u>Finish</u>	Cancel

- 4. 「次のディレクトリから」の横で、ヘルポートするファイルの場所を参照し、必要なファイルを選択します。
- 5. 「宛先フォルダー」の横の、「参照」をクトックして、ファイルを追加するプロジェクトを選択します(このサンプルでは、 *MapForceEclipseプロジェクト1*)。

Import into Folder	
Select a folder to import into.	
III ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	
bin 🔁	
src 🦳	
(?) OK	Cancel
	Current

6. 「OK」をクリックして、「完了」をクリックします。

12.4.4 自動ビルドの構成と MapForce コードの生成

自動的な MapForce コードのビルドと生成は、MapForce/Eclipse プロジェクト内でデフォルトとて有効化されて、ます <u>MapForce/Eclipse プロジェクトの作成</u>を参照してください。MapForce/Eclipse 型ではよい既存のプロジェクト内のMapForce コードの自動ビルドとコードの作成を行うには、MapForce コード生成ビルダーとMapForce ネイチャーを手動で追加して行います。

MapForce コード生成ビルダーをプロジェクトに追加する方法:

• Eclipse .project ファイル 以下で イライトされてい るラインを追加します:

```
<buildSpec>
<buildCommand>
<name>org.eclipse.jdt.core.javabuilder</name>
<arguments>
```

```
</arguments>
</buildCommand>
</buildCommand>
</buildCommand>
</name>com.altova.mapforceeclipseplugin.MapForceBuilder</name>
</arguments>
<//arguments>
<//buildCommand>
</buildCommand>
</buildSpec>
```

MapForce のネイチャーをプロジェクトに追加する:

• Eclipse .project ファイル 以下で イライトされてい ろラインを追加します:

```
<natures>
    <nature>org.eclipse.jdt.core.javanature</nature>
        <nature>com.altova.mapforceeclipseplugin.MapForceNature</nature>
        </natures>
```

ヒント: **.project** ファイルを素早く E clipse のナビケータービューから開くことができます(このビューを有効化する)コよメニューコマナド 「ウィンドウ|ビューの表示 | ナビゲーター」を選択します)。

自動的な Map Force コード 生成のオン/オフの切り替え:

• プロジェクト メニューから自動的にビルドをクリックします。

MapForce コード 生成ビルダーの無効化:

- 1. プロジェクト メニューから プロパティ」をクリックします。
- 2. 「ビルダー」をクリックします。

Properties for JavaProject1		
type filter text	Builders	↓ ↓ ↓ ↓
 Resource Builders Java Build Path Java Code Style Java Compiler Java Editor Javadoc Location Project References Refactoring History Run/Debug Settings Task Repository Task Tags Validation WikiText 	Configure the builders for the project: Image: Description of the project of the	New Import Edit Remove Up Down
?	ОК	Cancel

3. 「MapForce コード生成」チェックボックスのチェックを解除するためこクトックします。

12.5 Eclipse のための MapForce プラグインの有効化

Eclipse のようのMapForce プラグインは、ID を持つEclipse 拡張点 com.altova.mapforceeclipseplugin を提供します。 この拡張点を使用して、MapForce プラグインの機能性を適用ませは拡張します。拡張点は、<u>MapForce エトロールとMapForce</u> <u>API</u> のこのMーインターフェイス へのアクセンを提供します。

MapForce Eclipse インストール シケージは、プラグインの簡単なサンプルを含んでいます。新規 MapForce マンピングのファイルを開く イベントをチェックし、マンピングビューのズームのレベルを70% に設定します。

拡張ポイントのJavaDoc ドキュメントは、MapForce プラグイン インストールディレクトリ(C:\Program Files\Altova\MapForce2021\eclipse\docs\) で検索することができます。

サノプルMapForce プラブインをインストールして実行する前に、以下の必要条件が満たされていることを確認してくたさい

- 64ビオ Java を使用している場合は、64ビオ Eclipse、64ビオ MapForce おび64ビオ MapForce Integration Package が必要です。
- JDT (Java Development Tools) プラグインのインストール。
- Eclipse PDE (プラグイン開発環境) のインストール。

サンプル Map Force プラグイン プロジェクトをワーケプレースにインポート する

- 1. Eclipse を開始します。
- 2. 「ファイル」メニューから「インポート」をクリックします。
- 3. 「一般 | 既存のプロジェクトをワークプレースへ」を選択して、「次へ」をクリックします。
- 4. 「ルートディレクトリの選択」フィーリドの横の「参照」ボタンをクトックして、サンプルプロジェクトディレクトリを選択します。例 C: \Program Files\Altova\MapForce2021\eclipse\workspace\MapForceExtension.
- 5. 「プロジェクトをワークプレースへコピー」オブションを選択して、「完了」をクリックします。 "MapForceExtension" とう名前 の新規プロジェクトがワークプレイス内に作成されます。

サンプル拡張 プラグインの実行:

- 1. Java パースペクティブご切り替えます。
- 2. 「実行」メニューから構成を実行」をクリックします。
- 3. 「Eclipse アプリケーション」を右クトック、「新規作成」を選択します。(リスト内に "Eclipse アプリケーション" が存在しな し場合、Eclipse プラグイ、開発ツールがEclipse 環境にインストールされていない ことを意味します。Eclipse プラグインプラ グイ、開発ツールをインストールする」は、「ヘルプ」メニュー内の「新しいツフトウェアをインストールする」をクトックして、 "Eclipse プラグイ、開発ツール"を "Eclipse プロジェクト更新" ダウムードサイトからインストールます。
- 4. 名前と新しい構成を入力し、(このサンプルでは、SampleMapForcePlugin)「適用」をクリックします。
- 5. MapForceClient ワーケプレイスプラグインが、プラグインタブ内で選択されていることを確認してくたさい。
- 6. 「実行」をクルクして、新しいEclipse ワークベンチが開かれます。
- 7. MapForce マビングを新しいワークベンチで開きます。マビングは、70%のズームレベルで開かれます。

13 メニューレファレンス

以下のセクションでは MapForce に用意されているメニューやメニューオプションのリストと それらの簡単な説明を記します。

13.1 ファイル

新規作成

新規マピングドキュメトおけはマピングプロジェクト(*mfp)を開きます。

開く

既に保存されたマッピング(*.mfd)ませまマッピングプロジェクト(*.mfp)ファイルを開きます。使用中のMapForce エディションで使用できない機能が含まれているファイルは、開くことができません

上書き保存

現在アクティブなファイル名で、現在アクティブ」なっているマッピングを保存します。

名前をつけて保存

現在アクティブになっているマッピングを、別の名前で保存します。およ保存されていないワァイルの場合、ファイルに新たな名前をつけることができます。

全て保存

現在開かれている全てのマパングファイルを保存します。

再ロード

現在アクティブコンコンし、るマッピングを再ロードします。それまでご作成された変更点を破棄してもよりの尋ねられます。

閉じる

現在アクティブコンコンし、るマノピングを閉じます。ファイルを閉じる前に保存するか尋ねられます。

全て閉じる

現在開かれている全てのマピングファイルを閉じます。保存されていないファイルを保存するか尋ねられます。

印刷

印刷ダイアログボックスが開かれ、マンピングを印刷することができます。



印刷ダイアログボックス

「現在の倍率」を選択すると、現在マピングに対して指定されているズーム倍率が使用されます。「最適化された倍率」を選択すると、マピングがページの大きさコ収められるよう、印刷が最適化されます。コンポーネントのスクロール・イエ印刷されません。1 つのコンポーネント か複数のページュ切り分けて印刷してもよいが指定することもできます。

印刷プレビュー

上に記述された機能を持った印刷ダイアログボックスが表示されます。

印刷設定 印刷設定ダイアログボックスが表示され、使用するプレターや用紙の設定などを指定することができます。

マンピングの検証 マンピングを検証により、全てのマンピング(接続)か妥当であるかと、エラーや警告か表示されます。 詳細についてはマッピングの検証を参照くたさい。

マピングの設定

マピングの設定ダイアログボックスではギキュメトごとの設定を定義することができます。マピングの設定の変更を参照してくたさい。

選択された言語でコードを生成する

現在選択されているマピングの言語により一ドが生成されます。現在選択されている言語は、ツールレーで、イライトされているプログランク言語のアイエン(XSLT、XSLT2、XSLT3、XQuery、Java、C#、おゴはC++で確認することができます。

コード生成 | XSLT (XSLT2、XSLT3)

このコマンドにより、ソースファイルからの変換に必要な XSLT ファイルか生成されます。このオプションを選択すると、フォルダーの参照ダイアロ グボックスか表示され、生成された XSLT ファイルを配置する場所を指定することができます。マレング設定の変更を参照してくたさい。

コード生成 | XQuery

このコマイドにより、ソースファイルからの変換に必要なXQuery ファイルが生成されます。このオプションを選択すると、フォルダーの参照ダイア ログボックスが表示され、生成されたXQuery ファイルを配置する場所を指定することができます。マメニング設定の変更を参照してくたさい。

コード生成 | Java | C# | C++

これらのコマンドにより、ソースファイルから変換を行うのこ必要なアプリケーションプログラムのコードが生成されます。このコマンドを選択するとフォルダーの参照ダイアログボックスが表示され、生成されたファイルが配置される場所を指定することができます。生成されるアプリケーションファイル(ならび)にプロジェクトファイル・* csproj C# プロジェクトファイル * sln ソリューションファイル * vcproj Visual C++ プロジェクトファイル)の名前は、マンピング設定ダイアログボックスのアプリケーション名フィールドにより定義されます。メニューオアションの「ファイル | マンピングの設定」を選択することで、このダイアログへアクセスすることができます。マンピング設定の変更を参照してくたさい。

ターゲナがXML/スキーマドキュメントの場合、コードの実行により作成されるファイル名は、ユンポーネント設定ダイアログボックスの出力 XML ファイルフィーリドにて与えられている名前となります。

MapForce Server 実行ファイルのコンパイル

マルングの変換を実行するためにMapForce Server で実行されるファイルを生成します。MapForce マルングのエンパルを参照してくたさい。

Flow Force Server ヘデプロイ 現在アクティブなマピングをFlow Force Server ヘデプロ化ます。MapForce マピングのデプロイを参照してくたさい。

ドキュメントの生成

様々な出力フォーマトにより、マピングプロジェクトの詳細なドキュメンテーションを生成します(マソピングドキュメントの生成とカスタム化を参照してください)。

最近のプロジェクト

最近開かれたプロジェクトのノストを表示します。

終了

アプリケーションを終了します。保存されていないドキュメノトかある場合、それらを保存するか尋ねられます。

13.2 編集

このメニュー以下にあるコマドの殆どは、出力タブにてマピングの結果を参照している時、おけはXSLTタブにてXSLTコードのプレビューを行なっている際に利用することができます。

元に戻す

MapForce では使用回数に制限の無い「元に戻す」操作を行うことができ、マメピングのデザインステップを順を追って確認することができます。



や値しコマイにより、元に戻すコマイによし戻された動作を進めることができます。これらのコマイを使用することで、操作の履歴を確認することができます。

検索 🌺

XSLT、XSLT2、XSLT3、XQuery おけ出力タブにて、指定されたテキストを検索することができます。

次を検索 F3 📌

同じ検索文字列が次に出現する箇所を検索することができます。

前を検索 Shift F3 🌤

同じ検索文字列が前に出現する箇所を検索することができます。

切り取り/コピー/貼り付け/削除

標準的なWindowsの編集コマイです。マピングウィイウ内に表示されているコンポーネントや関数を切り取ったシレピーすることができます。

すべて選択

マッピングタブニある全てのエレポーネートを選択する、まけはXSLT、XSLT2、XSLT3、XQuery まけは出力タブニ表示されているテキスト/コード全体を選択することができます。

13.3 挿入

XML スキーマ/ファイル

XML スキーマおけはインスタンスをマンピングに追加します。スキーマを参照する XML ファイルを選択すると、追加情報をマンピングに追加す る必要はありません。スキーマ参照のない XML ファイルを選択すると、自動的に一致する XML スキーマを生成するかを問われます。(XML スキーマの生成 を参照してくたさい)。 XML スキーマファイルを選択すると、プレビューのためのデータを提供する XML インスタンスを任意で 追加するかを問われます。

データベース 📴

データノースませまターゲットコンポーネントとしてデータベースコンポーネントを挿入することができます。データベースによりスキーマ構造が与えられ、ツノー形式により表示されます。

テキストファイル

フラオなテキストファイルドキュメント(例: CSV)、おさば固定長のテキストファイルを挿入することができます。両方の種類とも、ソースならび

ロターゲトコンポーネトとして使用することができます。入力の挿入 マピングウイドウがマピングを表示すると このコマイドにより入力コンポーネントをマピングに追加することができます。(マピングリコ ラメ <u>ーターを与える</u>を参照してください)。マピングウイドウショーザー定義関数を表示すると、このコマイドはユーザー定義関数の入力コンポー ネノトを追加します。(ユーザー定義関数内の、ラメーターを参照してください)。

出力の挿入

マピングウイドウがマピングを表示すると、このコマドにお出力コンポーネントをマピングに追加することができます(マソピングから文字列 の値を返すを参照してくたさい)。マピングウイドウがユーザー定義関数を表示すると、このコマドはユーザー定義関数の出力コンポーネ ントを追加します。ユーザー定義関数内の、ラメーターを参照してくたさい。

定数 🖭

<u>コネクタ</u>に固定のデーダを提供する定数を挿入します。コンポーネントを作成する際にデータはダイアログボックスに入力されます。以下のデー タ型から選択することができます:文字列、数値、その他から選択することができます。



標準的な(非インライン型の)ユーザー定義関数と等価な中間変数を挿入します。変数はインスタンスファイルを持たない構造コンポーネントで、マッピング処理を単純にするために使用されます。詳細については、<u>中間変数</u>を参照くたとい。

並べ替え: node/row

ノードの並べ替えを許可するコンポーネントを挿入します。詳細に関しては、ノードノ行の並べ替えを参照してくたさい。



node/rowとbool、そしてon-trueとon-falseとら2つの入力と出力パーターを備えたエンポーネントを挿入します。 Boolean がtrue どこつている場合、node/row パラメーターの値がon-true パラメーターから渡され、Boolean がfalse どこつている場合、node/row パラメーターの値がon-false パラメーターへ渡されます。フィルターの使用方法についてはフィルターと条件を参照してくたさい。

SQL-WHERE/ORDER

データベースのデータに対して使用することができるフィルターコンポーネントを挿入することができます。MapForce により生成されたクロルに対して、任意のSQL W HERE 句を追加することができます。詳細については <u>SQL W HERE / ORDER コンポーネント</u>を参照くたさい。



ルックアップテーブルを使って入力された値を特定の値へ変換するエンポーネントを挿入します。(例えば、月の数値を月の名前にマップするなど)他の値のセットに値のセットをマップする場合に沿に立ちます。詳細については<u>Value-Mapの使用</u>を参照くたさい。

IF-Else 条件

"If-Else 条件"のかちのエレポーネトを挿入します。詳細に関しては、フィルターと条件を参照してくたさい。



例外コンポーネトによりマンプの処理を、特定の条件が満たされた場合に中断することができます。例外の追加を参照してくたさい。

13.4 **プロジェクト**

MapForce では、マルチドキュメントインターフェースがサポートされており、複数のマンピングを1つのマンピングプロジェクトへまとめることができます。詳細に関しては、マンピングプロジェクトとの作業を参照してくたさい。

プロジェクトの再ロード

現在アクティブなプロジェクトを再ロードして、プロジェクトタブを表示します。

プロジェクトを閉じる

現在アクティブコなっているプロジェクトを閉じます。

プロジェクトの保存

現在アクティブゴンっているプロジェクトを保存します。

プロジェクトへファイルを追加する

ファイルを開くダイアログボックスにより、現在開かれているプロジェクトにマッピングを追加することができます。

アクティブなファイルをプロジェクトへ追加する

現在アクティブなファイルを、現在開かれているプロジェクトに追加することができます。

フォルダーの作成

このオプションにより、現在開かれているプロジェクト構造へ、新たなフォルダーが追加されます。このメニューオプションは、使用できるときだけ選択可能になります。詳細に関しては、<u>プロジェクトフォルダーの管理</u>を参照してください。

プロジェクト全体のロードの生成

プロジェクトウイドウン現在表示されているプロジェクト全体に対してプロジェクトコードを生成します。現在選択されているデフォルトの言語で、各フォルダーニある*mfdマピングファイルのコードが生成されます。

コードの生成...

サブメニューから選択した言語にてプロジェクトコードを生成します。

プロ・ディ

プロジェクト全体の設定を定義することのできるダイアログボックスを開きます。詳細に関しては、コード生成設定の設定を参照してくたさい。

最近のプロジェクト - 1.2. etc. 最近に開かれプロジェクトをしみします。

13.5 コンポーネント

ルト要素の変更

XML インスタンスドキュメトのルト要素を変更することができます。

XMLSpy でスキーマ定義を編集

XML スキーマエンポーネントが選択された状態でこのオプションを選択すると、XMLSpy のスキーマビューで XML スキーマファイルが開かれ、 編集を行うことができます

データベースオブジェクトの追加/削除/編集

データベースコンポーネント内のデータベースオブジェクトを追加、削除ませる変更することができます。

更新

現在アクティブコンコンし、るデータベースコンポーネントの構造を、データベースから再ロードします。

前に入力を複製して追加

選択されたアイテムのコピークローンを、現在選択されているアイテムの前に挿入します。コピーされたアイテムコは出力アイコンが含まれておらず、データ構造として使用することはできません。この機能を使用した例については複数のノースから1つのターゲットにマップ セウションを参照 くたさい、複製されたアイテムを右クトックすると、コンテキストメニューから複製アイテムを移動することができます。必要につ応じて複製されたアイ テムを上/下へ移動させてくたさい。

後に入力を複製して追加

選択されたアイテムのコピークローンを、現在選択されているアイテムの後に挿入します。コピーされたアイテムコは出力アイテムか含まれておらず、データ構造として使用することはできません。この機能を使用した例については<u>複数のノースから1つのターゲルトにマップ</u>セクションを参照 くたさい、複製されたアイテムを右クリックすると、コンテキストメニューから複製アイテムを移動することができます。必要に「応じて複製されたアイ テムを上/下へ移動させてくたさい。

複製を削除

既に作成された複製アイテムを削除します。この機能を使用した例については複数のノースから1つのターゲットにマップセクションを参照ください。

データベーステーブルのアケション

選択されているターゲットデータベーステーブルへ、データがマッピングされた際に取られるアケションを定義することができます。詳細についてはデ <u>ータベーステーブルアケションの設定</u>のセケションを参照ください。

データベースのクエリ

データベースコンポーネント内で選択されているテーブル/フィールドをベースに、SELECTステートメントを作成します。テーブル/フィールドが選択されている状態でこのコマンドが使用可能になり、SELECTステートメントがSELECTウインドリニ自動的に配置されます。

ットを左揃えてする

全てのアイテムをコンポーネトの左端へ揃えます。

ットを右揃えこする

全てのアイテムをコンポーネントの右端へ揃えます。この表示方法はカーゲットスキーマへのマンピングを行う際に利用することができます。

プロ/ ^አጉィ

現在選択されているエレポーネトの設定を表示するダイアログボックスを表示します。エレポーネト設定を変更するを参照してくたさい。

13.6 接続

子要素の自動接続

子要素の自動接続機能を有効化ませば無効化します。ツールドーこあるアイエンからも選択を行うことができます。

子要素の接続設定

ー致する子要素の接続設定ダイアログボックスか表示され、接続設定を行うことができます。(一致する子要素の接続を参照)。

一致する子要素の接続

このコマンドにより、ソースとターゲットスキーマ間で同じ名前を持すを複数のアイテムに対して接続を作成することができます。このダイアログボッ

クスで定義した設定はそのまま保持され、ツール・ーニある子要素の自動接続アイコン 接続した際に適用されます。このアイコンをクトックすることで、アイコンの有効/無効状態を切り替えることができます。詳細については一致す る子要素の接続のセクションを参照くたさい。

標準マピング(ターゲト優先マピング)

接続の種類を標準的なマピングへ変更します。詳細については標準マピング(ターゲト優先マピング)を参照してください。

全てコピー(子要素)

親ユネクタのサブソリーとして表示されている子アイテムのユネクタが存在する状況で、マッチする全ての子要素に対してユネクタを作成します。 詳細については<u>すべてユピー接続</u>を参照くたさい。

ソース優先マピング(混合コンテンソ)

接続をノース優先 / 混合コンテンン型のコネクター変更して、新たご要素を選択してマンピングすることができるようになります。詳細に関しては、 ソース優先マンピング(混合コンテンツ)を参照してくたさい。

プロッティ

現在のコネクターズもして固有(混合コンテンツの設定を定義するオメクのダイアログボックスか表示されます。利用することができなしオプションは グレーアサンされます。これらの設定はテキストノードを持たされ、複合型のアイテムコズルても適用されます。詳細は<u>考読設定</u>を参照してください。 さい。

13.7 関数

ユーザー定義関数の作成

新たなユーザー定義関数を作成します。(詳細にてんてはユーザー定義関数を参照くたえい。

選択からユーザー定義関数作成

マピングウイドウェて現在選択されている要素をベースに、新たなユーザー定義関数を作成します。

関数設定

現在アクティブなユーザー定義関数の設定ダイアログボックスを開き、現在の設定を変更することができます。

関数削除

既存のユーザー定義関数を開き、その関数の名前がダブニ表示されている状態で、現在アクティブはユーザー定義関数を削除します。

入力の挿入

マピングウイドウがマピングを表示すると、このコマドは、マピングに入力コンポーネトを追加します(詳細については<u>単純型入力</u>を 参照ください)。マピングウイドウトローザー定義関数を表示すると、このコマドは、マピングに入力コンポーネントをユーザー定義関数に 追加します(詳細については<u>ユーザー定義関数内の、ラメーター</u>を参照ください)。

出力の挿入

マピングウイボウがマピングを表示すると、このコマボは、マピングに出力コンポーネトを追加します(詳細については<u>単純型出力</u>を 参照ください)。マピングウイボウカローザー定義関数を表示すると、このコマボは、マピングに出力コンポーネントをユーザー定義関数に 追加します(詳細については<u>ユーザー定義関数内の、ラメーター</u>を参照くたさい)。

13.8 出力

XSLT 1.0、XSLT 2.0、XSLT 3.0、XQuery、Java、C#、C++、Built-in 実行エンジン マピングが実行される変換言語を設定します。変換言語の選択を参照してくたさい。

出力ファイルの検証

参照されているスキーマイン対して出力 XML ファイルを検証します。

生成された出力の保存 出力タブン現在表示されているデータを保存します。

生成されたすべての出力を保存

動的マピングにより生成された出力をファイルに保存します。詳細については複数の入力ませる出力ファイルを動的に処理を参照くたさい。

出力を再生成 現在のマピングを出力ウイドウから再生成します。

SQL スクリプトを実行 SQL スクリプトが現在出カウイドウで表示されている場合、テーブルアクションの定義に従ってかたちで、ターゲトデータベースに対してマピングを実行します。

ブックマークの挿入/削除 出カウィンドウのカーノル位置にブックマークを挿入します

次のブックマーク 出カウィンドウターニある次のブックマークへ移動します。

前のブックマーク 出カウィンドウターンのある前のブックマークへ移動します。

全てのブックマークを削除 出カウィドウェて現在定義されている全てのブックマークを削除します。

XML テキストの整形

出カペインて表示されているXMLドキュメントの表示を、ドキュメントの構造に従ってかたちに再構成します。各子ノードが、親ノードよりイン デトされたかたちで表示されます。この機能では、タググループにて指定されたタブサイズの設定(タブませまスペースを挿入)が使用されます。

テキストビューの設定

テキストビュー設定ダイアログボックスを表示します。このダイアログボックスにより出力 ペイン、XSLT ペイン、とXQuery ペインでテキスト ビュー設定をカスタマイズすることができます。 おっこのダイアログボックスは、現在ウィンドウ内で適用されている定義済みのホットキーを表示し ています。 詳細に関しては、次を参照してください<u>テキストビュー機能</u>。
13.9 デバッグ

デバッグの開始(F11)

デバッグを開始おけよブレークポイントに到達おけはマッピングが完了するまで継続します。

デバッグの停止 (Shift + F5)

デバッグを停止します。このコマンドはデバッグモードを終了して、MapForceを標準モードに切り替えます。

ステップイン(F11)

マンピング内で単一のステップを終了するまでマンピングを実行します。マンピングでデバッガーはシーケンスの単一アイテムを生成する独立した計算の理論的なグループです。

マピングコンテキストにより、このコマンドは「左へ移動」ターゲナに移動子/ソースの親へ移動」と解釈することができます。

ステップオーバー(F10)

現在のステップが終了するまで、ませま地のシーケンスのアイテムを終了するまで、実行を継続します。このコマンドは、現在のステップの入力である計算をステップオーバーします。

ステップアウト (Shift + F11)

現在のステップの結果が使用されるまで、おけよ入力おけは使用の子ではないステップが実行されるまで実行を継続します。このコマンドは現在の計算をステップアサトします。

マピングコンテキストにより、このコマンドは「右へ移動/ターゲットへ移動 親/ノースの子へ移動」と解釈することができます。

最小ステップ(Ctrl + F11)

値が生成ませま使用されるまで、実行を継続します。このコマンドは、ステップを細分化し、接続ごとご通常2回停止します:ソースが値を生成した際、ターゲントが値を使用した際。MapForce は、マンピングの提示に従い値を計算する必要はなく、生成と使用イベントは順序に従わずよ、場合があります。

13.10 表示

注釈の表示

XML スキーマの注釈をエンポーネントウイドウニて表示します。 型の表示アイエンも有効になっている場合、両方の情報がグリンド形式で表示されます。

= F1060		
type	string	
ann.	Revision identifier	



各要素おけ属性に対してスキーマのデータ型を表示します。 注釈の表示アイエンも有効になっている場合、両方の情報がグリンド形式で表示されます。



関数タイトルコ活弧付きでライブラノ名を表示します。

ヒトの表示 🅎

マケスポインターか関数の上に配置された時に、ツールチップ内に関数の説明か表示されます。

選択されたコンポーネントの接続線の表示

すべてのマピングの接続線ませま、現在選択されているコンポーネントに関連した接続線を切り替えることができます。

ソースからターゲトへの接続線表示

以下のようは接続線の表示を切り替えることができます:

- 現在選択されているコンポーネトに直接接続されているコネクタ
- 現在選択されているコネクタで、ソースからターゲナコンポーネントまで

ズーム

ズームダイアログボックスか表示されます。ズームの培率を手動で入力することができるおか、スライダーをドラッグして、ズームの培率を変更することができます。

戻る

マピングタブニて現在開かれているマピングで、前に表示されていたマピングを表示します。

進む

マピングタブニて現在開かれているマピングで、戻るコマドにとい表示される前のマピングを表示します。

ステータスバー

メッセージウイドウの下に表示されているステータスviを表示/非表示にします。

ライブラリウィンドウ

ライブラノ関数か含まれるライブラリウインドウを表示/非表示してます。

ライブラノの管理

ライブラノの管理ウインドウを表示/非表示してます。

メッセージウィンドウ

検証出カウィボウを表示/隠します。コードの生成を行う場合、メッセージ出カウィボウか自動的に有効になり、検証結果が表示されます。

概要ウィンドウ

概要ウインドウを表示/非表示にします。長方形をドラッグすることで、マッピングの表示範囲を移動することができます

プロジェクトウィンドウ

プロジェクトウィンドで表示/非表示してます。

13.11 ツール

グロー・ジレノソース

グロー・ シレノノースの管理ダイアログボックスを表示して、グロー・ シレノノース XML ファイル内にあるグロー・ シレノノース情報を追加、編集、また 「消除することができます。詳細については グロー・ シレノノース - プロ・ディを参照ください。

アクティブな構成

現在アクティブニなっているグロー・シレノノースを、グロー・シレノノースの構成リストから選択/変更することができます。サブメニューから目的の構成を選択してくたさい。

反転マピングの作成

MapForce で現在アクティブコなっているマメピングをベースに、「反転した」マメピングを作成します。反転した結果生成されるマメピングは完成したものではなく、コンポーネント間における直接接続けが保持されるとう点に注意してくたさい。恐らく反転したマメピングは妥当なものではなく、更なる編集を行うこと無く出力タブをクリックしても、実行されません。

マピングを反転させると、ソースエンポーネントがターゲトコンポーネントになり、ターゲトコンポーネントがノースコンポーネントになります。入 カませる出力 XML インスタンスファイルがコンポーネントに割り当てられている場合、これらも交換されます。

以下のデータが保持されます

- コパーネト間の直接接続
- チェーンマピング内のエレポーネト間の直接接続
- 接続の種類:標準、混合コンテンソ、全てコピー
- パススルーコンポーネト設定
- データベースエレポーネト

以下のデータが保持されません

- 関数やフィルターなどを経由した接続は、関数などととも「削除されます
- ユーザー定義関数
- Web サゼスエンポーネント

ツールドーとウインドウの復元

ツールバーイドックウイドヴィとをデフォルトの状態にコセナします。変更を反映するコよ MapForce を再起動する必要があります。

カスタム化...

MapForce グラフィカルなユーザーインターフェイスをカスタム化することのできるダイアログボックスを開きます。これはおソール、その表示、非表示、およびメニューとキーボードのショートカルのカスタム化が含まれます。

オプション

デフォルトのMapForce 設定を変更することのできるダイアログボックスを開きます(MapForce オプションの変更を参照してくたさい)。

13.12 ウィンドウ

重ねて表示

開かれている全てのウインドウを重ねて表示されるように再配置します。

上下に並べて表示

開かれている全てのウイズウを上下に並べて表示することで、同時に表示されるように再配置します。

左右に並べて表示

開かれている全てのウインドウを左右に並べて表示することで、同時に表示されるように再配置します。

<u>1</u>

<u>2</u>

現在開かれているウイドウカ表示され、これらウイドウ間で素早い切り替えを行うことができます。 Ctrl + tab おさまCtrl + F6 キーボードショートカナを使用することでも、開かれているウイドウを切り替えることができます。

13.13 ヘルプメニュー

▼ 目次

⊟ <u>説明</u>

ヘルプケンドウの左側のペインに目次を表示した、MapForce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メント全体の概要を表示しています。目次のエントリをクト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>

へレプウイドウの左側のペインニキーワード、インデックスを表示したMapForceの画面上のヘルプマニュアルを開きます。目次は ヘルプドキュメ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「つ以上のトピックオコレクされている場合は、トピックのノストが表示されます。

- ▼ 検索
 - <u>説明</u>

ヘルプウイドウの左側のペイム検索ダイアログを表示したMapForceの画面上のヘルプマニュアルを開きます。単語を検索す るコよ、入力フィールドロ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」をクトックします。を押 します。ヘルプシステムよ、ヘルプドキュメント全体で全文検索を行い、セントしたリストを返します。アイテムを表示するためはコよアイ テムをダブルクトックします。

- ▼ ソトウェアのライセノス認証
 - <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、製品にライセンスを供与、おけよ、ライセンスの認証を行うことができます。

- ・ 無料 評価ライセンズ初めて製品のダウロードとインストールを行うと、ソストウェアライセンズ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 ライセノズをリクエスト 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。ライセノスファイルが入力された電子メールアドレスご送信されます。この手順には数分を要します。ライセノ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ノスファイルのの 次を取ります。ライセノスファイルを感照 おけよライセノスファイルへの 次を入力し「OK」をクトックします。(「ソストウエ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ノスファイルへの 次を入力する ダイアログにアクセスすることができます。)ソストウェアオ30日の間アンロックされます。
- ・ 永続的なライセンスキー:ソトウェアライセンス認証ダイアログコは永続的なライセンスキーを購入するよめのドダンが 含まれています。このドダンをクリックすると、製品の永続的なライセンスキーを購入することのできる Altova オンライン ショップは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ダを含むライセンスアイルの書式で送信されて います。3つの種類の永続的なライセンスが存在します: インストール済み、同時使用ユーザー、名前を持つユーザー。 インストール済みのライセンス は単一のエピューター上のノ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 台のエピューターのナポンコーター上のノ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 は単一のエピューター上のノ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 は単一のエピューター上のノ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 台のエピューターのナポコーター上でノ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 人のユーザーゴロ時にノトウェアを使用することを許可します。 (10N 台のエピューターコントウェアをインストールすることができます)。 名前を持つユーザーライ センスは特定のユーザーがもの異なるエピューター上でノトウェアを使用することを許可します。 ソトウェアのライ センスを認証するコム「新規のライセンスをアップロードする」をクトックして表示されるダイブログ内でライセンスアイ ルを参照、お台よライセンスファイルの いたを入力し「OK」をクトックします。

メモ 複数のユーザーライセノ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スライセンスファイルが添付ファイルとして含まれています。ライセンス ファイルは.altova licenses ファイル拡張子を有しています。

Altova 製品のライセンスを認証するコよ、以下のうちつ行ってくたさい

- 適切な場所にライセノスファイル(.altova_licenses)を保存し、ライセノ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ンドを選択し、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 やを入力し「OK」をクリックします。
- 適切な場所にライセスファイル(.altova_licenses)を保存し、Altova LicenseServer のライセスプールにアップロードします。以下を行うことができます:(i)製品のノ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)ませよ(ii) Altova LicenseServer から製品へのライセンスを割り当てます。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フトウェアをアクティブ化することができます。

- ソフトウェアライセンス認証ダイアログでライセンスを登録する方法。ダイアログ内で、「新規のライセンスをアップロード」をクリックして、ライセンスアイルを参照し選択します。「OK」をクリックしてライセンスアイルへの、なを確認します、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」をクリ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コキノストウェアのライセン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しるマンとを選択 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すめ、License Server は自動検知のすめのカライアントマシンと同じサブネットに「存在する必要があります。自動検知が作動しない場合、サーバーの名前を入力します。Altova License Server はライセンス プール内でAltova 製品のするのライセンを有している必要があります。License Server プール内に存在する場合、ソストウェアライセンス認証ダイプログ内に表示されます(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スを取得します。

Altova XMLSpy E	nterpri	ise Edition 2	2020 Software	Activation				
Thank you for choosing Altova XMLSpy Enterprise Edition 2020 and welcome to the software activation process. You can view your assigned license or select an Altova LicenseServer which provides a license for you. (NOTE: To use this software you must be licensed via Altova LicenseServer or a valid license from Altova.)								
If you do not wan	t to use	Altova Lice	nseServer dick	here to upload a lice	ense manually =>	Upload Licens	e	
To activate your s	To activate your software please enter or select the name of the Altova LicenseServer on your network.							
Altova LicenseServ	/er:	DEV02					~ [9
📮 😋 A license is	alread	y assigned	to you on Lice	enseServer at DEV02				
Name								
Company	Altov	/a GmbH						
User count	50							
License type	conc	urrent						
Expires in	703							
SMP	703 0	days left						
								Ŧ
Return Licens	e	Check o	out License	Copy Support Co	de	Save	Close	
			Coni	nected to Altova Lic	enseServer 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、マシンのライセスをLicenseServerに戻すこかでき、このライセスは、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。

ライセンスのチェックアウト

ライセンスが製品マシン上に保管されるように、ライセンスをライセンスプールから30日間チェックアナすることができます。これにより、オフラインで作業することが可能しなります。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、ライセンスは地のマシ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 ログのドダンを使用して「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お以下をおごれます:(i)ソトウェアのライセンスの認証ダイアログで「ライセンスの チェックアナ」をクトックします(上のスクリーンショナ参照)。(ii)ライセンスのチェックアナダイアログ内から、チェッ クアナの期間を選択し、「チェックアナ」をクトックします。ライセンスがチェックアナされます。ライセンスのチェックアウ ト後2つの状態が発生します:(i)ソ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」ポタイは「チェックイン」 ポタイご変更されます。「チェックイン」ポタンをクトックして、ライセンスのチェックアナ」ポタイは「チェックイン」 ポタイご変更されます。「チェックイン」ポ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すフライ ンで作業する期間をカバーするようご確認してくたさい。

メモ ライセノスのチェックアナを可能にするコま、LicenseServer上でチェックアナ機能が有効化されている必要があります。チェックアナを試みる際この機能が有効化されていない場合、エラーメッセージが表示されます。この場合、LicenseServer管理者に連絡してくたさい。

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たさい。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り、IT 管理者は、リアルタイムでや、トワーク上の全てのライセンスの概要、および、クライアトの割り当てと、クライアトのライセンスの使用状況を確認することができます。LicenseServer を使用する利点は、ですから、多数のAltova ライセンスを管理することのできる管理機能です。Altova LicenseServer は、<u>Altova Web</u>サイトで無料で提供されています。Altova LicenseServer および Altova LicenseServer を使用したライセンスの供与に関する詳細は、Altova LicenseServerドキュメートを参照してくたとい。

- ▼ 注文フォーム
 - ∃ <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タブに表示します。Altova ソフトウェアを登録することにより、最新の製品の情報が得られます。

- ▼ 更新のチェック
 - <u>説明</u>

Altova サーバーコ接続して、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。

- ザポトセター
 - □ <u>説明</u>

イターネト上にある Altova サポトセターへのレクとっています。サポトセターコは 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
 - <u>説明</u>

インターネット上にある Altova の FAQ へのレンクとなっています。 FAQ データベースは Altova のサポート スタッフィンド 常時更新されています。

- ▼ コレポーネトのダウロード
 - ∃ <u>説明</u>

イターネット上にある Altova のエポーネトダウレロードセターへのレクとなっています。このレク先が様々なコレポーネシ

トソフトウェアをダウンロードして、Altova 製品とされ使用することができます。ソフトウェアコンポーネントはXSLT やXSLF0 プロセッサーからアプリケーションサービスプラトフォームまで、幅広く提供されています。コンポーネントダウンロードセクターにてご利用しておれるノフトウェアは、通常無料でご利用しておます。

- ▼ インターネオ上のMapForce
 - <u>説明</u>

イクーネト上にある<u>Altova Webサイト</u>へのレクとなっています。<u>Altova Web サイト</u>では MapForce や関連するテクノロジーこへ で確認することができます。

- ▼ MapForce トレーニング
 - <u>説明</u>

<u>Altova Web サイト</u>でオンライトレーニングペーンを見つけることができますAltova の専門家トレーナーによるオンラインコースを 選択することができます。

- ▼ MapForce にんて
 - <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。MapForce の64ビット、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ビット・ノージョンイコンサフィックスは存在しません。

14 Code Generator

Code Generator is a MapForce built-in feature which enables you to generate Java, C++ or C# code from mapping files designed with MapForce. You can generate code not only from simple mappings with a single data source and target, but also from mappings with multiple sources and multiple targets. The result is a fully-featured and complete application which performs the mapping operation for you. Once you generate the code, you can execute the mapping by running the application directly as generated. You can also import the generated code into your own application, or extend it with your own functionality.

The primary goal of the generated code is to execute a MapForce mapping. If your mapping uses XML schemas or DTDs, you can optionally generate schema wrapper libraries for XML/DTD schemas used by the mapping, which enables you to read or write data to or from XML instances. The schema wrapper libraries enable you to work with XML data programmatically, using types generated from the schema, without too much concern for the underlying XML API. For details, see <u>Generating Code from XML Schemas or DTDs</u>.

Target Language	C++	C#	Java
Development environments	Microsoft Visual Studio 2010, 2013, 2015, 2017, 2019	Microsoft Visual Studio 2010, 2013, 2015, 2017, 2019 Target frameworks: • .NET Framework • .NET Core 3.1 • .NET 5.0	(OpenJDK を含む) Java SE JDK 7、8、おけよ11 Eclipse 4.4 or later Apache Ant (build.xml file)
XML DOM implementations	MSXML 6.0 Apache Xerces 3	System.Xml	JAXP
Database API	ADO	ADO.NET	JDBC

The generated code is expressed in C++, Java or C# programming languages.

Note: The MapForce-generated code can be considered thread-safe only if the underlying third-party XML DOM and database API libraries are. Although the thread safety of the generated code cannot be realistically proven or guaranteed, it is likely that multiple concurrent instances of the mapping code will run successfully in most cases.

C++

You can configure whether the C++ generated output should use MSXML 6.0 or Apache Xerces 3. MapForce generates complete project (.vcproj) and solution (.sln) files for all supported versions of Visual Studio (see table above). The generated code optionally supports MFC.

Prerequisites:

- 1. To compile the generated C++ code, Windows SDK must be installed on your computer.
- 2. To use Xerces 3 for C++, you will need to install and build it by following instructions from http://xerces.apache.org/xerces-c/. Make sure to add the XERCES3 environment variable that points to the directory where Xerces was installed, for example: C:\xerces-c-3.2.2. Also, the PATH

environment variable must include the path where the Xerces binaries are, for example: %XERCES3% \bin.

- 3. When building C++ code for Visual Studio and using a Xerces library precompiled for Visual C++, a compiler setting has to be changed in all projects of the solution:
 - a) Select all projects in the Solution Explorer.
 - b) On the **Project** menu, click **Properties**.
 - c) Click Configuration Properties | C/C++ | Language.
 - d) In the list of configurations, select All Configurations.
 - e) Change Treat wchar_t as Built-in Type to No (/Zc:wchar_t-).

C#

The generated C# code can be used from any .NET capable programming language, such as VB.NET, Managed C++, or J#. Project files can be generated for all supported versions of Visual Studio (see table above).

Java

The generated Java output is written against the Java API for XML Processing (JAXP) and includes an Ant build file and project files for supported versions of Java and Eclipse (see table above).

Generated output

The designated destination folder will include all the libraries and files required to execute the mapping, namely:

- A variable number of Altova libraries required by the mapping (for example, Altova function libraries, database libraries)
- A complete mapping application. When compiled and run, the application performs the mapping transformation.

Code generator templates

The generated code supports customization via a <u>template language</u> called SPL (Spy Programming Language). This can be useful, for example, when you want to customize code according to your company's writing conventions, or replace specific libraries in the generated code.

14.1 What's new ...

Version 2021 R2

• When generating C# code, you can select .NET Core 3.1 and .NET 5.0 as target frameworks from code generation options (this adds to existing support for .NET Framework projects).

Version 2020 R2

• Code generated for XML schema wrapper libraries now provides more control over element namespaces and prefixes. New methods are available to declare or override namespaces for an element, or to append an element with a prefixed namespace. See Example: Purchase Order.

Version 2020

- Added support for Visual Studio 2019
- End of support for Visual Studio 2008

Version 2018

- Added support for Microsoft Visual Studio 2013, 2015, 2017
- End of support for Visual Studio 2005 and Xerces 2.x

Version 2014

• Removal of compatibility mode option for code generation

Version 2011

• Contains bug fixes and enhancements

Version 2010 R3

- Support for Microsoft Visual Studio 2010
- Support for MSXML 6.0 in generated C++ code
- Support for 64-bit targets for C++ and C# projects

Version 2010

• Enumeration facets from XML schemas are now available as symbolic constants in the generated classes (using 2007r3 templates)

Version 2009 sp1

 Apache Xerces version 3.x support added (older versions starting from Xerces 2.6.x are still supported)

Version 2009

• The generated mapping implementation was redesigned to support sequences and grouping. The API has not changed.

Version 2008 R2

- Support for generation of Visual Studio 2008 project files for C# and C++ has been added
- Generated MapForce mapping code in C# and Java can use readers/writers, streams, strings or DOM documents as sources and targets

Version 2008

- The new 2007 R3-style SPL templates have been further enhanced:
 - It is now possible to remove single elements
 - Access to schema metadata (e.g. element names, facets, enumerations, occurrence, etc.) is provided
 - Complex types derived by extension are now generated as derived classes

Version 2007 R3

Code Generator has been redesigned for version 2007 release 3 to simplify usage of the generated code, reduce code volume and increase performance.

- Handling of XML documents and nodes with explicit ownership, to avoid memory leaks and to enable multi-threading
- New syntax to avoid name collisions
- New data types for simpler usage and higher performance (native types where possible, new null handling, ...)
- Attributes are no longer generated as collections
- Simple element content is now also treated like a special attribute, for consistency
- New internal object model (important for customized SPL templates)
- Compatibility mode to generate code in the style of older releases
- Type wrapper classes are now only generated on demand for smaller code

14.2 Generating C++ code

You can generate C++ code for Visual Studio 2010, 2013, 2015, 2017, 2019. The generated code includes **.sln** and **.vcproj** files for Visual Studio. Note the following when generating code:

- Certain MapForce features are not supported in generated program code. For details, see <u>Supported</u> <u>features in generated code</u>.
- You can generate code either from a single mapping design (.mfd), or from a mapping project (.mfp). If you generate code from a single mapping, the resulting application executes the respective mapping transformation. If you generate code from a MapForce project (.mfp) which includes multiple mappings, the resulting application executes in bulk all mappings included in the project.
- You can change the general code generation options from the Tools | Options menu, Generation tab. See <u>Code Generator Options</u>.
- You can change the name of the generated mapping application and other settings from the File | Mapping settings menu. The default application name is Mapping. See also <u>Changing the Mapping</u> <u>Settings</u>.
- For information about handling paths in generated code, see <u>Paths in Various Execution</u> <u>Environments</u>.
- If your mapping contains database components, you can view database specific settings by clicking a database component, and then selecting the menu option Component | Properties. See <u>Database</u> <u>Component Settings</u>.
- Support for database connections varies by platform and there are connection kinds that are not supported on all platforms. If your mapping connects to a database, choose a database connection that is compatible with the target environment for which you generate code. For details, see <u>Database</u> <u>mappings in various execution environments</u>.

A typical C++ solution generated by MapForce includes the following:

- Several Altova-signed libraries required by the mapping (all prefixed with Altova).
- The main mapping project (in this example, **Mapping**), which includes the mapping application and dependent files.



Sample C++ solution generated with MapForce

This section includes the following topics:

- <u>Generating code from a mapping</u>
- Generating code from a mapping project
- Building the project
- Running the application

14.2.1 Generating code from a mapping

To generate C++ code from a mapping design file (.mfd):

- 1. Review and select the <u>code generator options</u>.
- 2. On the File menu, click Generate code in | C++.
- 3. Select a destination directory for the generated files, and then click OK to confirm. The result of code generation (error or success message) is displayed in the Messages window.

The default name of the generated application is **Mapping**. If required, you can change this, and other settings, from the <u>Mapping Settings</u> dialog box.

14.2.2 Generating code from a mapping project

To generate code from a mapping project (.mfp):

- 1. If you haven't done so already, open the mapping project in MapForce.
- 2. Right-click the project in the Project window, and then click Properties.

Project Settings		X
Project Name:	Project01	
Project Directory:	C:\Project01\	
Output Settings		
Output Name:	Project01	
Output Directory:	C:\Project01\output\	Browse
Language:	Java 🔻	
Java Settings		
B <u>a</u> se package name:	com.mapforce	
	ОК	Cancel

- 3. Review and change the project settings if required (in particular, ensure that the target language and the output directory are set correctly), and then click **OK**.
- 4. On the Project menu, click Generate code for the Entire Project.

The progress and result of the code generation process (error or success message) is displayed in the Messages window.

By default, the name of the generated application is the same as the project name. If the project name contains spaces, these are converted to underscores in the generated code. By default, code is generated in the same directory as the MapForce project, in the **output** sub-directory.

To change the output directory and the name of the project, click the Project in the Project window, and then select **Project | Properties** from the menu. If your MapForce project contains folders, you can change the code generation settings for each individual folder (right click on the folder, and then select **Properties**). Otherwise, all project folders inherit the settings from the MapForce project.

14.2.3 Building the project

Once you generated the C++ code, building it in Visual Studio is the next step. To build the generated code:

1. Open the generated solution (.sln) file in Visual Studio.

By default, the name of the solution file is **Mapping.sln**, and it is located in the **Mapping** subdirectory relative to the directory where you saved the generated code. If you changed the application name from the mapping

settings, then the name of the .sln file is changed accordingly. For example, if you changed the application name to **MyApplication**, then the solution file is called **MyApplication.sln**, and it is located in the **MyApplication** subdirectory.

2. On the Build menu, click Configuration Manager.

Configuration Manager					? 💌	
Active solution configuration:		Active solution platform:				
Debug	Debug 🗸		Win32 👻			
Debug		ploy):				
Unicode Debug			Platform	Build	t l	
<pre>Vnicode Release <new></new></pre>			Win32		V	
<edit></edit>			Win32		V	
AltovaFunctions	Debug		Win32		V	
AltovaXML	Debug		Win32		V	
Mapping	Debug	-	Win32		V	
					Close	

- 3. Select the required build configuration (Debug, Release, Unicode Debug, Unicode Release). Note that only Unicode builds support the full Unicode character set in XML and other files. The non-Unicode builds work with the local codepage of your Windows installation.
- 4. On the **Build** menu, click **Build Solution**.

14.2.4 Running the application

Once you compile the Visual Studio project, a command-line application is produced, called **Mapping.exe**. (Note that if you changed the application name from the mapping settings, then the executable name is changed accordingly.)

You can locate the mapping application in one of the following subdirectories relative to the .sln file, depending on the build option you chose:

- Debug
- Release
- Unicode Debug
- Unicode Release

To run the application, open a command prompt, change the current directory to the path of the executable, and run it, for example:

C:\codegen\DB_CompletePOcpp\Mapping\Debug>Mapping.exe Mapping Application Finished

C:\codegen\DB_CompletePOcpp\Mapping\Debug>_

14.3 Generating C# code

You can generate .NET C# code for any mapping or mapping project where all the mapping components are supported in C#. For an overview of components supported in each language, including C#, see <u>Supported</u> <u>features in generated code</u>. Note that MapForce typically issues notification or warning messages whenever you attempt to add components that are not supported in a specific language.

When generating C# code, you can target not only the .NET Framework platform but also the .NET Core 3.1 and .NET 5.0 platforms. If you choose the latter, the result is a .NET (Core) cross-platform desktop application that can run on Windows, macOS, and Linux.

Note the following when generating code:

- Certain MapForce features are not supported in generated program code. For details, see <u>Supported</u> <u>features in generated code</u>.
- You can generate code either from a single mapping design (.mfd), or from a mapping project (.mfp). If you generate code from a single mapping, the resulting application executes the respective mapping transformation. If you generate code from a MapForce project (.mfp) which includes multiple mappings, the resulting application executes in bulk all mappings included in the project.
- You can change the general code generation options from the Tools | Options menu, Generation tab. See <u>Code Generator Options</u>.
- You can change the name of the generated mapping application and other settings from the File | Mapping settings menu. The default application name is Mapping. See also <u>Changing the Mapping</u> <u>Settings</u>.
- For information about handling paths in generated code, see <u>Paths in Various Execution</u> <u>Environments</u>.
- If your mapping contains database components, you can view database specific settings by clicking a database component, and then selecting the menu option Component | Properties. See <u>Database</u> <u>Component Settings</u>.
- Support for database connections varies by platform and there are connection kinds that are not supported on all platforms. If your mapping connects to a database, choose a database connection that is compatible with the target environment for which you generate code. For details, see <u>Database</u> <u>mappings in various execution environments</u>.

If your mapping uses database components, make sure to create the database connection using database drivers compatible with the platform on which you intend to run the generated program. For example, ADO database connections and some ODBC drivers are not supported on Linux.

A typical C# solution generated by MapForce includes the following:

- Solution (.sln) and project (.csproj) files that can be opened in Visual Studio.
- Several Altova-signed libraries required by the mapping (all prefixed with Altova).
- The main mapping project (in this example, **Mapping**), which includes the mapping application and dependent files.



Sample C# solution generated with MapForce

14.3.1 Selecting the target platform

Before generating C# code, you can set the target platform at program level, as follows:

- 1. On the **Tools** menu, click **Options**.
- 2. Click the **Generation** tab.
- 3. In the "C# settings" group, select a value from the drop-down list, as applicable.

C <u>+</u> + Settings Project File:	C <u>#</u> Settings Project File:
Microsoft® Visual Studio® 2015	✓ Microsoft® Visual Studio® 2015
XML Library ● <u>M</u> SXML 6 ○ <u>X</u> erces 3.x	Microsoft® Visual Studio® 2010 Microsoft® Visual Studio® 2013 Microsoft® Visual Studio® 2015 Microsoft® Visual Studio® 2017 Microsoft® Visual Studio® 2019
Library Type Static Library (.LIB)	Microsoft® .NET Core 3.1 Microsoft® .NET 5.0
O Dynamic-Link Library (.DLL)	Wrapper Classes
MFC Support	Generate Wrapper Classes

Select the option **Microsoft .NET Core 3.1** or **Microsoft .NET 5.0** to generate a Visual Studio solution targeting the respective platforms. If you need to target the **.NET Framework** platform for a specific Visual

Studio version, select any of the **Microsoft Visual Studio 2010-2019** options—in this case, the generated solution will target the .NET Framework version corresponding to the respective Visual Studio version.

The **Generate Wrapper Classes** check box, if selected, creates additional projects in the generated code, one for each XML schema or DTD component present on the mapping. The generated wrapper projects enable you to manipulate programmatically the respective XML instances outside of the mapping. The wrapper libraries are optional and independent of the mapping program, although you can use them from the mapping program if necessary. For more information about wrapper classes, see <u>Generating Code from XML Schemas</u> or DTDs.

14.3.2 Generating code from a mapping

To generate C# code from a mapping design file (.mfd):

- 1. If you haven't done so already, create a new mapping or open an existing one in MapForce.
- 2. Select C# as mapping transformation language. To do this, either click the toolbar button or select the menu command **Output | C# (Sharp)**.
- 3. Set or change the C# code generation options as described in Selecting the target platform.
- 4. Select the menu command File | Generate code in | C# (Sharp). You are now prompted to select a target directory.
- 5. Browse for the directory of your choice (for example, "C:\codegen\ConvertProducts") and click OK.

The code generation outcome is displayed in the Messages window, for example:

Mes	sages	×
F	∀▼▲□⊡▣९ぷぷХ	
	ConvertProducts.mfd: Mapping validation successful 0 error(s), 0 warning(s)	
	ConvertProducts.mfd: Code generation completed successfully 0 error(s), 0 warning(s)	
	Olick to open: C:\codegen\ConvertProducts/Mapping/Mapping.sln	
2 S		
÷		-

Optionally, click the link in the Messages window to open the generated solution in Visual Studio.

Note that, if you chose to generate code targeting .NET 5 or .NET Core platforms, you need the respective SDKs and a compatible Visual Studio version. For the download package applicable to your operating system and platform, refer to the Microsoft website (<u>https://dotnet.microsoft.com/download</u>).

To double-check the project's target framework from Visual Studio, right-click the "Mapping" project in the Solution Explorer, select **Properties**, and observe the value displayed in the **Target Framework** drop-down list, for example:

Mapping 👳 🗙		- \$
Application	Configuration: N/A Platform: N/A	
Build		
Build Events	Assembly <u>n</u> ame: Defau <u>l</u> t namespace:	
Package	Mapping Mapping	
Debug	Target framework: Output type:	
Signing	.NET Core 3.1 \checkmark Console Application	
Code Analysis	Startup <u>o</u> bject:	
Resources	(Not set) \checkmark	

The default name of the generated application is **Mapping**. Optionally, you can change this and other settings before generating code, from the <u>Mapping Settings</u> dialog box.

14.3.3 Generating code from a mapping project

You can generate C# code from a mapping project (.mfp) that consists of multiple mapping design (.mfd) files. Note that all mapping design files in the project must qualify for C# generation, that is, all their components must be supported in C#, as described in <u>Supported features in generated code</u>.

To generate code from a mapping project (.mfp):

- 1. If you haven't done so already, open the mapping project in MapForce.
- 2. Right-click the project in the Project window, and then click Properties.

Project Settings		×
Project Name:	MapForceExamples	
Output Settings	C. Users (altova (Documents (Altova (Haproice2021) Mapr	
Output Name:	MapForceExamples	
Output Directory:	C: \Users\altova \Documents\Altova \MapForce2021 \MapF Browse	
Language:	C# ~	
Java Settings		
Base package name:	com.mapforce	
	OK Cancel	

- 3. Review and change the project settings if required (in particular, ensure that the target language and the output directory are set correctly), and then click **OK**.
- 4. On the **Project** menu, click **Generate code for the Entire Project**.

The progress and result of the code generation process (error or success message) is displayed in the Messages window.

By default, the name of the generated application is the same as the project name. If the project name contains spaces, these are converted to underscores in the generated code. By default, code is generated in the same directory as the MapForce project, in the **output** sub-directory.

To change the output directory and the name of the project, click the Project in the Project window, and then select **Project | Properties** from the menu. If your MapForce project contains folders, you can change the code generation settings for each individual folder (right click on the folder, and then select **Properties**). Otherwise, all project folders inherit the settings from the MapForce project.

14.3.4 Building .NET Framework projects

If your generated C# targets the .NET Framework platform, the easiest way to build the project is as follows:

1. Open the generated solution (.sln) file in Visual Studio.

By default, the name of the solution file is **Mapping.sln**, and it is located in the **Mapping** subdirectory relative to the directory where you saved the generated code. If you changed the application name from the mapping settings, then the name of the .sln file is changed accordingly. For example, if you changed the application name to **MyApplication**, then the solution file is called **MyApplication.sln**, and it is located in the **MyApplication** subdirectory.

- 2. On the Build menu, click Configuration Manager.
- 3. Select the required build configuration (Debug, Release).
- 4. On the **Build** menu, click **Build Solution**.

Running the application

Once you compile the Visual Studio project, a command-line application is produced, called **Mapping.exe**. Note that if you changed the application name from the <u>Mapping Settings</u> dialog box, then the executable name is changed accordingly.

You can find the mapping application in one of the following subdirectories relative to the .sln file, depending on the build option you chose earlier:

- bin\Debug
- bin\Release

To run the application, open a command prompt, change the current directory to the path of the executable, and run it, for example:

C:\codegen\DB_CompletePOcs\Mapping\bin\Release>Mapping.exe Mapping Application Connecting to CustomersAndArticles database...

Finished

C:\codegen\DB_CompletePOcs\Mapping\bin\Release>

14.3.5 Building .NET 5 and .NET Core projects

To build and run the generated .NET Core 3.1 and .NET 5.0 solution, you must have the respective SDK installed. For the download package applicable to your operating system and platform, refer to the Microsoft website (<u>https://dotnet.microsoft.com/download</u>). After downloading and installing the required package, yon can easily view the installed SDKs by entering the following command at the command prompt:

dotnet --list-sdks

If you have the .NET Core 3.1 or .NET 5.0 SDK, you can proceed to building the generated code, either at the command line or in Visual Studio. If using Visual Studio, note that, at the time of this writing (December 2020), the .NET Core 3.1 and .NET 5.0 SDKs are compatible with Visual Studio 2019 (v16.7).

Building at the command line

To build the generated code at the command line:

 Open a command prompt window and change directory to the one where the mapping code was generated. For example, if you selected C:\codegen\ConvertProducts as target directory and the default mapping settings, run:

```
cd C:\codegen\ConvertProducts
```

2. Run the following command:

```
dotnet build Mapping\Mapping.sln --no-incremental --configuration Release --
verbosity normal --output Mapping\bin\release
```

Note: The path to the mapping solution and its name may be different if you have changed the **Application Name** property from the <u>Mapping Settings</u> dialog box.

The outcome for a successful build could be similar to the one below:

Command Prompt	-	- [- ×	(
Copying file from "c:\codegen\ConvertProducts\Mapping\obj\Release\ne ing.pdb" to "c:\codegen\ConvertProducts\Mapping\bin\release\Mapping.pdb". 2>Done Building Project "c:\codegen\ConvertProducts\Mapping\Mapping.cspr get(s)).	tcore oj" (≥app3. (Rebui	1∖Mapp ld tar	^
1>Done Building Project "c:\codegen\ConvertProducts\Mapping\Mapping.sln" (s)).	(Ret	build	target	
Build succeeded. 0 Warning(s) 0 Error(s)				
Time Elapsed 00:00:12.09				
c:\codegen\ConvertProducts>				
				\mathbf{v}

Building with Visual Studio

To build the generated code in Visual Studio:

- 1. Double-click **Mapping.sln** (or the respective custom solution name) to open it in Visual Studio.
- 2. On the Build menu, click Configuration Manager.
- 3. Select the required build configuration (Debug, Release) and solution platform (x64, x86, Any CPU).
- 4. On the Build menu, select Build Solution (Ctrl+Shift+B).

Running the generated program

After you build the generated code, a **Mapping.exe** (or a program with a custom name if configured) is produced in the project's output directory. The path of the output directory depends on the build configuration you have chosen earlier (Debug, Release).

If you build the generated code on Linux, the generated executable will be called simply **Mapping**, without any extension.

On Windows, you can execute the program by double-clicking the executable in Windows Explorer or by calling the executable from a command prompt window. On other platforms, you will need to call the executable from the terminal with a command like:

./Mapping

Note that, if the mapping has references to any input or output files on the disk, the program will execute successfully only if all the paths can be resolved. For example, if you configured the mapping so that a component should look for some input file in the same directory using a relative path, then the respective file must be present in the program's execution directory. For information about handling paths at mapping design time, see <u>Using Relative and Absolute Paths</u>.

14.4 Generating Java code

You can generate program code for (OpenJDK を含む) Java SE JDK 7、8、おは11. Note the following when generating code:

- Certain MapForce features are not supported in generated program code. For details, see <u>Supported</u> <u>features in generated code</u>.
- You can generate code either from a single mapping design (.mfd), or from a mapping project (.mfp). If you generate code from a single mapping, the resulting application executes the respective mapping transformation. If you generate code from a MapForce project (.mfp) which includes multiple mappings, the resulting application executes in bulk all mappings included in the project.
- You can change the general code generation options from the **Tools | Options** menu, **Generation** tab. See <u>Code Generator Options</u>.
- You can change the name of the generated mapping application and other settings from the File | Mapping settings menu. The default application name is Mapping. See also <u>Changing the Mapping</u> <u>Settings</u>.
- For information about handling paths in generated code, see <u>Paths in Various Execution</u> <u>Environments</u>.
- If your mapping contains database components, you can view database specific settings by clicking a database component, and then selecting the menu option Component | Properties. See <u>Database</u> <u>Component Settings</u>.
- Support for database connections varies by platform and there are connection kinds that are not supported on all platforms. If your mapping connects to a database, choose a database connection that is compatible with the target environment for which you generate code. For details, see <u>Database</u> <u>mappings in various execution environments</u>.

A typical Java project generated by MapForce includes the following:

- Several Altova-signed Java packages required by the mapping (all prefixed with com.altova).
- The **com.mapforce** package, which includes the mapping application and dependent files (as shown below, it is possible to change the name of this package). The two most important files in this package are as follows:
 - The Java mapping application as a dialog application (MappingApplication.java).
 - The Java mapping application as a console application (MappingConsole.java).
- A build.xml file which you can execute with Apache Ant to compile the project and generate JAR files.



Sample MapForce-generated Java application (Eclipse IDE)

This section includes the following topics:

- <u>Generating code from a mapping</u>
- <u>Generating code from a mapping project</u>
- Handling JDBC references
- <u>Building the project with Ant</u>
- Example: Generate and Run Java Code

14.4.1 Generating code from a mapping

To generate Java code from a mapping design file (.mfd):

- 1. Review and select the <u>code generator options</u>.
- 2. On the File menu, click Generate code in | Java.
- 3. Select a destination directory for the generated files, and then click **OK** to confirm. The result of code generation (error or success message) is displayed in the Messages window.

The default name of the generated application is **Mapping**, and the default name of the base package is **com.mapforce**. If required, you can change these from the <u>Mapping Settings</u> dialog box.

14.4.2 Generating code from a mapping project

To generate code from a mapping project (.mfp):

- 1. If you haven't done so already, open the mapping project in MapForce.
- 2. Right-click the project in the Project window, and then click Properties.

Project Settings		X
Project Name:	Project01	
Project Directory:	C:\Project01\	
Output Settings		
Output Name:	Project01	
Output Directory:	C:\Project01\output\	Browse
Language:	Java 🔻	
Java Settings		
B <u>a</u> se package name:	com.mapforce	
	ОК	Cancel

- 3. Review and change the project settings if required (in particular, ensure that the target language and the output directory are set correctly), and then click **OK**.
- 4. On the **Project menu**, click Generate code for the Entire Project.

The progress and result of the code generation process (error or success message) is displayed in the Messages window.

By default, the name of the generated application is the same as the project name. If the project name contains spaces, these are converted to underscores in the generated code. By default, code is generated in the same directory as the MapForce project, in the **output** sub-directory.

To change the output directory and the name of the project, click the Project in the Project window, and then select **Project | Properties** from the menu. If your MapForce project contains folders, you can change the code generation settings for each individual folder (right click on the folder, and then select **Properties**). Otherwise, all project folders inherit the settings from the MapForce project.

14.4.3 Building the project with Ant

Apache Ant is an open source tool which automates building and compilation of Java projects (see http://ant.apache.org/manual/). Ant works with build files (such files define the sources and targets from which code must be compiled, as well as any specific build options). Since any MapForce-generated project includes a **build.xml** file recognized by Ant, you can easily build MapForce-generated projects with Ant.

Ant may be available on your system either as a standalone installation, or bundled within Eclipse (or other Java IDEs). For instructions on how to install Ant on your system, see http://ant.apache.org/manual/. For instructions on how to use Ant in Eclipse, refer to the Eclipse documentation.

You can quickly check whether the standalone version of Ant (not the one bundled with Eclipse) is available on your system by opening a command prompt and typing ant at the command line. When Ant is already available, the resulting message will be similar to: Buildfile: buildxml does not exist! This message indicates that Ant is installed and it is attempting to build a **build.xml** file, but the latter does not exist in the current directory. If you run Ant from a directory which includes a **build.xml** file, Ant executes the **build.xml** file instead, with whatever build options are defined in it.

To build a MapForce-generated Java project with Ant:

- 1. Open a command prompt and navigate to the directory where the Java project was generated (note that the directory must contain the **build.xml** file).
- 2. At the command prompt, enter ant. This will compile and execute the Java code according to the default options defined in the **build.xml** file, for example:



To generate a JAR file with Ant:

• At the command prompt, enter ant jar.

For help with Ant command syntax and options, enter ant -help at the command line. For a demo, see Example: Generate and Run Java Code.

14.4.4 Handling JDBC references

If the mapping connects to a database through JDBC, ensure that the JDBC drivers used by the mapping are installed on your system. To view the current JDBC settings of any database component in MapForce, click it, and then select **Component | Properties** from the menu. For more information, see <u>Creating a JDBC</u> connection.

If the mapping uses a non-JDBC database connection, the connection may be converted to JDBC during Java code generation, to provide compatibility in a Java environment. For example, ADO, ADO.NET, and ODBC connections are converted to JDBC connections when you generate Java code from the mapping. For details, see <u>Database mappings in various execution environments</u>.

If you test run the generated Java application in Eclipse, the JDBC driver may need to be added as a "classpath" entry to the current run configuration. Otherwise, test running the application in Eclipse could fail with an error message such as: java.lang.ClassNotFoundException: com.mysql.jdbc.Driver. Taking MySQL database as an example, you can add the JDBC driver as a dependency in Eclipse as follows:

- 1. Generate the Java code from MapForce and import the project into Eclipse using the menu command **File | Import | Existing Projects into Workspace**.
- 2. On the **Run** menu, click **Run Configurations**. On the left side of the dialog box, under "Java Application", there are two configurations called **MappingApplication** and **MappingConsole**.
- 3. Click the desired configuration (for example, MappingApplication).
- 4. On the Dependencies tab, click Classpath entries, and then click Add External JARs.
- 5. Browse for the .jar file of the JDBC driver, for example:

Run Configurations			
Create, manage, and run configurations Run a Java application			
Image: Second system Image: Second system Image: Secon	Name: MappingApplication Image: Main (x)= Arguments Image: JRE Image: Dependencies Image: Source Dependencies Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Image: Source Imag		

6. Click **Run** to run the program with the database JDBC driver added as a dependency.

If you get the JDBC driver error above when test running the Ant **build.xml** file, add the JDBC driver to the classpath of the "test" task. The following is an example of an Ant "test" task that includes a reference to the .jar file of the JDBC driver.

```
<target name="test" depends="compile">
<java classpath="C:\codegen\java\mysql_mapping"
classname="com.mapforce.MappingConsole" fork="true" failonerror="true">
<classpath>
<pathelement path="${classpath}"/>
<pathelement path="${classpath}"/>
</classpath>
<arg line="${cmdline}"/>
</java>
</target>
```

Note: The example above assumes that the .jar file exists at the following path: C:\jdbc\mysql\mysqlconnector-java-5.1.16-bin.jar. Make sure to change the .jar file path as applicable to your environment.

Including JDBC drivers in the application's manifest

If you build JAR files from the generated Java application, add a reference to the database driver in the "manifest" section of the **build.xml** file. This ensures that the reference to the database driver is available in the manifest (**MANIFEST.MF**) file after you build the project.

To add the database reference to the manifest file:

- 1. Locate the manifest element in the build.xml file,
- 2. Add a new element called **attribute** where the "name" attribute is "Class-Path" and the "value" attribute is the name of the .jar file. For example, for MySQL 5.1.16, the new element could look as follows:

<attribute name="Class-Path" value="mysql-connector-java-5.1.16-bin.jar"/>

The **manifest** element of the **build.xml** file now looks as follows:

3. Copy the JAR file of the JDBC driver to the folder that contains the JAR file of the generated application.

14.4.5 Example: Generate and run Java code

This example walks you through the steps required to generate a Java application with MapForce, and compile it outside of MapForce using the Eclipse Integrated Development Environment (IDE) and Apache Ant. After completing this example, you will have created and compiled a complete Java application which executes one of the mapping samples available by default in MapForce.

If you can already compile successfully other Java applications with Eclipse and Ant, there are no special requirements to run this example. Otherwise, note the following prerequisites:

- Java Development Kit (JDK), Eclipse, and Ant must be installed on your system. Eclipse typically
 includes a bundled version of Ant but you can also install Ant separately, see also <u>Building the project</u>
 with Ant.
- To run Eclipse with OpenJDK, the PATH environment variable typically needs to include the path to the JDK bin directory (for example, C:\Java\jdk-11.0.1\bin).
- If you run Apache Ant standalone like in this example, the PATH environment variable should also include the path to Ant **bin** directory (for example, **C:\apache-ant-1.10.5\bin**), so that you can conveniently run Ant without having to type the full path to the executable at the command line.

This example uses the following configuration:

- MapForce 64-bit
- OpenJDK 11.0.1 64-bit
- Eclipse IDE for Java Developers, version 2019-03 (4.11.0), 64-bit
- Apache Ant 1.10.6 (installed as standalone) and Apache Ant 1.10.5 (bundled with Eclipse)

Step 1: Generate Java code

To generate the Java code from MapForce:

- 1. On the **File** menu, click **Open**, and browse for the **CompletePO.mfd** mapping available in the <マイド キュメント>¥Altova¥MapForce2021¥MapForceExamples¥directory.
- 2. On the **Output** menu, click **Java**. This changes the transformation language to Java.
- 3. On the **File** menu, click **Generate code in | Java**. When prompted, browse for the directory where the Java project should be saved. For the purpose of this example, you might want to generate Java code to **C:\codegen\java\CompletePO**. You will need to refer to this path in subsequent steps.

Step 2: Import the project into Eclipse

To import the project into Eclipse:

- 1. If you haven't done so already, run Eclipse and switch to the default Java perspective using the menu command **Window | Open Perspective**.
- 2. On the File menu, click Import, and then select General | Existing Projects into Workspace.

Import	_		\times
Select Create new projects from an archive file or directory.		Ľ	5
Select an import wizard:			
type niter text			
✓ (⇒ General)			~
_ 🕼 Archive File			
😫 Existing Projects into Workspace			
Eile System			
Preferences			
Projects from Folder or Archive			
Projects from Folder of Archive			¥
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>		Cance	I

3. Click Next.

🖨 Import						×
Import Projects Select a directory to search for existing Eclipse projects.						
 Select root directory: Select archive file: Projects: 	C:\codegen	ı∖java∖Comple	tePO	✓	B <u>r</u> owse	
Mapping (C:\code	egen\java\Co	ompletePO)		[<u>S</u> elect A <u>D</u> eselect R <u>e</u> fresl	All All
Options Searc <u>h</u> for nested pro <u>C</u> opy projects into w Cl <u>o</u> se newly imported H <u>i</u> de projects that alr	jects orkspace d projects up eady exist in	on completior the workspace	1			
Working sets	ing sets		~		Ne <u>w</u> S <u>e</u> lect	
?	ack	<u>N</u> ext >	<u>F</u> inish		Cance	I

4. Browse for the folder where you have previously saved the generated code, and then click **Finish**. The Java project created by MapForce is now available in the Package Explorer view. If you cannot see the Package Explorer view, display it using the menu command **Window | Show View | Package Explorer**.



Step 3: Run the project as dialog application

To run the Java project as a GUI application:

- 1. In the Package Explorer view of Eclipse, click the **MappingApplication.java** file available in the **com.mapforce** package.
- 2. On the Run menu, click Run As | Java application.
- 3. On the MapForce application window, click **Start** to execute the mapping.
| Mapforce Application | | | _ | | × |
|--|--|----------|---|--|---|
| THIS APPLICATION WAS
http://www.altova.com/
Please check the input a | GENERATED BY MapForce 2020r2
mapforce
and output files, and press the Start bu | itton | | | |
| Source instance of Customers.xsd: | | | | | ^ |
| C:/Users/altova/Documents/Altova/Ma | pForce2020/MapForceExamples/Custo | mers.xml | | | |
| Source instance of Articles.xsd: | | | | | |
| C:/Users/altova/Documents/Altova/Ma | pForce2020/MapForceExamples/Article | es.xml | | | |
| Source instance of ShortPO.xsd: | | | | | |
| C:/Users/altova/Documents/Altova/Ma | pForce2020/MapForceExamples/ShortF | PO.xml | | | |
| Instance of CompletePO.xsd: | | | | | |
| CompletePO.xml | | | | | |
| | | | | | |
| | | | | | ~ |
| Start | | | | | |
| Started | | | | | > |
| Loading com.altova.io.File | Input@2d6a94e7 | | | | |
| Loading com.altova.io.File | Input@76a8c72d | | | | |
| Loading com.altova.io.File | Input@lbe209af | | | | |
| Finished | | | | | 5 |
| | | | | | * |

If Eclipse encounters system configuration or run-time errors, you will be prompted, see also <u>Handling JDBC</u> <u>references</u>. Otherwise, the Java application executes the mapping transformation and generates the **CompletePO.xml** at the output path (in this example, the output path is **C:\codegen\Java\CompletePO**).

Step 4: Run the project as console application

To run the Java project as a console application:

- 1. In the Package Explorer view of Eclipse, click the **MappingConsole.java** file available in the **com.mapforce** package.
- 2. On the Run menu, click Run As | Java application.



If Eclipse detects system configuration or run-time errors, you will be prompted, see also <u>Handling JDBC</u> <u>references</u>. Otherwise, the Java application executes the mapping transformation and generates the **CompletePO.xml** at the output path (in this example, the output path is **C:\codegen\Java\CompletePO**).

Step 5: Build the JAR file

If Apache Ant is bundled with Eclipse, you can build the JAR file directly from the Eclipse IDE. Otherwise, to build the JAR file at the command line outside of Eclipse, Apache Ant must be installed separately.

To build the .jar file with Ant in Eclipse:

1. In the Package Explorer view of Eclipse, right-click the **build.xml** file, and point to the **Run As** menu. In the **Run As** menu, two possible options to run the Ant build file are displayed. If you choose the first option, Eclipse launches the Ant build with the default settings. If you choose the second option, you can change the settings of the Ant build before launching it. Select the second option.

0	Run As	米	1 Ant Build	Alt+Shift+X, Q
*	Debug As	*	2 Ant Build	
	Team	•	External Tools Configurations	
	Compare With 2	-		

2. Click to enable the targets that you wish to include in the Ant build. In this example, the targets **test** and **jar** are selected.

Edit Configuration	_		
Edit configuration and launch.		0_	
Run an Ant build file.			
			-
Name: Mapping build.xml			
📄 Main 🔗 Refresh 🚮 Build 🦟 Tar	gets 🛛 🔩 Classpath 🕬 Properties 🛋 JRE 📧 Environment 🗉	<u>Common</u>	
Check targets to execute:			
Name	Description		
🗌 🗆 🛞 compile			
🗹 🎯 test [default]			
Clean			
2 out of 5 selected		,	
<u>S</u> ort targets			
<u>H</u> ide internal targets not selected for	execution		
Target execution order:			
test, jar	^	<u>O</u> rder	
	×		
			-
	Re <u>v</u> ert	Appl <u>y</u>	
0	Dem	Class	
\bigcirc	Kun	Close	

3. Click **Run**. Eclipse executes the Ant build file and displays the result in the Console view.

To build the JAR file at the command line:

- 1. Make sure that the directory where Apache Ant is installed (for example, C:\apache-ant-1.10.5\bin) is added to the PATH environment variable.
- 2. Open a Command Prompt window and change the current directory to the one where you generated Java code (in this example, C:\codegen\java\CompletePO). This is also the directory where the **build.xml** file was generated.
- 3. At the command line, enter:

ant clean jar

Ant runs the "clean" and "jar" targets from the **build.xml** file (including any dependent targets), and displays the outcome at the command line:



14.5 Integrating MapForce-Generated Code

MapForce-generated code can be integrated, or adapted to your specific application, even though the result of code generation is a complete and fully-functioning application. Some typical scenarios where you might want to change the generated code are as follows:

- Define custom source or target files for the mapping application
- Add custom error handling code
- In C# or Java generated code, you can also change the data type of the mapping input programmatically (for example, from string to stream).

This section provides instructions on how to achieve these goals, based on the **DB_CompletePO.mfd** sample mapping available in the <マイドキュメント>¥Altova¥MapForce2021¥MapForceExamples¥directory.



DB_CompletePO.mfd mapping sample in MapForce

As illustrated above, the sample mapping consists of two sources and one target:

- ShortPO.xml is a source XML file
- CustomersAndArticles.mdb is a source database
- CompletePO.xml is the target XML file.

In the generated code, these sources and targets will translate to two input and one output parameters supplied to the run method which executes the mapping (as described in the subsequent topics). For now, note the following basic points about code generation:

- The number of source and targets in the mapping design corresponds to the number of mapping parameters to the <u>run</u> method in the generated code.
- If you change the number of sources or targets of the mapping, then you will need to re-generate the code accordingly.
- If you make changes to the generated code, and then re-generate the code at the same location, all changes will be overwritten.

If a mapping includes database components, the generated **run** method includes the database connection object at the appropriate location. For example, if the mapping uses three sources (text content, XML content and a database) to map to a single output file, MapForce generates the following **run** method:

```
Java
void run(Input in1, Input in2, java.sql.Connection dbConn, Output out1);
```

The argument order is important. As you will see in the subsequent examples, you can modify dbConn parameters, or use the default parameters generated by MapForce when integrating your code.

14.5.1 Java example

This example uses Eclipse as Java IDE. To begin, generate Java code from the **DB_CompletePO** sample mapping available in the <マイドキュメナト>¥Altova¥MapForce2021¥MapForceExamples¥directory, and then import the project into Eclipse.



Sample MapForce-generated Java application (Eclipse IDE)

To edit the generated Java console application, locate the main(String[] args) method of your generated application (see the screen shot above). If you did not change the default base package name before generating code, this method is in the MappingConsole class of the com.mapforce package. Otherwise, it is in the MappingConsole class of your custom defined package.

To edit the generated Java dialog application, locate the place in the code where the run method is invoked from your generated application. If you did not change the default base package name before generating code, the run method is invoked from the class called MappingFrame.java of the com.mapforce package.

The following code sample illustrates an extract from the main method in the generated Java console application. The mapping sources and targets are highlighted in yellow and are defined as parameters to the run method. Since this mapping uses a database connection, the corresponding parameter has a special structure. Namely, the connection consists of the connection string (in this case,

jdbc:odbc:;DRIVER=Microsoft Access Driver (*.mdb);DBQ=CustomersAndArticles.mdb), as well as two empty arguments intended for the Username and Password (in clear text) for those databases where this data is necessary.

Note that the file paths in the code below have been changed from absolute to relative.

```
com.altova.io.Input ShortPO2Source =
com.altova.io.StreamInput.createInput("ShortPO.xml");
com.altova.io.Output CompletePO2Target = new
com.altova.io.FileOutput("CompletePO.xml");
```

To define custom mapping source or target files:

• Locate the parameters passed to the run method and edit them as required. In the sample above, com.altova.db.Dbs.newConnection and ShortPO2Source is the mapping input and CompletePO2Target is the mapping output.

To add extra error handling code:

- Edit the code below the catch (Exception e) code (in case of a Java console application)
- Edit the code below the catch (Exception ex) code (in case of a Java dialog application)

For instructions on how to change the data type of parameters supplied as mapping input/output, see <u>Changing Input and Output Programmatically</u>.

14.5.2 C# example

This example uses the Visual Studio 2010 IDE. To begin, generate C# code from the **DB_CompletePO** sample mapping available in the <マイドキュメナ>¥Altova¥MapForce2021¥MapForceExamples¥directory, and then open the solution in Visual Studio.

By default, the name of the solution file is **Mapping.sln**, and it is located in the **Mapping** subdirectory relative to the directory where you saved the generated code. If you changed the application name from the mapping settings, then the name of the .sln file is changed accordingly. For example, if you changed the application name to **MyApplication**, then the solution file is called **MyApplication.sln**, and it is located in the **MyApplication** subdirectory.



Sample C# solution generated with MapForce

Open the MappingConsole.cs file, and locate the main(String[] args) method. The following code sample illustrates an extract from the main method. The mapping sources and targets are defined as parameters to the **Run** method. Since this mapping reads data from a database, there is also an input parameter which is a database connection string. If necessary, you can modify the connection string of the database.

Note that the file paths in the code below have been changed from absolute to relative.

To define custom mapping source or target files:

• Locate the parameters passed to the Run method and edit them as required. In the sample above, the mapping input is a connection string to the CustomersAndArticles.mdb and <code>ShortPO2Source</code>. The mapping output is <code>CompletePO2Target</code>.

To add extra error handling code:

• Edit the code below the catch (Exception e) code

For instructions on how to change the data type of parameters supplied as mapping input/output, see <u>Changing Input and Output Programmatically</u>.

14.5.3 C++ example

This example uses the Visual Studio 2010 IDE. To begin, generate C++ code from the **DB_CompletePO** sample mapping available in the <マイドキュメナト>¥Altova¥MapForce2021¥MapForceExamples¥directory, and then open the solution in Visual Studio.

By default, the name of the solution file is **Mapping.sln**, and it is located in the **Mapping** subdirectory relative to the directory where you saved the generated code. If you changed the application name from the mapping settings, then the name of the .sln file is changed accordingly. For example, if you changed the application name to **MyApplication**, then the solution file is called **MyApplication.sln**, and it is located in the **MyApplication** subdirectory.



Sample C++ solution generated with MapForce

Open the **Mapping.cpp** file, and locate the _tmain method. The following code sample illustrates an extract from this method. The mapping sources and targets are defined as parameters to the **Run** method. Since this mapping reads data from a database, there is also an input parameter which is a database connection string. If necessary, you can modify the connection string of the database.

Note that the file paths in the code below have been changed from absolute to relative.

```
MappingMapToCompletePO MappingMapToCompletePOObject;
MappingMapToCompletePOObject.Run(
```

```
"),

_T("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=CustomersAndArticles.mdb;

_T("ShortPO.xml"),

_T("CompletePO.xml"));
```

To define custom mapping source or target files:

Locate the parameters passed to the Run method and edit them as required. In the code sample above, the mapping input is a connection string to the CustomersAndArticles.mdb database and <u>T("ShortPO.xml"</u>). The mapping output is <u>T("CompletePO.xml"</u>).

To add extra error handling code:

• Edit the code below the catch (CAltovaException& e) code.

14.5.4 Changing Input and Output Programmatically

After generating Java or C# code with MapForce, you can optionally change the data type of the mapping input or output by editing the generated code. More specifically, you can use as mapping parameters objects of types other than those generated by default. For example, instead of having the mapping read the input from a file on the disk, you can provide a string or a stream object as input. Note that this feature is specific to code generated in C# or Java only.

The object types supported as input or output are listed in the first column of the table below. Each subsequent column specifies data formats where that specific type is supported. For a more precise definition of each type, see the "Type definitions" section below.

	XML	Microsoft Excel*	EDI (includes X12, HL7)*	FlexText*	CSV/Text
Files	Yes	Yes	Yes	Yes	Yes
Streams	Yes	Yes	Yes	Yes	Yes
Strings	Yes	Ι	Yes	Yes	Yes
Reader/Writer	Yes	-	Yes	Yes	Yes
DOM documents	Yes	-	-	_	-

* Formats supported only in MapForce Enterprise Edition

To change the data type of the mapping input or output:

- 1. Generate C# or Java code from a mapping.
- 2. In the generated code, find the call to the **run** method (in Java) or **Run** method (in C#), as follows:
 - a. If using C#, open the MappingConsole.cs file.

- b. If using Java, open the **MappingConsole.java** (the console program) or the **MappingFrame.java** file (the GUI program).
- Note: The name of the file may be different if you have changed the application name in the <u>mapping</u> <u>settings</u>. For example, if you changed it to "MyApp", then name of the generated file becomes **MyAppConsole.js** and **MyAppConsole.java**, and **MyAppFrame.java**, respectively.
 - 3. Create an instance of the required type (see the "Type definitions" section).
 - 4. Supply the declared objects as parameters to the **run** method (in Java) or **Run** method (in C#), as shown in the examples below.

The **run** method is the most important method of generated mapping classes. It has one parameter for each *static* source or input component in the mapping, and a final parameter for the output component. If your mapping contains components that process multiple files <u>dynamically</u>, the respective parameters do not appear in generated code, because in this case the file names are processed dynamically inside the mapping.

Type definitions

In C#, the types that you can provide as parameters to the Run method are classes defined in the Altova.IO namespace. The base classes are Altova.IO.Input and Altova.IO.Output, respectively.

Files	Altova.IO.FileInput(string filename) Altova.IO.FileOutput(string filename)
Streams	Altova.IO.StreamInput(System.IO.Stream stream) Altova.IO.StreamOutput(System.IO.Stream stream)
Strings	Altova.IO.StringInput(string content) Altova.IO.StringOutput(System.Text.StringBuilder sbuilder)
Reader/Writer	Altova.IO.ReaderInput(System.IO.TextReader reader) Altova.IO.WriterOutput(System.IO.TextWriter writer)
DOM documents	Altova.IO.DocumentInput(System.Xml.XmlDocument document) Altova.IO.DocumentOutput(System.Xml.XmlDocument document)

C# types

In Java, the types that you can provide as parameters to the **run** method are classes defined in the com.altova.io **package**. The base classes are com.altova.io.Input and com.altova.io.Output, respectively.

Java types

Files	com.altova.io.FileInput(String filename) com.altova.io.FileOutput(String filename)
Streams	com.altova.io.StreamInput(java.io.InputStream stream) com.altova.io.StreamOutput(String filename)
Strings	<pre>com.altova.io.StringInput(String content) com.altova.io.StringOutput()</pre>
Reader/Writer	com.altova.io.ReaderInput(java.io.Reader reader) com.altova.io.WriterOutput(java.io.Writer writer)
DOM documents	<pre>com.altova.io.DocumentInput(org.w3c.dom.Document document)</pre>

Example

To illustrate changing the input and output programmatically, we will use the **ConvertProducts.mfd** mapping as a model. After installing MapForce and running it at least once, you can find this mapping in the following directory: **C:\Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\Tutorials**.



ConvertProducts.mfd

As illustrated above, the mapping converts data from a source XML document to another XML document. Our goals are as follows:

- 1. Generate Java and C# program code from this mapping.
- 2. Change the data type of the source component to a string type.
- 3. Change the data type of the target component to a string writer type.

To generate the program code, open the **ConvertProducts.mfd** mapping and run the **File | Generate code in** | **C#** (or **Java**) command. For the scope of this example, we will assume that the mapping settings of **ConvertProducts.mfd** are the default ones.

😢 Mapping Settings		×
Code Generation		_
Application name:	Mapping	
<u>J</u> ava base package name:	com.mapforce	
Make paths absolute in	generated code	
Reference libraries with	paths relative to generated <u>X</u> SLT / XQuery files	
Ensure Windows path co	onvention for file path output for files from a local file system	

This example uses the following target directories for the generated code (feel free to change the path if necessary):

- C:\codegen\cs\ConvertProducts, for C#
- C:\codegen\java\ConvertProducts, for Java

Having generated the program code, open the **MappingConsole.cs** (in C#) or **MappingConsole.java** (in Java) and find the following lines:

C#

```
Altova.IO.Input Products2Source = Altova.IO.StreamInput.createInput("Products.xml");
Altova.IO.Output ProductValuePairs2Target = new
Altova.IO.FileOutput("ProductValuePairs.xml");
```

Java

```
com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
com.altova.io.Output ProductValuePairs2Target = new
com.altova.io.FileOutput("ProductValuePairs.xml");
```

Comment out the lines above and change the code as follows:

C#

```
//Altova.IO.Input Products2Source = Altova.IO.StreamInput.createInput("Products.xml");
//Altova.IO.Output ProductValuePairs2Target = new
Altova.IO.FileOutput("ProductValuePairs.xml");
Altova.IO.Input Products2Source = new Altova.IO.StringInput("<?xml version=\"1.0\"
encoding=\"UTF-8\"?>\r\n" +
                                        "<products
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"products.xsd\">\r\n" +
                                        ...
                                          <product>\r\n" +
                                        <id>100</id>\r\n" +
                                        ......
                                              <color>blue</color>\r\n" +
                                        .....
                                              <size>XXL</size>\r\n" +
                                        11
                                           </product>\r\n" +
                                        "</products>\r\n");
System.IO.StringWriter writer = new System.IO.StringWriter(new
System.Text.StringBuilder());
Altova.IO.Output ProductValuePairs2Target = new Altova.IO.WriterOutput(writer);
try
{
    MappingMapToProductValuePairsObject.Run(Products2Source, ProductValuePairs2Target);
    // Print out the writer object
    Console.Write(writer.ToString());
finally
{
    Products2Source.Close();
    ProductValuePairs2Target.Close();
}
```

Java

```
//com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
//com.altova.io.Output ProductValuePairs2Target = new
com.altova.io.FileOutput("ProductValuePairs.xml");
com.altova.io.Input Products2Source = new com.altova.io.StringInput("<?xml
version=\"1.0\" encoding=\"UTF-8\"?>\r\n" +
               "<products xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xsi:noNamespaceSchemaLocation=\"products.xsd\">\r\n" +
              " <product>\r\n" +
               <id>100</id>\r\n" +
               <color>blue</color>\r\n" +
               .....
                    <size>XXL</size>\r\n" +
               </product>\r\n" +
               "</products>\r\n");
java.io.StringWriter writer = new java.io.StringWriter();
com.altova.io.Output ProductValuePairs2Target = new com.altova.io.WriterOutput(writer);
try {
  MappingMapToProductValuePairsObject.run(Products2Source, ProductValuePairs2Target);
   // Print out the writer object
  System.out.print(writer.toString());
   } finally {
      (Products2Source).close();
      ProductValuePairs2Target.close();
}
```

In the C# and Java code listings above, the following happens:

- The two original lines that provide the input and output to the **run** method were commented out. Consequently, the mapping application no longer reads data from **Products.xml**. In fact, we did not even need to copy this file to the program's working directory.
- The type Products2Source has been declared as a StringInput that provides the content of the input XML file to be processed.
- The type ProductValuePairs2Target has been declared as a WriterOutput type that takes a string writer as argument.
- After the mapping completes running, the contents of the string writer is printed out to the console.

Usage guidelines for streams and Reader/Writer objects

When using binary streams or Reader/Writer objects as input or output to the mapping, note the following:

- Binary stream objects and Reader/Writer objects are expected to be opened and ready to use before calling the **run** method.
- By default, the **run** method closes the stream when finished. To prevent this behavior, insert (or uncomment) the following line before calling the **run** method:

C#

MappingMapToSomething.CloseObjectsAfterRun = false;

Java

MappingMapToSomething.setCloseObjectsAfterRun(false);

Note: Make sure to change MappingMapToSomething to the name of the mapping object as applicable to your generated code.

Usage guidelines for strings

In Java, the constructor of *stringOutput* does not take an argument. The string content produced by the mapping can be accessed by calling the *getstring()* method, for example:

Java

```
com.altova.io.Input Products2Source =
com.altova.io.StreamInput.createInput("Products.xml");
com.altova.io.StringOutput ProductValuePairs2Target = new com.altova.io.StringOutput();
try {
    // Run the mapping
    MappingMapToProductValuePairsObject.run(Products2Source, ProductValuePairs2Target);
    // Get the string object
    String str = ProductValuePairs2Target.getString().toString();
}
```

In C#, the constructor of StringOutput takes a parameter of type StringBuilder which you need to declare beforehand. If the StringBuilder object already contains data, the mapping output will be appended to it.

C#

```
Altova.IO.Input Products2Source = Altova.IO.StreamInput.createInput("Products.xml");
System.Text.StringBuilder sb = new System.Text.StringBuilder();
Altova.IO.Output ProductValuePairs2Target = new Altova.IO.StringOutput(sb);
try
{
    // Run the mapping
    MappingMapToProductValuePairsObject.Run(Products2Source, ProductValuePairs2Target);
    // Get the string output
    String str = sb.ToString();
}
```

To run these code listings, you can use the same generated project as in the previous example. Make sure, however, to copy the file **Products.xml** from **C**:

\Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\Tutorials\ to your program's working directory, since the mapping code reads data from this file.

Usage guidelines for DOM documents

When using DOM documents as mapping input or output, note the following:

- The document instance supplied as parameter to the DocumentOutput constructor must be empty.
- After calling <u>run</u>, the DOM Document generated by the constructor of <u>DocumentOutput</u> already contains the mapping output, and you can manipulate the document as necessary.

14.6 Generating Code from XML Schemas or DTDs

When you generate code from a mapping, MapForce generates a complete application that executes all steps of the mapping automatically. Optionally, you can generate libraries for all the XML schemas used in the mapping. These allow your code to easily create or read XML instances that are used or created by the mapping code.

To generate libraries for all the XML schemas used in the mapping, select the **Generate Wrapper Classes** check box from the <u>code generator options</u>. Next time when you generate code, MapForce will create not only the mapping application, but also wrapper classes for all schemas used in the mapping, as follows:

C++ or C#	Java	Purpose
Altova	com.altova	Base library containing common runtime support, identical for every schema.
AltovaXML	com.altova.xml	Base library containing runtime support for XML, identical for every schema.
[YourSchema]	com.YourSchema	A library containing declarations generated from the input schema, named as the schema file or DTD. This library is a DOM (W3C Document Object Model) wrapper that allows you to read, modify and create XML documents easily and safely. All data is held inside the DOM, and there are methods for extracting data from the DOM, and to update and create data into the DOM. The generated C++ code supports either Microsoft MSXML or Apache Xerces 3. The syntax for using the generated code is generally similar for both DOM implementations, except for a few slight differences (for example, Xerces supports more overloaded functions). The generated C# code uses the .NET standard System.XML library as the underlying DOM implementation. The generated Java code uses JAXP (Java API for XML Processing) as the underlying DOM interface.

While prototyping an application from a frequently changing XML schema, you may need to frequently generate code to the same directory, so that the schema changes are immediately reflected in the code. Note that the generated test application and the Altova libraries are overwritten every time when you generate code into the same target directory. Therefore, do not add code to the generated test application. Instead, integrate the Altova libraries into your project (see Integrating Schema Wrapper Libraries).

In addition to the base libraries listed above, some supporting libraries are also generated. The supporting libraries are used by the Altova base libraries and are not meant for custom integrations, since they are subject to change.

Name generation and namespaces

MapForce generates classes corresponding to all declared elements or complex types which redefine any complex type in your XML Schema, preserving the class derivation as defined by extensions of complex types in your XML Schema. In the case of complex schemas which import schema components from multiple namespaces, MapForce preserves this information by generating the appropriate C# or C++ namespaces or Java packages.

Generally, the code generator tries to preserve the names for generated namespaces, classes and members from the original XML Schema. Characters that are not valid in identifiers in the target language are replaced by a "_". Names that would collide with other names or reserved words are made unique by appending a number. Name generation can be influenced by changing the default settings in the <u>SPL</u> template.

The namespaces from the XML Schema are converted to packages in Java or namespaces in C# or C++ code, using the namespace prefix from the schema as code namespace. The complete library is enclosed in a package or namespace derived from the schema file name, so you can use multiple generated libraries in one program without name conflicts.

Data Types

XML Schema has a more elaborate data type model than Java, C# or C++. Code Generator converts the builtin XML Schema types to language-specific primitive types, or to classes delivered with the Altova library. Complex types and derived types defined in the schema are converted to classes in the generated library. Enumeration facets from simple types are converted to symbolic constants.

The mapping of simple types can be configured in the SPL template, see <u>SPL Reference</u>.

If your XML instance files use schema types related to time and duration, these are converted to Altova native classes in the generated code. For information about the Altova library classes, see:

- <u>Reference to Generated Classes (C++)</u>
- <u>Reference to Generated Classes (C#)</u>
- <u>Reference to Generated Classes (Java)</u>

For information about type conversion and other details applicable to each language, see:

- About Schema Wrapper Libraries (C++)
- About Schema Wrapper Libraries (C#)
- About Schema Wrapper Libraries (Java)

Memory management

A DOM tree is comprised of nodes, which are always owned by a specific DOM document - even if the node is not currently part of the document's content. All generated classes are references to the DOM nodes they represent, not values. This means that assigning an instance of a generated class does not copy the value, it only creates an additional reference to the same data.

XML Schema support

The following XML Schema constructs are translated into code:

a) XML namespaces

- b) Simple types:
 - Built-in XML schema types
 - Simple types derived by extension
 - Simple types derived by restriction
 - Facets
 - Enumerations
 - Patterns

c) Complex types:

- Built-in anyType node
- User-defined complex types
- Derived by extension: Mapped to derived classes
- Derived by restriction
- Complex content
- Simple content
- Mixed content

The following advanced XML Schema features are not supported (or not fully supported) in generated wrapper classes:

- Wildcards: xs:any and xs:anyAttribute
- Content models (sequence, choice, all). Top-level compositor is available in <u>SPL</u>, but is not enforced by generated classes.
- Default and fixed values for attributes. These are available in <u>SPL</u>, but are not set or enforced by generated classes.
- The attributes xsi:type, abstract types. When you need to write the xsi:type attribute, use the SetXsiType() method of the generated classes.
- Union types: not all combinations are supported.
- Substitution groups are partially supported (resolved like "choice").
- Attribute nillable="true" and xsi:nil
- Uniqueness constraints
- Identity constraints (key and keyref)

14.6.1 About Schema Wrapper Libraries (C++)

Character Types

The generated C++ code can be compiled with or without Unicode support. Depending on this setting, the types string_type and tstring will both be defined as std::string or std::wstring, consisting of narrow or wide characters. To use Unicode characters in your XML file that are not representable with the current 8-bit character set, Unicode support must be enabled. Pay special attention to the $_T()$ macros. This macro ensures that string constants are stored correctly, whether you're compiling for Unicode or non-Unicode programs.

Data Types

The default mapping of XML Schema types to C++ data types is:

XML Schema	C++	Remarks
xs:string	string_type	string_type is defined as std::string or std:wstring
xs:boolean	bool	
xs:decimal	double	C++ does not have a decimal type, so double is used.
xs:float, xs:double	double	
xs:integer	int64	xs:integer has unlimited range, mapped toint64 for efficiency reasons.
xs:nonNegativeInteger	unsignedint64	see above
xs:int	int	
xs:unsignedInt	unsigned int	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	altova::DateTime	
xs:duration	altova::Duration	
xs:hexBinary and xs:base64Binary	<pre>std::vector<unsigned char=""></unsigned></pre>	Encoding and decoding of binary data is done automatically.
xs:anySimpleType	string_type	

All XML Schema types not contained in this list are derived types, and mapped to the same C++ type as their respective base type.

Generated Classes

For each type in the schema, a class is generated that contains a member for each attribute and element of the type. The members are named the same as the attributes or elements in the original schema (in case of possible collisions, a number is appended). For simple types, assignment and conversion operators are generated. For simple types with enumeration facets, the methods <code>GetEnumerationValue()</code> and <code>SetEnumerationValue(int)</code> can be used together with generated constants for each enumeration value. In addition, the method <code>StaticInfo()</code> allows accessing schema information as one of the following types:

altova::meta::SimpleType
altova::meta::ComplexType

Classes generated from complex types include the method <code>SetXsiType()</code>, which enables you to set the <code>xsi:type</code> attribute of the type. This method is useful when you want to create XML instance elements of a derived type.

In addition to the classes for the types declared in the XML Schema, a document class (identified with "CDoc" below) is generated. It contains all possible root elements as members, and various other methods. For more information about the class, see [YourSchema]::[CDoc].

Note: The actual class name depends on the name of the .xsd schema.

For each member attribute or element of a schema type, a new class is generated. For more information about such classes, see:

[YourSchema]::MemberAttribute [YourSchema]::MemberElement

Note: The actual class names depend on the name of the schema attribute or element.

See also Example: Using the Schema Wrapper Libraries.

Error Handling

Errors are reported by exceptions. The following exception classes are defined in the namespace altova:

Class	Base Class	Description
Error	std::logic_error	Internal program logic error (independent of input data).
Exception	<pre>std::runtime_error</pre>	Base class for runtime errors.
InvalidArgumentsException	Exception	A method was called with invalid argument values.
ConversionException	Exception	Exception thrown when a type conversion fails.
StringParseException	ConversionException	A value in the lexical space cannot be converted to value space.
ValueNotRepresentableExcep tion	ConversionException	A value in the value space cannot be converted to lexical space.
OutOfRangeException	ConversionException	A source value cannot be represented in target domain.
InvalidOperationException	Exception	An operation was attempted that is not valid in the given context.
DataSourceUnavailableExcep tion	Exception	A problem occurred while loading an XML instance.
DataTargetUnavailableExcep tion	Exception	A problem occurred while saving an XML instance.

All exception classes contain a message text and a pointer to a possible inner exception.

Method	Purpose
<pre>string_type message()</pre>	Returns a textual description of the exception.
<pre>std::exception inner()</pre>	Returns the exception that caused this exception, if available, or NULL.

Accessing schema information

The generated library allows accessing static schema information via the following classes. All methods are declared as const. The methods that return one of the metadata classes return a NULL object if the respective property does not exist.

altova::meta::Attribute
altova::meta::ComplexType
altova::meta::Element
altova::meta::SimpleType

14.6.2 About Schema Wrapper Libraries (C#)

The default mapping of XML Schema types to C# data types is as follows.

XML Schema	C#	Remarks
xs:string	string	
xs:boolean	bool	
xs:decimal	decimal	xs:decimal has unlimited range and precision, mapped to decimal for efficiency reasons.
<pre>xs:float, xs:double</pre>	double	
xs:long	long	
xs:unsignedLong	ulong	
xs:int	int	
xs:unsignedInt	uint	
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	<u>Altova.Types.DateTime</u>	
xs:duration	Altova.Types.Duration	
xs:hexBinary and xs:base64Binary	byte[]	Encoding and decoding of binary data is done automatically.
xs:anySimpleType	string	

All XML Schema types not contained in this list are derived types, and mapped to the same C# type as their respective base type.

Generated Classes

For each type in the schema, a class is generated that contains a member for each attribute and element of the type. The members are named the same as the attributes or elements in the original schema (in case of possible collisions, a number is appended). For simple types, assignment and conversion operators are

generated. For simple types with enumeration facets, the methods GetEnumerationValue() and SetEnumerationValue(int) can be used together with generated constants for each enumeration value. In addition, the method StaticInfo() allows accessing schema information as one of the following types:

Altova.Xml.Meta.SimpleType Altova.Xml.Meta.ComplexType

Classes generated from complex types include the method <code>SetXsiType()</code>, which enables you to set the <code>xsi:type</code> attribute of the type. This method is useful when you want to create XML instance elements of a derived type.

In addition to the classes for the types declared in the XML Schema, a document class (identified with "Doc" below) is generated. It contains all possible root elements as members, and various other methods. For more information about the class, see <u>[YourSchema].[Doc]</u>.

Note: The actual class name depends on the name of the .xsd schema.

For each member attribute or element of a schema type, a new class is generated. For more information about such classes, see:

[YourSchemaType].MemberAttribute [YourSchemaType].MemberElement

Note: The actual class names depend on the name of the schema attribute or element.

Error Handling

Errors are reported by exceptions. The following exception classes are defined in the namespace Altova:

Class	Base Class	Description
ConversionException	Exception	Exception thrown when a type conversion fails
StringParseException	ConversionException	A value in the lexical space cannot be converted to value space.
DataSourceUnavailableException	System.Exception	A problem occurred while loading an XML instance.
DataTargetUnavailableException	System.Exception	A problem occurred while saving an XML instance.

In addition, the following .NET exceptions are commonly used:

Class	Description
System.Exception	Base class for runtime errors
System.ArgumentException	A method was called with invalid argument values, or a type conversion failed.
System.FormatException	A value in the lexical space cannot be converted to value space.

Class	Description
System.InvalidCastException	A value cannot be converted to another type.
System.OverflowException	A source value cannot be represented in target domain.

Accessing schema information

The generated library allows accessing static schema information via the following classes:

Altova.Xml.Meta.Attribute Altova.Xml.Meta.ComplexType Altova.Xml.Meta.Element Altova.Xml.Meta.SimpleType

The properties that return one of the metadata classes return null if the respective property does not exist.

14.6.3 About Schema Wrapper Libraries (Java)

The default mapping of XML Schema types to Java data types is as follows:

XML Schema	Java	Remarks
xs:string	String	
xs:boolean	boolean	
xs:decimal	java.math.BigDecimal	
<pre>xs:float, xs:double</pre>	double	
xs:integer	java.math.BigInteger	
xs:long	long	
xs:unsignedLong	java.math.BigInteger	Java does not have unsigned types.
xs:int	int	
xs:unsignedInt	long	Java does not have unsigned types.
xs:dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth	<u>com.altova.types.DateTi</u> <u>me</u>	
xs:duration		
	<u>com.altova.types.Durati</u> <u>on</u>	
xs:hexBinary and xs:base64Binary	<u>com.altova.types.Durati</u> <u>on</u> byte[]	Encoding and decoding of binary data is done automatically.

All XML Schema types not contained in this list are derived types, and mapped to the same Java type as their respective base type.

Generated Classes

For each type in the schema, a class is generated that contains a member for each attribute and element of the type. The members are named the same as the attributes or elements in the original schema (in case of possible collisions, a number is appended). For simple types, assignment and conversion operators are generated. For simple types with enumeration facets, the methods GetEnumerationValue() and SetEnumerationValue(int) can be used together with generated constants for each enumeration value. In addition, the method <code>StaticInfo()</code> allows accessing schema information as one of the following types:

com.altova.xml.meta.SimpleType
com.altova.xml.meta.ComplexType

Classes generated from complex types include the method <code>SetXsiType()</code>, which enables you to set the <code>xsi:type</code> attribute of the type. This method is useful when you want to create XML instance elements of a derived type.

In addition to the classes for the types declared in the XML Schema, a document class (identified with "Doc" below) is generated. It contains all possible root elements as members, and various other methods. For more information about the class, see <u>com.[YourSchema].[Doc]</u>.

Note: The actual class name depends on the name of the .xsd schema.

For each member attribute or element of a schema type, a new class is generated. For more information about such classes, see:

com.[YourSchema].[YourSchemaType].MemberAttribute
com.[YourSchema].[YourSchemaType].MemberElement

Note: The actual class names depend on the name of the schema attribute or element.

Error Handling

Errors are reported by exceptions. The following exception classes are defined in the namespace com.altova:

Class	Base Class	Description
SourceInstanceUnvailableException	Exception	A problem occurred while loading an XML instance.
TargetInstanceUnavailableException	Exception	A problem occurred while saving an XML instance.

In addition, the following Java exceptions are commonly used:

Class	Description
java.lang.Error	Internal program logic error (independent of input data)
java.lang.Exception	Base class for runtime errors

Class	Description
java.lang.IllegalArgumentsException	A method was called with invalid argument values, or a type conversion failed.
java.lang.ArithmeticException	Exception thrown when a numeric type conversion fails.

Accessing schema information

The generated library allows accessing static schema information via the following classes:

com.altova.xml.meta.Attribute com.altova.xml.meta.ComplexType com.altova.xml.meta.Element com.altova.xml.meta.SimpleType

The properties that return one of the metadata classes return null if the respective property does not exist.

14.6.4 Integrating Schema Wrapper Libraries

To use the Altova libraries in your custom project, refer to the libraries from your project (or include them into your project), as shown below for each language.

C#

To integrate the Altova libraries into an existing C# project:

- 1. After MapForce generates code from a schema (for example, **YourSchema.xsd**), build the generated **YourSchema.sin** solution in Visual Studio. This solution is in a project folder with the same name as the schema.
- 2. Right-click your existing project in Visual Studio, and select Add Reference.
- 3. On the Browse tab, browse for the following libraries: Altova.dll, AltovaXML.dll, and YourSchema.dll located in the output directory of the generated projects (for example, bin\Debug).

🕫 Add Reference
.NET COM Projects Browse Recent
Look in: 🌗 Debug 🗸 🌀 🎓 📰 🔻
Name
S Altova.dll AltovaXML.dll Library.dll
۰ III ۲
File name:
Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)
OK Cancel

C++

The easiest way to integrate the libraries into an existing C++ project is to add the generated project files to your solution. For example, let's assume that you generated code from a schema called **Library.xsd** and selected **c:\codegen\cpp\library** as target directory. The generated libraries in this case are available at:

- c:\codegen\cpp\library\Altova.vcxproj
- c:\codegen\cpp\library\AltovaXML\AltovaXML.vcxproj
- c:\codegen\cpp\library\Library.vcxproj

First, open the generated c:\codegen\cpp\library\Library.sln solution and build it in Visual Studio.

Next, open your existing Visual Studio solution (in Visual Studio 2010, in this example), right-click it, select **Add | Existing Project**, and add the project files listed above, one by one. Be patient while Visual Studio parses the files. Next, right-click your project and select **Properties**. In the Property Pages dialog box, select **Common Properties | Framework and References**, and then click **Add New Reference**. Next, select and add each of the following projects: *Altova, AltovaXML*, and *Library*.

Add Reference	?
Projects	
Project Name	Project Directory
Altova	C:\codegen\cpp\library\Altova\
AltovaXML	C:\codegen\cpp\library\AltovaXML\
Library	C:\codegen\cpp\library\Library\
•	4 III
	OK Cancel

See also the MSDN documentation for using functionality from a custom library, as applicable to your version of Visual Studio, for example:

- If you chose to generate static libraries, see <u>https://msdn.microsoft.com/en-us/library/ms235627(v=vs.100).aspx</u>
- If you chose to generate dynamic libraries, see https://msdn.microsoft.com/en-us/library/ms235636(v=vs.100).aspx

The option to generate static or dynamic libraries is available in code generation options (see <u>Code generator</u> <u>options</u>).

Java

One of the ways to integrate the Altova packages into your Java project is to copy the **com** directory of the generated code to the directory which stores the source packages of your Java project (for example, **C**: **Workspace\MyJavaProject\src**). For example, let's assume that you generated code in **c**: **codegen\java\library**. The generated Altova classes in this case are available at **c**: **codegen\java\library\com**.

After copying the libraries, refresh the project. To refresh the project in Eclipse, select it in the Package Explorer, and press **F5**. To refresh the project in NetBeans IDE 8.0, select the menu command **Source | Scan for External Changes**.

Once you perform the copy operation, the Altova packages are available in the Package Explorer (in case of Eclipse), or under "Source Packages" in the Projects pane (in case of NetBeans IDE).



Altova packages in Eclipse 4.4



Altova packages in NetBeans IDE 8.0.2

14.6.5 Example: Book Library

This example illustrates how to use the generated schema wrapper libraries in order to write or read programmatically XML documents conformant to the schema. Before using the sample code, take some time to understand the structure of the schema below.

The schema used in this example describes a library of books. The complete definition of the schema is shown below. Save this code listing as **Doc.xsd** if you want to get the same results as this example. You will need this schema to generate the code libraries used in this example.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.nanonull.com/LibrarySample"</pre>
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.nanonull.com/LibrarySample" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="Library">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Book" type="BookType" minOccurs="0"</pre>
maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="LastUpdated" type="xs:dateTime"/>
      </ms:complexType>
   </r></r></r>
   <xs:complexType name="BookType">
      <xs:sequence>
         <xs:element name="Title" type="xs:string"/>
         <xs:element name="Author" type="xs:string" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ID" type="xs:integer" use="required"/>
      <xs:attribute name="Format" type="BookFormatType" use="required"/>
   </xs:complexType>
   <xs:complexType name="DictionaryType">
      <re><xs:complexContent>
         <xs:extension base="BookType">
            <xs:sequence>
               <xs:element name="FromLang" type="xs:string"/>
               <xs:element name="ToLang" type="xs:string"/>
            </xs:sequence>
         </xs:extension>
      </ms:complexContent>
   </r></r></r>
   <xs:simpleType name="BookFormatType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Hardcover"/>
         <xs:enumeration value="Paperback"/>
         <xs:enumeration value="Audiobook"/>
         <xs:enumeration value="E-book"/>
      </ms:restriction>
   </xs:simpleType>
</xs:schema>
```

Library is a root element of a complexType which can be graphically represented as follows in the schema view of XMLSpy:



As shown above, the library has a **LastUpdated** attribute (defined as xs:dateTime), and stores a sequence of books. Each book is an xs:complexType and has two attributes: an **ID** (defined as xs:integer), and a **Format**. The format of any book can be hardcover, paperback, audiobook, or e-book. In the schema, **Format** is defined as xs:simpleType which uses an enumeration of the above-mentioned values.

Each book also has a **Title** element (defined as xs:string), as well as one or several **Author** elements (defined as xs:string).

The library may also contain books that are dictionaries. Dictionaries have the type DictionaryType, which is derived by extension from the BookType. In other words, a dictionary inherits all attributes and elements of a Book, plus two additional elements: **FromLang** and **ToLang**, as illustrated below.



The FromLang and ToLang elements store the source and destination language of the dictionary.

1112 Code Generator

An XML instance file valid according to the schema above could therefore look as shown in the listing below (provided that it is in the same directory as the schema file):

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd"
LastUpdated="2016-02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

The next topics illustrate how to read from such a file programmatically, or write to such a file programmatically. To begin, generate the schema wrapper code from the schema above, using the steps described in <u>Generating Code from XML Schemas or DTD</u>.

14.6.5.1 Reading and Writing XML Documents (C++)

After you generate code from the <u>example schema</u>, a test C++ application is created, along with several supporting Altova libraries.

About the generated C++ libraries

The central class of the generated code is the CDOC class, which represents the XML document. Such a class is generated for every schema and its name depends on the schema file name (**Doc.xsd**, in this example). As shown in the diagram, this class provides methods for loading documents from files, binary streams, or strings (or saving documents to files, streams, strings). For a description of all members exposed by this class, see the class reference ([YourSchema]::[CDoc]).

CDoc
Library:MemberElement <membertype->CLibraryType,MemberIndex->_altova_mi_altova_CDoc_altova_Library></membertype->
«constructor» CDoc(in unnamed1:int)
«constructor» CDoc(in init:«const» CDoc&)
operator=(in other:«const» CDoc&):void
<u>StaticInfo():ComplexType</u>
SetXsiType():void
LoadFromFile(in fileName:«const» string type&):CDoc
LoadFromString(in xml:«const» string type&):CDoc
LoadFromBinary(in data:«const» vector< Ty-> unsigned_char>&):CDoc
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool):void
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmIDecl:bool):void
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in encoding:«const» string_type&():void
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmIDecl:bool, in encoding:«const» string_type&):void
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in encoding:«const» string_type&, in bBigEndian:bool, in bBOM:bool):void
SaveToFile(in fileName:«const» string_type&, in prettyPrint:bool, in omitXmIDecl:bool, in encoding:«const» string_type&, in bBigEndian:bool, in bBOM:bool):void
SaveToString(in prettyPrint:bool):string_type
SaveToString(in prettyPrint:bool, in omitXmlDecl:bool):string_type
SaveToBinary(in prettyPrint:bool):vector<_Ty-> unsigned_char>
SaveToBinary(in prettyPrint:bool, in encoding:« <mark>const»</mark> string_type&):vector<_Ty-> unsigned_char>
SaveToBinary(in prettyPrint:bool, in encoding:« <mark>const»</mark> string_type&, in bBigEndian:bool, in bBOM:bool):vector<_Ty-> unsigned_char>
<u>CreateDocument():CDoc</u>
DestroyDocument():void
SetDTDLocation(in dtdLocation:«const» string_type&):void
SetSchemaLocation(in schemaLocation:«const» string_type&):void
DeclareAllNamespacesFromSchema(in node:TypeBase&):void
••

The Library field of the CDoc class represents the actual root of the document. Library is an element in the XML file, so in the C++ code it has a template class as type (MemberElement). The template class exposes methods and properties for interacting with the Library element. In general, each attribute and each element of a type in the schema is typed in the generated code with the MemberAttribute and MemberElement template classes, respectively. For more information, see [YourSchema]::MemberAttribute and [YourSchema]::MemberElement class reference.

The class CLibraryType is generated from the LibraryType complex type in the schema. Notice that the CLibraryType class contains two fields: Book and LastUpdated. According to the logic already mentioned above, these correspond to the Book element and LastUpdated attribute in the schema, and enable you to manipulate programmatically (append, remove, etc) elements and attributes in the instance XML document.

CLibraryType
LastUpdated:MemberAttribute <membertype->DateTime,MemberIndex->_altova_mi_altova_CLibraryType_altova_LastUpdated,EnumOffset->0,EnumCount->0> Book:MemberElement<membertype->CBookType,MemberIndex->_altova_mi_altova_CLibraryType_altova_Book></membertype-></membertype->
struct» Book

The DictionaryType is a complex type derived from **BookType** in the schema, so this relationship is also reflected in the generated classes. As illustrated in the diagram, the class CDictionaryType inherits the CBookType class.

СВоокТуре		
X X X	ID:MemberAttribute <membertype->long_long,MemberIndex->_altova_mi_altova_CBookType_altova_ID,EnumOffset->0,EnumCount->0> Format:MemberAttribute<membertype->string_type,MemberIndex->_altova_mi_altova_CBookType_altova_Format,EnumOffset->1,EnumCount->4> Title:MemberElement<membertype->CstringType,MemberIndex->_altova_mi_altova_CBookType_altova_Title> Author:MemberElement<membertype->CstringType,MemberIndex->_altova_mi_altova_CBookType_altova_Title></membertype-></membertype-></membertype-></membertype->	
	<pre>«constructor» CBookType(in unnamed1:int) «constructor» CBookType(in init:«const» CBookType&) operator=(in other:«const» CBookType&):void StaticInfo():ComplexType SetXsiType():void</pre>	
	«struct» Title «struct» Author	
	∆ ∝visibility»	«visibility» value = public
CDictionaryType		
	FromLang:MemberElement <membertype->CstringType,MemberIndex->_altova_mi_altova_CDictionaryType_altova_FromLang> ToLang:MemberElement<membertype->CstringType,MemberIndex->_altova_mi_altova_CDictionaryType_altova_ToLang></membertype-></membertype->	
00000	<pre>«constructor» CDictionaryType(in unnamed1:int) «constructor» CDictionaryType(in init:«const» CDictionaryType&) operator=(in other:«const» CDictionaryType&):void <u>StaticInfo():ComplexType</u> SetXsiType():void</pre>	
	«struct» FromLang «struct» ToLang	

If your XML schema defines simple types as enumerations, the enumerated values become available as enum values in the generated code. In the schema used in this example, a book format can be hardcover, paperback, e-book, and so on. Therefore, in the generated code, these values would be available through an enum that is a member of the CBookFormatType class.

Writing an XML document

1. Open the **LibraryTest.sln** solution in Visual Studio generated from the Library schema mentioned earlier in this example.

While prototyping an application from a frequently changing XML schema, you may need to frequently generate code to the same directory, so that the schema changes are immediately reflected in the code. Note that the generated test application and the Altova libraries are overwritten every time when you generate code into the same target directory. Therefore, do not add code to the generated test application. Instead, integrate the Altova libraries into your project (see <u>Integrating Schema Wrapper Libraries</u>).

2. In Solution Explorer, open the LibraryTest.cpp file, and edit the Example() method as shown below.

```
#include <ctime> // required to get current time
using namespace Doc; // required to work with Altova libraries
void Example()
{
```

```
// Create a new, empty XML document
  CDoc libDoc = CDoc::CreateDocument();
   // Create the root element <Library> and add it to the document
  CLibraryType lib = libDoc.Library.append();
   // Get current time and set the "LastUpdated" attribute using Altova classes
  time t t = time(NULL);
  struct tm * now = localtime( & t );
  altova::DateTime dt = altova::DateTime(now->tm year + 1900, now->tm mon + 1, now-
>tm mday, now->tm hour, now->tm min, now->tm sec);
  lib.LastUpdated = dt;
   // Create a new <Book> and add it to the library
  CBookType book = lib.Book.append();
   // Set the "ID" attribute of the book
  book.ID = 1;
   // Set the "Format" attribute of the <Book> using an enumeration constant
  book.Format.SetEnumerationValue( CBookFormatType::k Paperback );
   // Add the <Title> and <Author> elements, and set values
  book.Title.append() = T("The XML Spy Handbook");
  book.Author.append() = T("Altova");
  // Append a dictionary (book of derived type) and populate its attributes and
elements
  CDictionaryType dictionary = CDictionaryType(lib.Book.append().GetNode());
  dictionary.ID = 2;
  dictionary.Format.SetEnumerationValue( CBookFormatType::k E book);
  dictionary.Title.append() = _T("English-German Dictionary");
  dictionary.Author.append() = T("John Doe");
  dictionary.FromLang.append() = T("English");
  dictionary.ToLang.append() = T("German");
   // Since dictionary a derived type, set the xsi:type attribute of the book element
  dictionary.SetXsiType();
   // Optionally, set the schema location
  libDoc.SetSchemaLocation( T("Library.xsd"));
   // Save the XML document to a file with default encoding (UTF-8),
   // "true" causes the file to be pretty-printed.
   libDoc.SaveToFile( T("GeneratedLibrary.xml"), true);
   // Destroy the document
   libDoc.DestroyDocument();
}
```

 Press F5 to start debugging. If the code was executed successfully, a GeneratedLibrary.xml file is created in the solution output directory.

Reading an XML document

- 1. Open the LibraryTest.sln solution in Visual Studio.
- 2. Save the code below as **Library1.xml** to a directory that can be read by the program code (for example, the same directory as **LibraryTest.sln**).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd"
LastUpdated="2016-02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

3. In Solution Explorer, open the LibraryTest.cpp file, and edit the Example() method as shown below.

```
using namespace Doc;
void Example()
   // Load XML document
   CDoc libDoc = CDoc::LoadFromFile( T("Library1.xml"));
   // Get the first (and only) root element <Library>
   CLibraryType lib = libDoc.Library.first();
   // Check whether an element exists:
   if (!lib.Book.exists())
   {
      tcout << "This library is empty." << std::endl;</pre>
      return;
   }
   // iteration: for each <Book>...
   for (Iterator<CBookType> itBook = lib.Book.all(); itBook; ++itBook)
      // output values of ISBN attribute and (first and only) title element
      tcout << "ID: " << itBook->ID << std::endl;</pre>
      tcout << "Title: " << tstring(itBook->Title.first()) << std::endl;</pre>
      // read and compare an enumeration value
      if (itBook->Format.GetEnumerationValue() == CBookFormatType::k_Paperback)
         tcout << "This is a paperback book." << std::endl;</pre>
      // for each <Author>...
      for (CBookType::Author::iterator itAuthor = itBook->Author.all(); itAuthor; +
+itAuthor)
         tcout << "Author: " << tstring(itAuthor) << std::endl;</pre>
      // alternative: use count and index
      for (unsigned int j = 0; j < itBook->Author.count(); ++j)
         tcout << "Author: " << tstring(itBook->Author[j]) << std::endl;</pre>
   }
   // Destroy the document
```
```
libDoc.DestroyDocument();
}
```

4. Press **F5** to start debugging.

14.6.5.2 Reading and Writing XML Documents (C#)

After you generate code from the <u>example schema</u>, a test C# application is created, along with several supporting Altova libraries.

About the generated C# libraries

The central class of the generated code is the Doc2 class, which represents the XML document. Such a class is generated for every schema and its name depends on the schema file name (**Doc.xsd**, in this example). Note that this class is called Doc2 to avoid a possible conflict with the namespace name. As shown in the diagram, this class provides methods for loading documents from files, binary streams, or strings (or saving documents to files, streams, strings). For a description of this class, see the class reference ([YourSchema]. [Doc]).

	Doc2		
	Library:MemberElement_Library		
	C# Properties		
	«GetAccessor, property» StaticInfo():Complex Type		
	Methods		
	LoadFromFile(in filename:string):Doc2		
	LoadFromString(in xmlstring:string):Doc2		
	LoadFromBinary(in binary:byte[*]):Doc2		
	SaveToFile(in filename:string, in prettyPrint:bool):void		
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool):void		
	SaveToFileWithLineEnd(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in lineend:string):void		
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string):void		
	SaveToFile(in filename:string, in prettyPrint:bool, in encoding:string, in lineend:string):void		
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string, in lineend:string):void		
	SaveToFile(in filename:string, in prettyPrint:bool, in omitXmlDecl:bool, in encoding:string, in bBigEndian:bool, in bBOM:bool, in lineend:string):void		
	SaveToString(in prettyPrint:bool):string		
	SaveToString(in prettyPrint:bool, in omitXmlDecl:bool):string		
	SaveToBinary(in prettyPrint:bool):byte[*]		
	SaveToBinary(in prettyPrint:bool, in encoding:string):byte[*]		
	SaveToBinary(in prettyPrint:bool, in encoding:string, in bBigEndian:bool, in bBOM:bool):byte[*]		
	CreateDocument():Doc2		
	CreateDocument(in encoding:string):Doc2		
	SetDTDLocation(in dtdLocation:string):void		
	SetSchemaLocation(in schemaLocation:string):void		
	<u>DeclareAllNamespacesFromSchema(in node:TypeBase):void</u>		
\diamond	«constructor» Doc2(in init:System.Xml.XmlNode)		
	SetXsiType():void		
	m		

The Library member of the Doc2 class represents the actual root of the document.

1118 Code Generator

According to the code generation rules mentioned in <u>About Schema Wrapper Libraries (C#</u>), member classes are generated for each attribute and for each element of a type. In the generated code, the name of such member classes is prefixed with MemberAttribute_ and MemberElement_, respectively. Examples of such classes are MemberAttribute_ID and MemberElement_Author, generated from the **Author** element and **ID** attribute of a book, respectively (in the diagram below, they are classes nested under BookType). Such classes enable you to manipulate programmatically the corresponding elements and attributes in the instance XML document (for example, append, remove, set value, etc). For more information, see the [YourSchemaType].MemberAttribute and [YourSchemaType].MemberElement class reference.

Since the **DictionaryType** is a complex type derived from **BookType** in the schema, this relationship is also reflected in the generated classes. As illustrated in the diagram below, the class <code>DictionaryType</code> inherits the <code>BookType</code> class.



If your XML schema defines simple types as enumerations, the enumerated values become available as Enum values in the generated code. In the schema used in this example, a book format can be hardcover, paperback, e-book, and so on. Therefore, in the generated code, these values would be available through an Enum that is a member of the BookFormatType class.

Writing an XML document

1. Open the **LibraryTest.sln** solution in Visual Studio generated from the Library schema mentioned earlier in this example.

While prototyping an application from a frequently changing XML schema, you may need to frequently generate code to the same directory, so that the schema changes are immediately reflected in the code. Note that the generated test application and the Altova libraries are overwritten every time when you generate code into the same target directory. Therefore, do not add code to the generated test application. Instead, integrate the Altova libraries into your project (see Integrating Schema Wrapper Libraries).

2. In Solution Explorer, open the LibraryTest.cs file, and edit the Example() method as shown below.

```
protected static void Example()
    // Create a new XML document
   Doc2 doc = Doc2.CreateDocument();
    // Append the root element
   LibraryType root = doc.Library.Append();
    // Create the generation date using Altova DateTime class
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
   // Append the date to the root
   root.LastUpdated.Value = dt;
    // Add a new book
   BookType book = root.Book.Append();
    // Set the value of the ID attribute
   book.ID.Value = 1;
    // Set the format of the book (enumeration)
   book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
   // Set the Title and Author elements
   book.Title.Append().Value = "The XMLSpy Handbook";
   book.Author.Append().Value = "Altova";
    // Append a dictionary (book of derived type) and populate its attributes
and elements
   DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
   dictionary.ID.Value = 2;
   dictionary.Title.Append().Value = "English-German Dictionary";
   dictionary.Format.EnumerationValue = BookFormatType.EnumValues.eE book;
   dictionary.Author.Append().Value = "John Doe";
   dictionary.FromLang.Append().Value = "English";
   dictionary.ToLang.Append().Value = "German";
    // Since it's a derived type, make sure to set the xsi:type attribute of the
book element
   dictionary.SetXsiType();
   // Optionally, set the schema location (adjust the path if
    // your schema is not in the same folder as the generated instance file)
   doc.SetSchemaLocation("Library.xsd");
    // Save the XML document with the "pretty print" option enabled
   doc.SaveToFile("GeneratedLibrary.xml", true);
}
```

3. Press **F5** to start debugging. If the code was executed successfully, a **GeneratedLibrary.xml** file is created in the solution output directory (typically, **bin/Debug**).

Reading an XML document

- 1. Open the LibraryTest.sln solution in Visual Studio.
- 2. Save the code below as **Library.xml** to the output directory of the project (by default, **bin/Debug**). This is the file that will be read by the program code.

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd"
LastUpdated="2016-02-03T17:10:08.4977404">
   <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="nl:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
  </Book>
</Library>
```

3. In Solution Explorer, open the LibraryTest.cs file, and edit the Example() method as shown below.

```
protected static void Example()
    // Load the XML file
   Doc2 doc = Doc2.LoadFromFile("Library.xml");
    // Get the root element
   LibraryType root = doc.Library.First;
    // Read the library generation date
   Altova.Types.DateTime dt = root.LastUpdated.Value;
   string dt as string = dt.ToString(DateTimeFormat.W3 dateTime);
   Console.WriteLine("The library generation date is: " + dt as string);
    // Iteration: for each <Book>...
    foreach (BookType book in root.Book)
     // Output values of ID attribute and (first and only) title element
     Console.WriteLine("Title: " + book.Title.First.Value);
     // Read and compare an enumeration value
     if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
        Console.WriteLine("This is a paperback book.");
     // Iteration: for each <Author>
     foreach (xs.stringType author in book.Author)
        Console.WriteLine("Author: " + author.Value);
      // Determine if this book is of derived type
     if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
      {
```

```
// Find the value of the xsi:type attribute
         string xsiTypeValue =
book.Node.Attributes.GetNamedItem("xsi:type").Value;
         // Get the namespace URI and the lookup prefix of this namespace
         string namespaceUri = book.Node.NamespaceURI;
         string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);
         // if this book has DictionaryType
         if (namespaceUri == "http://www.nanonull.com/LibrarySample" &&
xsiTypeValue.Equals(prefix + ":DictionaryType"))
         {
            // output additional fields
            DictionaryType dictionary = new DictionaryType(book.Node);
           Console.WriteLine("Language from: " +
dictionary.FromLang.First.Value);
           Console.WriteLine("Language to: " + dictionary.ToLang.First.Value);
         }
         else
         {
            throw new Exception("Unexpected book type");
         }
      }
    }
    Console.ReadLine();
}
```

4. Press **F5** to start debugging. If the code was executed successfully, **Library.xml** will be read by the program code, and its contents displayed as console output.

Reading and writing elements and attributes

Values of attributes and elements can be accessed using the value property of the generated member element or attribute class, for example:

```
// Output values of ID attribute and (first and only) title element
Console.WriteLine("ID: " + book.ID.Value);
Console.WriteLine("Title: " + book.Title.First.Value);
```

To get the value of the **Title** element in this particular example, we also used the <code>First()</code> method, since this is the first (and only) **Title** element of a book. For cases when you need to pick a specific element from a list by index, use the <code>At()</code> method.

The class generated for each member element of a type implements the standard System.Collections.IEnumerable interface. This makes it possible to loop through multiple elements of the same type. In this particular example, you can loop through all books of a Library object as follows:

```
// Iteration: for each <Book>...
foreach (BookType book in root.Book)
{
    // your code here...
}
```

To add a new element, use the Append() method. For example, the following code appends the root element to the document:

```
// Append the root element to the library
LibraryType root = doc.Library.Append();
```

You can set the value of an attribute (like ID in this example) as follows:

```
// Set the value of the ID attribute
book.ID.Value = 1;
```

Reading and writing enumeration values

If your XML schema defines simple types as enumerations, the enumerated values become available as Enum values in the generated code. In the schema used in this example, a book format can be hardcover, paperback, e-book, and so on. Therefore, in the generated code, these values would be available through an Enum:

```
EnumValues
Enum
eHardcover
ePaperback
eAudiobook
eE_book
Invalid
```

To assign enumeration values to an object, use code such as the one below:

```
// Set the format of the book (enumeration)
book.Format.EnumerationValue = BookFormatType.EnumValues.eHardcover;
```

You can read such enumeration values from XML instance documents as follows:

```
// Read and compare an enumeration value
if (book.Format.EnumerationValue == BookFormatType.EnumValues.ePaperback)
Console.WriteLine("This is a paperback book.");
```

When an "if" condition is not enough, create a switch to determine each enumeration value and process it as required.

Working with xs:dateTime and xs:duration types

If the schema from which you generated code uses time and duration types such as xs:dateTime, or xs:duration, these are converted to Altova native classes in generated code. Therefore, to write a date or duration value to the XML document, do the following:

- 1. Construct an <u>Altova.Types.DateTime</u> or <u>Altova.Types.Duration</u> object (either from System.DateTime, or by using parts such as hours and minutes, see <u>Altova.Types.DateTime</u> and <u>Altova.Types.Duration</u> for more information).
- 2. Set the object as value of the required element or attribute, for example:

```
// Create the library generation date using Altova DateTime class
Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
// Append the date to the root
root.LastUpdated.Value = dt;
```

To read a date or duration from an XML document, do the following:

- 1. Declare the element value (or attribute) as <u>Altova.Types.DateTime</u> or <u>Altova.Types.Duration</u> object.
- 2. Format the required element or attribute, for example:

```
// Read the library generation date
Altova.Types.DateTime dt = root.LastUpdated.Value;
string dt_as_string = dt.ToString(DateTimeFormat.W3_dateTime);
Console.WriteLine("The library generation date is: " + dt as string);
```

For more information, see <u>Altova.Types.DateTime</u> and <u>Altova.Types.Duration</u> class reference.

Working with derived types

If your XML schema defines derived types, you can preserve type derivation in XML documents that you create or load programmatically. Taking the schema used in this example, the following code listing illustrates how to create a new book of derived type DictionaryType:

```
// Append a dictionary (book of derived type) and populate its attributes and elements
DictionaryType dictionary = new DictionaryType(root.Book.Append().Node);
dictionary.ID.Value = 2;
dictionary.Title.Append().Value = "English-German Dictionary";
dictionary.Author.Append().Value = "John Doe";
dictionary.FromLanguage.Append().Value = "English";
dictionary.ToLanguage.Append().Value = "German";
// Since it's a derived type, make sure to set the xsi:type attribute of the book
element
dictionary.SetXsiType();
```

Note that it is important to set the xsi:type attribute of the newly created book. This ensures that the book type will be interpreted correctly by the schema when the XML document is validated.

When you load data from an XML document, the following code listing shows how to identify a book of derived type DictionaryType in the loaded XML instance. First, the code finds the value of the xsi:type attribute of the book node. If the namespace URI of this node is http://www.nanonull.com/LibrarySample, and if the URI lookup prefix and type matches the value of the xsi:type attribute, then this is a dictionary:

```
// Determine if this book is of derived type
if (book.Node.Attributes.GetNamedItem("xsi:type") != null)
{
    // Find the value of the xsi:type attribute
    string xsiTypeValue = book.Node.Attributes.GetNamedItem("xsi:type").Value;
    // Get the namespace URI and the lookup prefix of this namespace
    string namespaceUri = book.Node.NamespaceURI;
    string prefix = book.Node.GetPrefixOfNamespace(namespaceUri);
    // if this book has DictionaryType
```

14.6.5.3 Reading and Writing XML Documents (Java)

After you generate code from the <u>example schema</u>, a test Java project is created, along with several supporting Altova libraries.

About the generated Java libraries

The central class of the generated code is the Doc2 class, which represents the XML document. Such a class is generated for every schema and its name depends on the schema file name (**Doc.xsd**, in this example). Note that this class is called Doc2 to avoid a possible conflict with the namespace name. As shown in the diagram, this class provides methods for loading documents from files, binary streams, or strings (or saving documents to files, streams, strings). For a description of this class, see the <u>com.[YourSchema].[Doc]</u> class reference.

	Doc2
	Library: MemberElement_Library
\diamond	getStaticInfo():ComplexType
	loadFromFile(in filename:String):Doc2
	loadFromString(in xmlstring:String):Doc2
	loadFromBinary[in binary:byte[*]):Doc2
	saveToFile(in filename:String, in prettyPrint:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmlDecl:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in encoding:String):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmlDecl:boolean, in encoding:String):void
	saveToFile(in filename:String, in prettyPrint:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean):void
	saveToFile(in filename:String, in prettyPrint:boolean, in omitXmIDecl:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean):void
	saveToString(in prettyPrint:boolean):String
	saveToString(in prettyPrint:boolean, in omitXmlDecl:boolean):String
	saveToBinary(in prettyPrint:boolean):byte[*]
	saveToBinary(in prettyPrint:boolean, in encoding:String):byte[*]
	saveToBinary(in prettyPrint:boolean, in encoding:String, in bBigEndian:boolean, in bBOM:boolean):byte[*]
	<pre>createDocument():Doc2</pre>
	setSchemaLocation(in schemaLocation:String):void
	declareAlINamespacesFromSchema(in node:TypeBase):void
	«constructor» Doc2(in init:org.w3c.dom.Node)
	setXsiType():void
	«static» MemberElement_Library

The Library member of the Doc2 class represents the actual root of the document.

According to the code generation rules mentioned in <u>About Generated Java Code</u>, member classes are generated for each attribute and for each element of a type. In the generated code, the name of such member classes is prefixed with <u>MemberAttribute_</u> and <u>MemberElement_</u>, respectively. In the diagram below, examples of such classes are <u>MemberAttribute_ID</u> and <u>MemberElement_Author</u>, generated from the **Author** element and **ID** attribute of a book, respectively. Such classes enable you to manipulate programmatically the corresponding elements and attributes in the instance XML document (for example, append, remove, set value, etc). For more information, see the <u>com.[YourSchema].</u>[<u>YourSchemaType].MemberAttribute</u> and <u>com.[YourSchema].[YourSchemaType].MemberElement</u> Class reference.

Since the **DictionaryType** is a complex type derived from **BookType** in the schema, this relationship is also reflected in the generated classes. As illustrated in the diagram below, the class <code>DictionaryType</code> inherits the <code>BookType</code> class.



If your XML schema defines simple types as enumerations, the enumerated values become available as Enum values in the generated code. In the schema used in this example, a book format can be hardcover, paperback, e-book, and so on. Therefore, in the generated code, these values would be available through an Enum that is a member of the BookFormatType class.

Writing an XML document

- 1. On the File menu of Eclipse, click Import, select Existing Projects into Workspace, and click Next.
- 2. Next to **Select root directory**, click **Browse**, select the directory to which you generated the Java code, and then click **Finish**.
- 3. In the Eclipse Package Explorer, expand the **com.LibraryTest** package and open the **LibraryTest.java** file.

While prototyping an application from a frequently changing XML schema, you may need to frequently generate code to the same directory, so that the schema changes are immediately reflected in the code. Note that the generated test application and the Altova libraries are overwritten every time when you generate code into the same target directory. Therefore, do not add code to the generated test application. Instead, integrate the Altova libraries into your project (see Integrating Schema Wrapper Libraries).

4. Edit the Example() method as shown below.

```
protected static void example() throws Exception {
      // create a new, empty XML document
      Doc2 libDoc = Doc2.createDocument();
      // create the root element <Library> and add it to the document
      LibraryType lib = libDoc.Library.append();
      // set the "LastUpdated" attribute
      com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
      lib.LastUpdated.setValue(dt);
      // create a new <Book> and populate its elements and attributes
      BookType book = lib.Book.append();
      book.ID.setValue(java.math.BigInteger.valueOf(1));
      book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
      book.Title.append().setValue("The XML Spy Handbook");
      book.Author.append().setValue("Altova");
      // create a dictionary (book of derived type) and populate its elements and
attributes
      DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
      dict.ID.setValue(java.math.BigInteger.valueOf(2));
      dict.Title.append().setValue("English-German Dictionary");
      dict.Format.setEnumerationValue(BookFormatType.EE BOOK);
      dict.Author.append().setValue("John Doe");
      dict.FromLang.append().setValue("English");
      dict.ToLang.append().setValue("German");
      dict.setXsiType();
      // set the schema location (this is optional)
      libDoc.setSchemaLocation("Library.xsd");
     // save the XML document to a file with default encoding (UTF-8). "true" causes
the file to be pretty-printed.
     libDoc.saveToFile("Library1.xml", true);
  }
```

5. Build the Java project and run it. If the code is executed successfully, a **Library1.xml** file is created in the project directory.

Reading an XML document

- 1. On the File menu of Eclipse, click Import, select Existing Projects into Workspace, and click Next.
- 2. Next to **Select root directory**, click **Browse**, select the directory to which you generated the Java code, and then click **Finish**.
- 3. Save the code below as **Library1.xml** to a local directory (you will need to refer to the path of the **Library1.xml** file from the sample code below).

```
<?xml version="1.0" encoding="utf-8"?>
<Library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nanonull.com/LibrarySample"
xsi:schemaLocation="http://www.nanonull.com/LibrarySample Library.xsd"
LastUpdated="2016-02-03T17:10:08.4977404">
  <Book ID="1" Format="E-book">
      <Title>The XMLSpy Handbook</Title>
      <Author>Altova</Author>
   </Book>
   <Book ID="2" Format="Paperback" xmlns:n1="http://www.nanonull.com/LibrarySample"
xsi:type="n1:DictionaryType">
      <Title>English-German Dictionary</Title>
      <Author>John Doe</Author>
      <FromLang>English</FromLang>
      <ToLang>German</ToLang>
   </Book>
</Library>
```

- 4. In the Eclipse Package Explorer, expand the **com.LibraryTest** package and open the **LibraryTest.java** file.
- 5. Edit the Example() method as shown below.

```
protected static void example() throws Exception {
     // load XML document from a path, make sure to adjust the path as necessary
     Doc2 libDoc = Doc2.loadFromFile("Library1.xml");
     // get the first (and only) root element <Library>
    LibraryType lib = libDoc.Library.first();
     // check whether an element exists:
     if (!lib.Book.exists()) {
        System.out.println("This library is empty.");
        return;
     // read a DateTime schema type
     com.altova.types.DateTime dt = lib.LastUpdated.getValue();
     System.out.println("The library was last updated on: " + dt.toDateString());
     // iteration: for each <Book>...
     for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
        BookType book = (BookType) itBook.next();
        // output values of ID attribute and (first and only) title element
        System.out.println("ID: " + book.ID.getValue());
        System.out.println("Title: " + book.Title.first().getValue());
        // read and compare an enumeration value
        if (book.Format.getEnumerationValue() == BookFormatType.EPAPERBACK)
           System.out.println("This is a paperback book.");
        // for each <Author>...
        for (java.util.Iterator itAuthor = book.Author.iterator(); itAuthor
              .hasNext();)
           System.out.println("Author: " + ((com.Doc.xs.stringType)
itAuthor.next()).getValue());
```

```
// find the derived type of this book
        // by looking at the value of the xsi:type attribute, using DOM
        org.w3c.dom.Node bookNode = book.getNode();
        if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
           // Get the value of the xsi:type attribute
          String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();
           // Get the namespace URI and lookup prefix of this namespace
           String namespaceUri = bookNode.getNamespaceURI();
           String lookupPrefix = bookNode.lookupPrefix(namespaceUri);
           // If xsi:type matches the namespace URI and type of the book node
           if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                 && ( xsiTypeValue.equals(lookupPrefix + ":DictionaryType")))
                                                                                    {
              // ...then this is a book of derived type (dictionary)
              DictionaryType dictionary = new DictionaryType( book.getNode());
              // output the value of the "FromLang" and "ToLang" elements
              System.out.println("From language: " +
dictionary.FromLang.first().getValue());
              System.out.println("To language: " +
dictionary.ToLang.first().getValue());
           else
              // throw an error
              throw new java.lang.Error("This book has an unknown type.");
           }
        }
     }
   }
```

6. Build the Java project and run it. If the code is executed successfully, **Library1.xml** will be read by the program code, and its contents displayed in the Console view.

Reading and writing elements and attributes

Values of attributes and elements can be accessed using the getValue() method of the generated member element or attribute class, for example:

```
// output values of ID attribute and (first and only) title element
System.out.println("ID: " + book.ID.getValue());
System.out.println("Title: " + book.Title.first().getValue());
```

To get the value of the **Title** element in this particular example, we also used the first() method, since this is the first (and only) **Title** element of a book. For cases when you need to pick a specific element from a list by index, use the at() method.

To iterate through multiple elements, use either index-based iteration or java.util.Iterator. For example, you can iterate through the books of a library as follows:

```
// index-based iteration
for (int j = 0; j < lib.Book.count(); ++j ) {
    // your code here
}</pre>
```

```
// alternative iteration using java.util.Iterator
for (java.util.Iterator itBook = lib.Book.iterator(); itBook.hasNext();) {
    // your code here
}
```

To add a new element, use the append() method. For example, the following code appends an empty root **Library** element to the document:

```
// create the root element <Library> and add it to the document
LibraryType lib = libDoc.Library.append();
```

Once an element is appended, you can set the value of any of its elements or an attributes by using the setValue() method.

```
// set the value of the Title element
book.Title.append().setValue("The XML Spy Handbook");
// set the value of the ID attribute
book.ID.setValue(java.math.BigInteger.valueOf(1));
```

Reading and writing enumeration values

If your XML schema defines simple types as enumerations, the enumerated values become available as Enum values in the generated code. In the schema used in this example, a book format can be hardcover, paperback, e-book, and so on. Therefore, in the generated code, these values would be available through an Enum (see the BookFormatType class diagram above). To assign enumeration values to an object, use code such as the one below:

```
// set an enumeration value
book.Format.setEnumerationValue( BookFormatType.EPAPERBACK );
```

You can read such enumeration values from XML instance documents as follows:

When an "if" condition is not enough, create a switch to determine each enumeration value and process it as required.

Working with xs:dateTime and xs:duration types

If the schema from which you generated code uses time and duration types such as xs:dateTime, or xs:duration, these are converted to Altova native classes in generated code. Therefore, to write a date or duration value to the XML document, do the following:

- 1. Construct a com.altova.types.DateTime or com.altova.types.Duration object.
- 2. Set the object as value of the required element or attribute, for example:

```
// set the value of an attribute of DateTime type
com.altova.types.DateTime dt = new com.altova.types.DateTime(DateTime.now());
lib.LastUpdated.setValue(dt);
```

To read a date or duration from an XML document:

- 1. Declare the element value (or attribute) as <u>com.altova.types.DateTime</u> or <u>com.altova.types.Duration</u> object.
- 2. Format the required element or attribute, for example:

```
// read a DateTime type
com.altova.types.DateTime dt = lib.LastUpdated.getValue();
   System.out.println("The library was last updated on: " + dt.toDateString());
```

For more information, see <u>com.altova.types.DateTime</u> and <u>com.altova.types.Duration</u> class reference.

Working with derived types

If your XML schema defines derived types, you can preserve type derivation in XML documents that you create or load programmatically. Taking the schema used in this example, the following code listing illustrates how to create a new book of derived type DictionaryType:

```
// create a dictionary (book of derived type) and populate its elements and attributes
DictionaryType dict = new DictionaryType(lib.Book.append().getNode());
dict.ID.setValue(java.math.BigInteger.valueOf(2));
dict.Title.append().setValue("English-German Dictionary");
dict.Format.setEnumerationValue(BookFormatType.EE_BOOK);
dict.Author.append().setValue("John Doe");
dict.FromLang.append().setValue("English");
dict.ToLang.append().setValue("German");
dict.setXsiType();
```

Note that it is important to set the xsi:type attribute of the newly created book. This ensures that the book type will be interpreted correctly by the schema when the XML document is validated.

When you load data from an XML document, the following code listing shows how to identify a book of derived type DictionaryType in the loaded XML instance. First, the code finds the value of the xsi:type attribute of the book node. If the namespace URI of this node is http://www.nanonull.com/LibrarySample, and if the URI lookup prefix and type matches the value of the xsi:type attribute, then this is a dictionary:

```
// find the derived type of this book
         // by looking at the value of the xsi:type attribute, using DOM
         org.w3c.dom.Node bookNode = book.getNode();
         if (bookNode.getAttributes().getNamedItem("xsi:type") != null) {
            // Get the value of the xsi:type attribute
            String xsiTypeValue =
bookNode.getAttributes().getNamedItem("xsi:type").getNodeValue();
            // Get the namespace URI and lookup prefix of the book node
            String namespaceUri = bookNode.getNamespaceURI();
            String lookupPrefix = bookNode.lookupPrefix(namespaceUri);
            // If xsi:type matches the namespace URI and type of the book node
            if (namespaceUri == "http://www.nanonull.com/LibrarySample"
                  && ( xsiTypeValue.equals(lookupPrefix + ":DictionaryType")))
                                                                                     {
               // ...then this is a book of derived type (dictionary)
               DictionaryType dictionary = new DictionaryType( book.getNode());
```

14.6.6 Example: Purchase Order

This example illustrates how to work with program code generated from a "main" XML schema that imports other schemas. Each of the imported schema has a different target namespace. The goal here is to create programmatically an XML document where all elements are prefixed according to their namespace. More specifically, the XML document created from your C++, C#, or Java code should look like the one below:

```
<?xml version="1.0" encoding="utf-8"?>
<p:Purchase xsi:schemaLocation="http://NamespaceTest.com/Purchase Main.xsd"
            xmlns:p="http://NamespaceTest.com/Purchase"
            xmlns:o="http://NamespaceTest.com/OrderTypes"
            xmlns:c="http://NamespaceTest.com/CustomerTypes"
            xmlns:cmn="http://NamespaceTest.com/CommonTypes"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <p:OrderDetail>
        <o:Item>
            <o:ProductName>Lawnmower</o:ProductName>
            <o:Quantity>1</o:Quantity>
            <o:UnitPrice>148.42</o:UnitPrice>
        </o:Item>
   </p:OrderDetail>
   <p:PaymentMethod>VISA</p:PaymentMethod>
    <p:CustomerDetails>
        <c:Name>Alice Smith</c:Name>
        <c:DeliveryAddress>
            <cmn:Line1>123 Maple Street</cmn:Line1>
            <cmn:Line2>Mill Valley</cmn:Line2>
        </c:DeliveryAddress>
        <c:BillingAddress>
            <cmn:Line1>8 Oak Avenue</cmn:Line1>
            <cmn:Line2>Old Town</cmn:Line2>
        </c:BillingAddress>
   </p:CustomerDetails>
</p:Purchase>
```

The main schema used in this example is called **Main.xsd**. As illustrated in the code listing below, it imports three other schemas: **CommonTypes.xsd**, **CustomerTypes.xsd**, and **OrderTypes.xsd**. To get the same results as in this example, save all the code listings below to files, and use the same file names as above. Notice that the schema maps each of the prefixes **ord**, **pur**, **cmn**, and **cust** to some namespace (Order types, Purchase types, Common types, and Customer types, respectively). This means that, in the generated code,

the classes corresponding to Orders, Purchases, Customers, and so on, will be available under their respective namespace.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
      targetNamespace="http://NamespaceTest.com/Purchase"
      xmlns:ord="http://NamespaceTest.com/OrderTypes"
      xmlns:pur="http://NamespaceTest.com/Purchase"
      xmlns:cmn="http://NamespaceTest.com/CommonTypes"
      xmlns:cust="http://NamespaceTest.com/CustomerTypes"
      elementFormDefault="qualified">
    <xs:import schemaLocation="CommonTypes.xsd"</pre>
namespace="http://NamespaceTest.com/CommonTypes" />
    <xs:import schemaLocation="CustomerTypes.xsd"</pre>
namespace="http://NamespaceTest.com/CustomerTypes" />
    <xs:import schemaLocation="OrderTypes.xsd"</pre>
namespace="http://NamespaceTest.com/OrderTypes" />
    <xs:element name="Purchase">
        <re><xs:complexType>
            <xs:sequence>
                <xs:element name="OrderDetail" type="ord:OrderType" />
                <xs:element name="PaymentMethod" type="cmn:PaymentMethodType" />
                <xs:element ref="pur:CustomerDetails" />
            </xs:sequence>
        </ms:complexType>
    </ms:element>
    <xs:element name="CustomerDetails" type="cust:CustomerType" />
</xs:schema>
```

Main.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
      targetNamespace="http://NamespaceTest.com/CommonTypes"
      elementFormDefault="qualified">
   <xs:complexType name="AddressType">
      <xs:sequence>
         <xs:element name="Line1" type="xs:string"/>
         <xs:element name="Line2" type="xs:string"/>
      </xs:sequence>
   </ms:complexType>
   <xs:simpleType name="PriceType">
      <xs:restriction base="xs:decimal">
         <xs:fractionDigits value="2"/>
      </ms:restriction>
  </xs:simpleType>
   <xs:simpleType name="PaymentMethodType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="VISA"/>
         <xs:enumeration value="MasterCard"/>
         <xs:enumeration value="Cash"/>
         <xs:enumeration value="AMEX"/>
      </ms:restriction>
   </xs:simpleType>
</xs:schema>
```

CommonTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://NamespaceTest.com/CustomerTypes"
xmlns:cmn="http://NamespaceTest.com/CommonTypes"
elementFormDefault="qualified">
<xs:import schemaLocation="CommonTypes.xsd"
namespace="http://NamespaceTest.com/CommonTypes" />
<xs:complexType name="CustomerType">
<xs:complexType name="CustomerType">
<xs:sequence>
<xs:element name="Name" type="xs:string" />
<xs:element name="DeliveryAddress" type="cmn:AddressType" />
<xs:element name="BillingAddress" type="cmn:AddressType" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

CustomerTypes.xsd



OrderTypes.xsd

To complete this example, take the following steps:

- 1. Save all schemas from the code listings above to files on the disk, making sure that you preserve the indicated file names.
- 2. Generate the schema wrapper code from the **Main.xsd** schema above, using the steps described in <u>Generating Code from XML Schemas or DTD</u>. After completing this step, you should have generated a compilable program in the language of your choice (C++, C#, or Java).
- 3. Add code to your C++, C#, or Java program from one the following example code listings, as required:
 - XML Namespaces and Prefixes (C++)

- XML Namespaces and Prefixes (C#)
- XML Namespaces and Prefixes (Java)

14.6.6.1 XML Namespaces and Prefixes (C++)

After you generate code from the <u>example schema</u>, a test C++ application is created, along with several supporting Altova libraries. Recall that the example schema (**Main.xsd**) has multiple namespace declarations. Consequently, the generated code includes namespaces that correspond to namespace aliases (prefixes) from the schema, namely: <u>Main::ord</u>, <u>Main::pur</u>, <u>Main::cmn</u>, and <u>Main::cust</u>.

In general, in order to control XML namespaces and prefixes with the help of the schema wrapper libraries, you have the following methods at your disposal:

- <u>DeclareAllNamespacesFromSchema()</u>. Call this method if you want to declare the same namespaces in your XML instance as in the schema. Otherwise, if you need different namespaces as in this example, then <u>DeclareNamespace()</u> should be used. The method <u>DeclareAllNamespacesFromSchema()</u> is not used in this example because we specifically want to create XML elements with prefixes that are slightly different from those declared in the schema.
- <u>DeclareNamespace()</u>. Call this method to create or override the existing namespace prefix attribute on an element. The element must already be created using either the <u>append()</u> or <u>appendWithPrefix()</u> methods, as further illustrated below.
- <u>appendWithPrefix()</u>. Use this method to append an instance element with a specific prefix. To create the XML instance illustrated in this example, it was sufficient to call this method for the root element only. All other elements were appended using just <u>append()</u>, and their prefixes were added automatically based on their namespaces, according to the rules above.

The code listing below shows you how to create an XML document with multiple namespace declarations and prefixed element names. Specifically, it generates a Purchase Order instance as illustrated in the <u>Example:</u> <u>Purchase Order</u>. Importantly, for illustrative purposes, some prefixes are overridden in the XML instance (that is, they are not exactly the same as the ones declared in the schema).

```
void Example()
   // Create the XML document and append the root element
   Main::pur::CMain doc = Main::pur::CMain::CreateDocument();
   Main::pur::CPurchaseType purchase = doc.Purchase.appendWithPrefix( T("p"));
   // Set schema location
   doc.SetSchemaLocation( T("Main.xsd"));
   // Declare namespaces on root element
   purchase.DeclareNamespace(_T("o"), _T("http://NamespaceTest.com/OrderTypes"));
   purchase.DeclareNamespace(_T("c"), _T("http://NamespaceTest.com/CustomerTypes"));
purchase.DeclareNamespace(_T("cmn"), _T("http://NamespaceTest.com/CommonTypes"));
   // Append the OrderDetail element
   Main::ord::COrderType order = purchase.OrderDetail.append();
   Main::ord::CltemType item = order.Item.append();
   item.ProductName.append() = T("Lawnmower");
   item.Quantity.append() = 1;
   item.UnitPrice.append() = 148.42;
   // Append the PaymentMethod element
   Main::cmn::CPaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
```

}

```
paymentMethod.SetEnumerationValue(Main::cmn::CPaymentMethodTypeType::k_VISA);
// Append the CustomerDetails element
Main::cust::CCustomerType customer = purchase.CustomerDetails.append();
customer.Name.append() = _T("Alice Smith");
Main::cmn::CAddressType deliveryAddress = customer.DeliveryAddress.append();
deliveryAddress.Line1.append() = _T("123 Maple Street");
deliveryAddress.Line2.append() = _T("Mill Valley");
Main::cmn::CAddressType billingAddress = customer.BillingAddress.append();
billingAddress.Line1.append() = _T("8 Oak Avenue");
billingAddress.Line2.append() = _T("Old Town");
// Save to file and release object from memory
doc.SaveToFile(_T("Main1.xml"), true);
doc.DestroyDocument();
```

14.6.6.2 XML Namespaces and Prefixes (C#)

After you generate code from the <u>example schema</u>, a test C# application is created, along with several supporting Altova libraries. Recall that the example schema (**Main.xsd**) has multiple namespace declarations. Consequently, the generated code includes namespaces that correspond to namespace aliases (prefixes) from the schema, namely: <u>Main.ord</u>, <u>Main.pur</u>, <u>Main.cmn</u>, and <u>Main.cust</u>.

In general, in order to control XML namespaces and prefixes with the help of the schema wrapper libraries, you have the following methods at your disposal:

- <u>DeclareAllNamespacesFromSchema()</u>. Call this method if you want to declare the same namespaces in your XML instance as in the schema. Otherwise, if you need different namespaces as in this example, then <u>DeclareNamespace()</u> should be used. The method <u>DeclareAllNamespacesFromSchema()</u> is not used in this example because we specifically want to create XML elements with prefixes that are slightly different from those declared in the schema.
- <u>DeclareNamespace()</u>. Call this method to create or override the existing namespace prefix attribute on an element. The element must already be created using either the <u>Append()</u> or <u>AppendWithPrefix()</u> methods, as further illustrated below.
- <u>AppendWithPrefix()</u>. Use this method to append an instance element with a specific prefix. To create
 the XML instance illustrated in this example, it was sufficient to call this method for the root element
 only. All other elements were appended using just <u>Append()</u>, and their prefixes were added
 automatically based on their namespaces, according to the rules above.

The code listing below shows you how to create an XML document with multiple namespace declarations and prefixed element names. Specifically, it generates a Purchase Order instance as illustrated in the <u>Example:</u> <u>Purchase Order</u>. Importantly, for illustrative purposes, some prefixes are overridden in the XML instance (that is, they are not exactly the same as the ones declared in the schema).

```
protected static void Example()
{
    // Create the XML document and append the root element
    pur.Main2 doc = pur.Main2.CreateDocument();
    pur.PurchaseType purchase = doc.Purchase.AppendWithPrefix("p");
    // Set schema location
    doc.SetSchemaLocation(@"Main.xsd");
```

}

```
// Declare namespaces on root element
purchase.DeclareNamespace("o", "http://NamespaceTest.com/OrderTypes");
purchase.DeclareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
purchase.DeclareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");
// Append the OrderDetail element
ord.OrderType order = purchase.OrderDetail.Append();
ord.ItemType item = order.Item.Append();
item.ProductName.Append().Value = "Lawnmower";
item.Quantity.Append().Value = 1;
item.UnitPrice.Append().Value = 148.42M;
// Append the PaymentMethod element
cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.Append();
paymentMethod.EnumerationValue = cmn.PaymentMethodTypeType.EnumValues.eVISA;
// Append the CustomerDetails element
cust.CustomerType customer = purchase.CustomerDetails.Append();
customer.Name.Append().Value = "Alice Smith";
cmn.AddressType deliveryAddress = customer.DeliveryAddress.Append();
deliveryAddress.Line1.Append().Value = "123 Maple Street";
deliveryAddress.Line2.Append().Value = "Mill Valley";
cmn.AddressType billingAddress = customer.BillingAddress.Append();
billingAddress.Line1.Append().Value = "8 Oak Avenue";
billingAddress.Line2.Append().Value = "Old Town";
// Save to file
doc.SaveToFile("PurchaseOrder.xml", true);
```

14.6.6.3 XML Namespaces and Prefixes (Java)

After you generate code from the <u>example schema</u>, a test Java application is created, along with several supporting Altova libraries. Recall that the example schema (**Main.xsd**) has multiple namespace declarations. Consequently, the generated code includes namespaces that correspond to namespace aliases (prefixes) from the schema, namely: <u>com.Main.ord</u>, <u>com.Main.pur</u>, <u>com.Main.cmn</u>, and <u>com.Main.cust</u>.

In general, in order to control XML namespaces and prefixes with the help of the schema wrapper libraries, you have the following methods at your disposal:

- <u>declareAllNamespacesFromSchema()</u>. Call this method if you want to declare the same namespaces in your XML instance as in the schema. Otherwise, if you need different namespaces as in this example, then <u>declareNamespace()</u> should be used. The method <u>declareAllNamespacesFromSchema()</u> is not used in this example because we specifically want to create XML elements with prefixes that are slightly different from those declared in the schema.
- <u>declareNamespace()</u>. Call this method to create or override the existing namespace prefix attribute on an element. The element must already be created using either the <u>append()</u> or <u>appendWithPrefix()</u> methods, as further illustrated below.
- <u>appendWithPrefix()</u>. Use this method to append an instance element with a specific prefix. To create
 the XML instance illustrated in this example, it was sufficient to call this method for the root element
 only. All other elements were appended using just <u>append()</u>, and their prefixes were added
 automatically based on their namespaces, according to the rules above.

The code listing below shows you how to create an XML document with multiple namespace declarations and prefixed element names. Specifically, it generates a Purchase Order instance as illustrated in the <u>Example:</u>

<u>Purchase Order</u>. Importantly, for illustrative purposes, some prefixes are overridden in the XML instance (that is, they are not exactly the same as the ones declared in the schema).

```
protected static void example() throws Exception {
   // Create the XML document and append the root element
   com.Main.pur.Main2 doc = com.Main.pur.Main2.createDocument();
   com.Main.pur.PurchaseType purchase = doc.Purchase.appendWithPrefix("p");
   // Set schema location
  doc.setSchemaLocation("Main.xsd");
   // Declare namespaces on root element
  purchase.declareNamespace("o", "http://NamespaceTest.com/OrderTypes");
  purchase.declareNamespace("c", "http://NamespaceTest.com/CustomerTypes");
  purchase.declareNamespace("cmn", "http://NamespaceTest.com/CommonTypes");
   // Append the OrderDetail element
  com.Main.ord.OrderType order = purchase.OrderDetail.append();
  com.Main.ord.ItemType item = order.Item.append();
   item.ProductName.append().setValue("Lawnmower");
  item.Quantity.append().setValue(1);
   java.math.BigDecimal price = new java.math.BigDecimal("148.42");
  item.UnitPrice.append().setValue(price);
   // Append the PaymentMethod element
   com.Main.cmn.PaymentMethodTypeType paymentMethod = purchase.PaymentMethod.append();
   paymentMethod.setEnumerationValue(com.Main.cmn.PaymentMethodTypeType.EVISA);
   // Append the CustomerDetails element
   com.Main.cust.CustomerType customer = purchase.CustomerDetails.append();
   customer.Name.append().setValue("Alice Smith");
   com.Main.cmn.AddressType deliveryAddress = customer.DeliveryAddress.append();
  deliveryAddress.Line1.append().setValue("123 Maple Street");
  deliveryAddress.Line2.append().setValue("Mill Valley");
  com.Main.cmn.AddressType billingAddress = customer.BillingAddress.append();
  billingAddress.Line1.append().setValue("8 Oak Avenue");
  billingAddress.Line2.append().setValue("Old Town");
   // Save to file
  doc.saveToFile("PurchaseOrder.xml", true);
}
```

14.7 Reference to Generated Classes (C++)

This chapter includes a description of C++ classes generated with MapForce from a DTD or XML schema (see <u>Generating Code from XML Schemas or DTDs</u>). You can integrate these classes into your code to read, modify, and write XML documents.

Note: The generated code does include other supporting classes, which are not listed here and are subject to modification.

14.7.1 altova::DateTime

This class enables you to process XML attributes or elements that have date and time types, such as xs:dateTime.

Constructors

Name	Description
DateTime()	Initializes a new instance of the DateTime class to 12:00:00 midnight, January 1, 0001.
DateTime(int64 value, short timezone)	Initializes a new instance of the DateTime class. The value parameter represents the number of ticks (100-nanosecond intervals) that have elapsed since 12:00:00 midnight, January 1, 0001.
DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second)	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, and second supplied as argument.
DateTime(int year, unsigned char month, unsigned char day, unsigned char hour, unsigned char minute, double second, short timezone)	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, second and timezone supplied as argument. The timezone is expressed in minutes and can be positive or negative. For example, the timezone "UTC-01:00" is expressed as "-60".

Name	Description
unsigned char Day() const	Returns the day of month of the current DateTime object. The return values range from 1 through 31.
int DayOfYear() const	Returns the day of year of the current DateTime object. The return values range from 1 through 366.
bool HasTimezone() const	Returns Boolean true if the current DateTime object has a timezone defined; false otherwise.

Name	Description
unsigned char Hour() const	Returns the hour of the current DateTime object. The return values range from 0 through 23.
static bool IsLeapYear(int year)	Returns Boolean true if the year of the DateTime class is a leap year; false otherwise.
unsigned char Minute() const	Returns the minute of the current DateTime object. The return values range from 0 through 59.
unsigned char Month() const	Returns the month of the current DateTime object. The return values range from 1 through 12.
int64 NormalizedValue() const	Returns the value of the DateTime object expressed as the Coordinated Universal Time (UTC).
double Second() const	Returns the second of the current DateTime object. The return values range from 0 through 59.
void SetTimezone(short tz)	Sets the timezone of the current DateTime object to the timezone value supplied as argument. The tz argument is expressed in minutes and can be positive or negative.
short Timezone() const	Returns the timezone, in minutes, of the current DateTime object. Before using this method, make sure that the object actually has a timezone, by calling the HasTimezone() method.
int64 Value() const	Returns the value of the DateTime object, expressed in the number of ticks (100-nanosecond intervals) that have elapsed since 12:00:00 midnight, January 1, 0001.
int Weekday() const	Returns the day of week of the current DateTime object, as an integer. Values range from 0 through 6, where 0 is Monday (ISO-8601).
int Weeknumber() const	Returns the number of week in the year of the current DateTime object. The return values are according to ISO-8601.
int WeekOfMonth() const	Returns the number of week in the month of the current DateTime object. The return values are according to ISO-8601.
int Year() const	Returns the year of the current DateTime object.

Example

```
void Example()
{
    // initialize a new DateTime instance to 12:00:00 midnight, January 1st, 0001
    altova::DateTime dt1 = altova::DateTime();
    // initialize a new DateTime instance using the year, month, day, hour, minute, and
    second
    altova::DateTime dt2 = altova::DateTime(2015, 11, 10, 9, 8, 7);
    // initialize a new DateTime instance using the year, month, day, hour, minute,
```

```
second, and UTC +01:00 timezone
   altova::DateTime dt = altova::DateTime(2015, 11, 22, 13, 53, 7, 60);
   // Get the value of this DateTime object
   std::cout << "The number of ticks of the DateTime object is: " << dt.Value() <<</pre>
std::endl;
   // Get the year
   cout << "The year is: " << dt.Year() << endl;</pre>
   // Get the month
   cout << "The month is: " << (int)dt.Month() << endl;</pre>
   // Get the day of the month
   cout << "The day of the month is: " << (int) dt.Day() << endl;</pre>
   // Get the day of the year
   cout << "The day of the year is: " << dt.DayOfYear() << endl;</pre>
   // Get the hour
   cout << "The hour is: " << (int) dt.Hour() << endl;</pre>
   // Get the minute
   cout << "The minute is: " << (int) dt.Minute() << endl;</pre>
   // Get the second
   cout << "The second is: " << dt.Second() << endl;</pre>
   // Get the weekday
   cout << "The weekday is: " << dt.Weekday() << endl;</pre>
   // Get the week number
   cout << "The week of year is: " << dt.Weeknumber() << endl;</pre>
   // Get the week in month
   cout << "The week of month is: " << dt.WeekOfMonth() << endl;</pre>
   // Check whether a DateTime instance has a timezone
   if (dt.HasTimezone() == TRUE)
   {
      // output the value of the Timezone
      cout << "The timezone is: " << dt.Timezone() << endl;</pre>
   }
   else
   {
      cout << "No timezone has been defined." << endl;</pre>
   }
   // Construct a DateTime object with a timezone UTC+01:00 (Vienna)
   altova::DateTime vienna dt = DateTime(2015, 11, 23, 14, 30, 59, +60);
   // Output the result in readable format
   cout << "The Vienna time: "</pre>
       << (int) vienna dt.Month()
       << "-" << (int) vienna_dt.Day()
       << " " << (int) vienna_dt.Hour()
       << ":" << (int) vienna dt.Minute()
       << ":" << (int) vienna dt.Second()
       << endl;
   // Convert the value to UTC time
   DateTime utc dt = DateTime(vienna dt.NormalizedValue());
   // Output the result in readable format
   cout << "The UTC time:
                              ...
      << (int) utc dt.Month()
      << "-" << (int) utc dt.Day()
      << " " << (int) utc_dt.Hour()
      << ":" << (int) utc_dt.Minute()
      << ":" << (int) utc_dt.Second()
```

```
<<pre><< endl;
// Check if a year is a leap year
int year = 2016;
if( altova::DateTime::IsLeapYear(year) )
{ cout << year << " is a leap year" << endl; }
else
{ cout << year << " is not a leap year" << endl; }</pre>
```

14.7.2 altova::Duration

This class enables you to process XML attributes or elements of type xs:duration.

Constructors

}

Name	Description
Duration()	Initializes a new instance of the Duration class to an empty value.
Duration(const DayTimeDuration& dt)	Initializes a new instance of the Duration class to a duration defined by the dt argument (see <u>altova::DayTimeDuration</u>).
Duration(const YearMonthDuration& ym)	Initializes a new instance of the Duration class to the duration defined by the ym argument (see <u>altova::YearMonthDuration</u>).
Duration(const YearMonthDuration& ym, const DayTimeDuration& dt)	Initializes a new instance of the Duration class to the duration defined by both the dt and the ym arguments (see <u>altova::YearMonthDuration</u> and <u>altova::DayTimeDuration</u>).

Name	Description
int Days() const	Returns the number of days in the current Duration instance.
DayTimeDuration DayTime() const	Returns the day and time duration in the current Duration instance, expressed as a DayTimeDuration object (see <u>altova::DayTimeDuration</u>).
int Hours() const	Returns the number of hours in the current Duration instance.
bool IsNegative() const	Returns Boolean true if the current Duration instance is negative.
bool IsPositive() const	Returns Boolean true if the current Duration instance is positive.
int Minutes() const	Returns the number of minutes in the current Duration instance.
int Months() const	Returns the number of months in the current Duration instance.
double Seconds() const	Returns the number of seconds in the current Duration instance.

Name	Description
YearMonthDuration YearMonth() const	Returns the year and month duration in the current Duration instance, expressed as a YearMonthDuration object (see <u>altova::YearMonthDuration</u>).
int Years() const	Returns the number of years in the current Duration instance.

Example

The following code listing illustrates creating a new Duration object, as well as reading values from it.

```
void ExampleDuration()
ł
   // Create an empty Duration object
   altova::Duration empty_duration = altova::Duration();
   // Create a Duration object using an existing duration value
   altova::Duration duration1 = altova::Duration(empty duration);
   // Create a YearMonth duration of six years and five months
   altova::YearMonthDuration yrduration = altova::YearMonthDuration(6, 5);
   // Create a DayTime duration of four days, three hours, two minutes, and one second
   altova::DayTimeDuration dtduration = altova::DayTimeDuration(4, 3, 2, 1);
   // Create a Duration object by combining the two previously created durations
   altova::Duration duration = altova::Duration(yrduration, dtduration);
   // Get the number of years in this Duration instance
   cout << "Years: " << duration.Years() << endl;</pre>
   // Get the number of months in this Duration instance
   cout << "Months: " << duration.Months() << endl;</pre>
   // Get the number of days in this Duration instance
   cout << "Days: " << duration.Days() << endl;</pre>
   // Get the number of hours in this Duration instance
   cout << "Hours: " << duration.Hours() << endl;</pre>
   // Get the number of hours in this Duration instance
   cout << "Minutes: " << duration.Minutes() << endl;</pre>
   // Get the number of seconds in this Duration instance
   cout << "Seconds: " << duration.Seconds() << endl;</pre>
}
```

14.7.3 altova::DayTimeDuration

This class enables you to process XML schema duration types that consist of a day and time part.

Constructors

Name	Description
DayTimeDuration()	Initializes a new instance of the DayTimeDuration class to an empty value.
DayTimeDuration(int days, int hours, int minutes, double seconds)	Initializes a new instance of the DayTimeDuration class to the number of days, hours, minutes, and seconds supplied as arguments.
explicit DayTimeDuration(int64 value)	Initializes a new instance of the DayTimeDuration class to a duration that consists of as many ticks (100-nanosecond intervals) as supplied in the value argument.

Name	Description
int Days() const	Returns the number of days in the current DayTimeDuration instance.
int Hours() const	Returns the number of hours in the current DayTimeDuration instance.
bool IsNegative() const	Returns Boolean true if the current DayTimeDuration instance is negative.
bool IsPositive() const	Returns Boolean true if the current DayTimeDuration instance is positive.
int Minutes() const	Returns the number of minutes in the current DayTimeDuration instance.
double Seconds() const	Returns the number of seconds in the current DayTimeDuration instance.
int64 Value() const	Returns the value (in ticks) of the current DayTimeDuration instance.

14.7.4 altova::YearMonthDuration

This class enables you to process XML schema duration types that consist of a year and month part.

Constructors

Name	Description
YearMonthDuration()	Initializes a new instance of the YearMonthDuration class to an empty value.
YearMonthDuration(int years, int months)	Initializes a new instance of the YearMonthDuration class to the number of years and months supplied in the years and months arguments.
explicit YearMonthDuration(int value)	Initializes a new instance of the YearMonthDuration class to a duration that consists of as many ticks (100-nanosecond intervals) as supplied in the value argument.

Methods

Name	Description
bool IsNegative() const	Returns Boolean true if the current YearMonthDuration instance is negative.
bool IsPositive() const	Returns Boolean true if the current YearMonthDuration instance is positive.
int Months() const	Returns the number of months in the current YearMonthDuration instance.
int Value() const	Returns the value (in ticks) of the current YearMonthDuration instance.
int Years()	Returns the number of years in the current YearMonthDuration instance.

14.7.5 altova::meta::Attribute

This class enables you to access schema information about classes generated from attributes. Note that this class is not meant to provide dynamic information about particular instances of an attribute in an XML document. Instead, it enables you to obtain programmatically information about a particular attribute defined in the XML schema.

Name	Description
SimpleType GetDataType()	Returns the type of the attribute content.

Name	Description
<pre>string_type GetLocalName()</pre>	Returns the local name of the attribute.
<pre>string_type GetNamespaceURI()</pre>	Returns the namespace URI of the attribute.
<pre>bool IsRequired()</pre>	Returns true if the attribute is required.

Operators

Name	Description
bool operator()	Returns true if this is not the NULL Attribute.
<pre>bool operator!()</pre>	Returns true if this is the NULL Attribute.

14.7.6 altova::meta::ComplexType

This class enables you to access schema information about classes generated from complex types. Note that this class is not meant to provide dynamic information about particular instances of a complex type in an XML document. Instead, it enables you to obtain programmatically information about a particular complex type defined in the XML schema.

Name	Description
Attribute FindAttribute(const char_type* localName, const char_type* namespaceURI)	Finds the attribute with the specified local name and namespace URI.
Element FindElement(const char_type* localName, const char_type* namespaceURI)	Finds the element with the specified local name and namespace URI.
<pre>std::vector<attribute> GetAttributes()</attribute></pre>	Returns a list of all attributes.
ComplexType GetBaseType()	Returns the base type of this type.
SimpleType GetContentType()	Returns the simple type of the content.
<pre>std::vector<element> GetElements()</element></pre>	Returns a list of all elements.
<pre>string_type GetLocalName()</pre>	Returns the local name of the type.
<pre>string_type GetNamespaceURI()</pre>	Returns the namespace URI of the type.

Operators

Name	Description
bool operator()	Returns true if this is not the NULL ComplexType.
<pre>bool operator!()</pre>	Returns true if this is the NULL ComplexType.

14.7.7 altova::meta::Element

This class enables you to access information about classes generated from schema elements. Note that this class is not meant to provide dynamic information about particular instances of an element in an XML document. Instead, it enables you to obtain programmatically information about a particular element defined in the XML schema.

Methods

Name	Description
ComplexType GetDataType()	Returns the type of the element. Note that this is always a complex type even if declared as simple in the original schema. Use GetContentType() of the returned object to get the simple content type.
<pre>string_type GetLocalName()</pre>	Returns the local name of the element.
unsigned int GetMaxOccurs()	Returns the maxOccurs value defined in the schema.
unsigned int GetMinOccurs()	Returns the minOccurs value defined in the schema.
<pre>string_type GetNamespaceURI()</pre>	Returns the namespace URI of the element.

Operators

Name	Description
bool operator()	Returns true if this is not the NULL Element.
bool operator!()	Returns true if this is the NULL Element.

14.7.8 altova::meta::SimpleType

This class enables you to access schema information about classes generated from simple types. Note that this class is not meant to provide dynamic information about particular instances of simple types in an XML document. Instead, it enables you to obtain programmatically information about a particular simple type defined in the XML schema.

Methods

Name	Description
SimpleType GetBaseType()	Returns the base type of this type.
<pre>std::vector<string_type> GetEnumerations()</string_type></pre>	Returns a list of all enumeration facets.
unsigned int GetFractionDigits()	Returns the value of this facet.
unsigned int GetLength()	Returns the value of this facet.
<pre>string_type GetLocalName()</pre>	Returns the local name of the type.
<pre>string_type GetMaxExclusive()</pre>	Returns the value of this facet.
<pre>string_type GetMaxInclusive()</pre>	Returns the value of this facet.
unsigned int GetMaxLength()	Returns the value of this facet.
<pre>string_type GetMinExclusive()</pre>	Returns the value of this facet.
<pre>string_type GetMinInclusive()</pre>	Returns the value of this facet.
unsigned int GetMinLength()	Returns the value of this facet.
<pre>string_type GetNamespaceURI()</pre>	Returns the namespace URI of the type.
<pre>std::vector<string_type> GetPatterns()</string_type></pre>	Returns a list of all pattern facets.
unsigned int GetTotalDigits()	Returns the value of this facet.
WhitespaceType GetWhitespace()	 Returns the value of the whitespace facet, which is one of: Whitespace_Unknown Whitespace_Preserve Whitespace_Replace Whitespace_Collapse

Operators

Name	Description
bool operator()	Returns true if this is not the NULL SimpleType.
bool operator!()	Returns true if this is the NULL SimpleType.

14.7.9 [YourSchema]::[CDoc]

When code is generated from an XML Schema, the generated code provides a document class with the same name as the schema. This class contains all possible root elements as members, as well as the following

methods. Note that, in the method names below, "CDoc" stands for the name of the generated document class itself.

Name	Description
static CDoc CreateDocument()	Creates a new, empty XML document. Must be released using DestroyDocument().
static void DeclareAllNamespacesFromSchema(El ementType& node)	Declares all namespaces from the XML Schema on the element supplied as argument (typically, the XML root element). Calling this method is useful if your schema has multiple namespace declarations, each mapped to a prefix, and you would like to declare all of them on the element supplied as argument.
<pre>void DestroyDocument()</pre>	Destroys a document. All references to the document and its nodes are invalidated. This must be called when you finished working with a document.
<pre>static CDoc LoadFromBinary(const std:vector<unsigned char="">& xml)</unsigned></pre>	Loads an XML document from a byte array.
static CDoc LoadFromFile(const string_type& fileName)	Loads an XML document from a file.
<pre>static CDoc LoadFromString(const string_type& xml)</pre>	Loads an XML document from a string.
std::vector <unsigned char=""> SaveToBinary(bool prettyPrint)</unsigned>	Saves an XML document to a byte array. When set to true, the <pre>prettyPrint</pre> argument re-formats the XML document for better readability.
<pre>std::vector<unsigned char=""> SaveToBinary(bool prettyPrint, const string_type & encoding)</unsigned></pre>	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding.
<pre>std::vector<unsigned char=""> SaveToBinary(bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)</unsigned></pre>	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
<pre>void SaveToFile(const string_type & fileName, bool prettyPrint)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting.
<pre>void SaveToFile(const string_type & fileName, bool omitXmlDecl)</pre>	Saves an XML document to a file. If the <code>omitXmlDecl</code> argument is set to true, the XML declaration will not be written.
<pre>void SaveToFile(const string_type & fileName, bool omitXmlDecl, const string_type & encoding)</pre>	Saves an XML document to a file with the specified encoding. If the <code>omitXmlDecl</code> argument is set to true, the XML declaration will not be written.
<pre>void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. Byte order and Unicode byte-order mark can be specified for Unicode encodings.

Name	Description
<pre>void SaveToFile(const string_type & fileName, bool prettyPrint, bool omitXmlDecl, const string_type & encoding, bool bBigEndian, bool bBOM, const</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding and the specified line end. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
<pre>string_type & lineend)</pre>	This method is only available if you generated the code for the Xerces3 XML library (see <u>Code Generator Options</u>).
<pre>void SaveToFile(const string_type& fileName, bool prettyPrint, bool omitXmlDecl,</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding and the specified line end.
const string_type & lineend)	This method is only available if you generated the code for the Xerces3 XML library (see <u>Code Generator Options</u>).
<pre>void SaveToFile(const string_type & fileName, bool prettyPrint, const string_type & encoding)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding.
<pre>void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
<pre>void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, bool bBigEndian, bool bBOM, const string_type & lineend)</pre>	Saves an XML document to a file with the specified encoding and the specified line end. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
	This method is only available if you generated the code for the Xerces3 XML library (see <u>Code Generator Options</u>).
<pre>void SaveToFile(const string_type& fileName, bool prettyPrint, const string_type & encoding, const string_type & lineend)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding and the specified line end.
	This method is only available if you generated the code for the Xerces3 XML library (see <u>Code Generator Options</u>).
<pre>string_type SaveToString(bool prettyPrint)</pre>	Saves an XML document to a string, with optional "pretty-print" formatting.
<pre>string_type SaveToString(bool prettyPrint, bool omitXmlDecl)</pre>	Saves an XML document to a string, with optional "pretty-print" formatting. If the <code>omitXmlDecl</code> argument is set to true, the XML declaration will not be written.
<pre>void SetDTDLocation(const string_type & dtdLocation)</pre>	Adds a DOCTYPE declaration with the specified system ID. A root element must already exist. This method is not supported for MSXML, since it is not possible to add a DOCTYPE declaration to a document in memory.
<pre>void SetSchemaLocation(const string_type & schemaLocation)</pre>	Adds an xsi:schemaLocation or xsi:noNamespaceSchemaLocation attribute to the root element. A root element must already exist.

14.7.10 [YourSchema]::[ElementType]

This class provides methods for manipulating XML elements from your schema. Methods of this class can be called on elements, not on the XML document itself. Note that, in order to call methods of this class, you don't need to instantiate the class directly. Any element created using the append() or appendWithPrefix() methods is of [ElementType] type.

Name	Description
<pre>void DeclareNamespace(const string_type prefix, const string_type nsURI)</pre>	This method takes two arguments that are both of string type: the prefix and the namespace URI that you want to use. The prefix supplied as argument will be mapped to the namespace URI value supplied as argument. If the prefix supplied as argument is empty, the method creates or overrides the default namespace declaration in the element. For example, let's assume that the XML document has an XML element called "purchase". If you call
	<pre>purchase.DeclareNamespace(_T("ord"), _T("http://OrderTypes"));</pre>
	then the XML document becomes
	<purchase xmlns:ord="http://OrderTypes"></purchase>
	Another example, if you call:
	<pre>purchase.DeclareNamespace(_T(""), _T("http://OrderTypes"));</pre>
	then the XML document becomes
	<purchase xmlns="http://OrderTypes"></purchase>
	Note: The declared namespace is used when appending subsequent child elements or attributes, according to the following rules:
	1. If the child namespace is the default, then use empty
	2. If the child namespace is equal to the parent one, then
	 use the parent prefix. Otherwise, search for nearest prefix from parent to top, using the lookup algorithm described in section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/namespaces-algorithms.html.

Name	Description
	 If there is no prefix for element namespace found, then use empty prefix.

14.7.11 [YourSchema]::MemberAttribute

When code is generated from an XML schema, a class such as this one is created for each member attribute of a type.

Methods

Name	Description
bool exists()	Returns true if the attribute exists.
int GetEnumerationValue()	Generated for enumeration types only. Returns one of the constants generated for the possible values, or "Invalid" if the value does not match any of the enumerated values in the schema.
altova::meta::Attribute info()	Returns an object for querying schema information (see <u>altova::meta::Attribute</u>).
void remove()	Removes the attribute from its parent element.
void SetEnumerationValue(int)	Generated for enumeration types only. Pass one of the constants generated for the possible values to this method to set the value.

14.7.12 [YourSchema]::MemberElement

When code is generated from an XML schema, a class such as this one is created for each member element of a type. In the descriptions below, "MemberType" stands for the name of the member element itself.

Name	Description
<pre>Iterator<membertype> all()</membertype></pre>	Returns an object for iterating instances of the member element.
MemberType append()	Creates a new element and appends it to its parent.
MemberType appendWithPrefix(string_type prefix)	Creates a new element having the prefix supplied as argument, and appends it to its parent. For an example, see <u>Example:</u> <u>Purchase Order</u> .
unsigned int count()	Returns the count of elements.

Name	Description
int GetEnumerationValue()	Generated for enumeration types only. Returns one of the constants generated for the possible values, or Invalid if the value does not match any of the enumerated values in the schema.
bool exists()	Returns true if at least one element exists.
MemberType first()	Returns the first instance of the member element.
MemberType operator[](unsigned int index)	Returns the member element specified by the index.
altova::meta::Element info()	Returns an object for querying schema information (see <u>altova::meta::Element</u>).
MemberType last()	Returns the last instance of the member element.
void remove()	Deletes all occurrences of the element from its parent.
<pre>void removeAt(unsigned int index)</pre>	Deletes the occurrence of the element specified by the index.
void SetEnumerationValue(int)	Generated for enumeration types only. Pass one of the constants generated for the possible values to this method to set the value.
14.8 Reference to Generated Classes (C#)

This chapter includes a description of C# classes generated with MapForce from a DTD or XML schema (see <u>Generating Code from XML Schemas or DTDs</u>). You can integrate these classes into your code to read, modify, and write XML documents.

Note: The generated code does include other supporting classes, which are not listed here and are subject to modification.

14.8.1 Altova.Types.DateTime

This class enables you to process XML attributes or elements that have date and time types, such as xs:dateTime.

Constructors

	Name	Description
=	DateTime(DateTime obj)	Initializes a new instance of the DateTime class to the DateTime object supplied as argument.
= \$	DateTime(System.DateTime newvalue)	Initializes a new instance of the DateTime class to the System.DateTime object supplied as argument.
=≬	DateTime(int year, int month, int day, int hour, int minute, double second, int offsetTZ)	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, second, and timezone offset supplied as arguments.
=\$	DateTime(int year, int month, int day, int hour, int minute, double second)	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, and second supplied as arguments.
=\$	DateTime(int year, int month, int day)	Initializes a new instance of the DateTime class to the year, month and day supplied as arguments.

Properties

	Name	Description
*	bool HasTimezone	Gets a Boolean value which indicates if the DateTime has a timezone.
*	static DateTime Now	Gets a DateTime object that is set to the current date and time on this computer.
*	short TimezoneOffset	Gets or sets the timezone offset, in minutes, of the DateTime object.
8	System.DateTime Value	Gets or sets the value of the DateTime object as a System.DateTime value.

	Name	Description	
-≡∳	int CompareTo(object obj)	The DateTime class implements the IComparable interface. This method compares the current instance of DateTime to another object and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. See also <u>https://msdn.microsoft.com/en-</u> us/library/system.icomparable.compareto(v=vs.110).aspx	
= \$	override bool Equals(object obj)	Returns true if the specified object is equal to the current object; false otherwise.	
- :	System.DateTime GetDateTime(bool correctTZ)	Returns a System.DateTime object from the current Altova.Types.DateTime instance. The correctTZ Boolean argument specifies whether the time of the returned object must be adjusted according to the timezone of the current Altova.Types.DateTime instance.	
=0	override int GetHashCode()	Returns the hash code of the current instance.	
=¢	int GetWeekOfMonth()	Returns the number of the week in month as an integer.	
=♥	static DateTime Parse(string s)	Creates a DateTime object from the string supplied as argument. For example, the following sample string values would be converted successfully to a DateTime object: 2015-01-01T23:23:23 2015-01-01 2015-11 23:23:23 An exception is raised if the string cannot be converted to a DateTime object. Note that this method is static and can only be called on the Altova.Types.DateTime class itself, not on an instance of the class.	
=∳	static DateTime Parse(string s, DateTimeFormat format)	Creates a DateTime object from a string, using the format supplied as argument. For the list of possible formats, see <u>Altova.Types.DateTimeFormat</u> . An exception is raised if the string cannot be converted to a DateTime object. Note that this method is static and can only be called on the Altova.Types.DateTime class itself, not on an instance of the class.	
=	override string ToString()	Converts the DateTime object to a string.	

	Name	Description
=\$	string ToString(DateTimeFormat format)	Converts the DateTime object to a string, using the format supplied as argument. For the list of possible formats, see <u>Altova.Types.DateTimeFormat</u> .

Operators

Name	Description	
!=	Determines if DateTime a is not equal to DateTime b.	
<	Determines if DateTime a is less than DateTime b.	
<=	Determines if DateTime a is less than or equal to DateTime b.	
==	Determines if DateTime a is equal to DateTime b.	
>	Determines if DateTime a is greater than DateTime b.	
>=	Determines if DateTime a is greater than or equal to DateTime b.	

Examples

Before using the following code listings in your program, ensure the Altova types are imported:

using Altova.Types;

The following code listing illustrates various ways to create DateTime objects:

```
protected static void DateTimeExample1()
   // Create a DateTime object from the current system time
   Altova.Types.DateTime dt = new Altova.Types.DateTime(System.DateTime.Now);
   Console.WriteLine("The current time is: " + dt.ToString());
   // Create an Altova DateTime object from parts (no timezone)
   Altova.Types.DateTime dt1 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33);
   Console.WriteLine("My custom time is : " + dt1.ToString());
   // Create an Altova DateTime object from parts (with UTC+60 minutes timezone)
   Altova.Types.DateTime dt2 = new Altova.Types.DateTime(2015, 10, 12, 10, 50, 33, 60);
   Console.WriteLine("My custom time with timezone is : " + dt2.ToString());
   // Create an Altova DateTime object by parsing a string
   Altova.Types.DateTime dt3 = Altova.Types.DateTime.Parse("2015-01-01T23:23:23");
   Console.WriteLine("Time created from string: " + dt3.ToString());
   // Create an Altova DateTime object by parsing a string formatted as schema date
  Altova.Types.DateTime dt4 = Altova.Types.DateTime.Parse("2015-01-01",
DateTimeFormat.W3 date);
   Console.WriteLine ("Time created from string formatted as schema date: " +
dt4.ToString());
}
```

The following code listing illustrates various ways to format DateTime objects:



14.8.2 Altova.Types.DateTimeFormat

The DateTimeFormat enum type has the following constant values:

Value	Description	Example
S_DateTime	Formats the value as standard dateTime, with a precision of a ten- millionth of a second, including timezone.	2015-11-12 12:19:03.9019132+01:00
S_Days	Formats the value as number of days elapsed since the UNIX epoch.	735913.6318973451087962962963
S_Seconds	Formats the value as number of seconds elapsed since the UNIX epoch, with a precision of a ten- millionth of a second.	63582937678.0769062
W3_date	Formats the value as schema date.	2015-11-12
W3_dateTime	Formats the value as schema dateTime.	2015-11-12T15:12:14.5194251
W3_gDay	Formats the value as schema gDay.	12 (assuming that the date is 12th of the month)
W3_gMonth	Formats the value as schema gMonth.	11

Value	Description	Example
		(assuming that the month is November)
W3_gMonthDay Formats the value as schema gMonthDay.		11-12 (assuming that the date is 12th of November)
W3_gYear	Formats the value as schema gYear.	2015 (assuming that the year is 2015)
W3_gYearMonth	Formats the value as schema gYearMonth.	2015-11 (assuming that the year is 2015 and the month is November)
W3_time	Formats the value as schema time, with a precision of a ten-millionth of a second.	15:19:07.5582719

14.8.3 Altova.Types.Duration

This class enables you to process XML attributes or elements of type xs:duration.

Constructors

	Name	Description	
=\$	Duration(Duration obj)	Initializes a new instance of the Duration class to the Duration object supplied as argument.	
=\$	Duration(System.TimeSp an newvalue)	Initializes a new instance of the Duration class to the System.TimeSpan object supplied as argument.	
=0	Duration(long ticks)	Initializes a new instance of the Duration class to the number of ticks supplied as argument.	
.≘∳	Duration(int newyears, int newmonths, int days, int hours, int minutes, int seconds, double partseconds, bool bnegative)	Initializes a new instance of the Duration class to a duration built from parts supplied as arguments.	

Properties

	Name	Description
2	int Months	Gets or sets the number of months of the current instance of Duration.
2	System.TimeSpan Value	Gets or sets the value (as System.TimeSpan) of the current instance of Duration.
*	int Years	Gets or sets the number of years of the current instance of Duration.

	Name	Description
=0	override bool Equals(object other)	Returns true if the specified object is equal to the current object; false otherwise.
=\$	override int GetHashCode()	Returns the hash code of the current instance.
=\$	<pre>bool IsNegative()</pre>	Returns true if the current instance of Duration represents a negative duration.
₩	static Duration Parse(string s, ParseType pt)	Returns an Altova.Types.Duration object parsed from the string supplied as argument, using the parse type supplied as argument. Valid parse type values:

	Name	Description	
		DURATION	Parse duration assuming that year, month, day, as well as time duration parts exist.
		YEARMON TH	Parse duration assuming that only year and month parts exist.
		DAYTIME	Parse duration assuming that only the day and time parts exist.
		Note that this r not on an insta	nethod is static and can only be called on the class itself, ance of the class.
#	override string ToString()	Converts the current Duration instance to string. For example, a time span of 3 hours, 4 minutes, and 5 seconds would be converted to "PT3H4M5S".	
=0	string ToYearMonthString()	Converts the current Duration instance to string, using the "Year and Month" parse type.	

Operators

Name	Description
! =	Determines if Duration a is not equal to Duration b.
==	Determines if Duration a is equal to Duration b.

Examples

Before using the following code listings in your program, ensure the Altova types are imported:

using Altova.Types;

The following code listing illustrates various ways to create Duration objects:

```
Duration dr2 = Altova.Types.Duration.Parse("-P4DT3H2M1S",
Duration.ParseType.DAYTIME);
// The output is -P4DT3H2M1S
Console.WriteLine("Duration created from string: " + dr2.ToString());
// Create a duration from ticks
Duration dr3 = new Duration(System.DateTime.UtcNow.Ticks);
// Output the result
Console.WriteLine("Duration created from ticks: " + dr3.ToString());
}
```

The following code listing illustrates getting values from Duration objects:

```
protected static void DurationExample2()
{
    // Create a negative Altova.Types.Duration from 6 years, 5 months, 4 days, 3 hours,
    // 2 minutes, 1 second, and .33 of a second
    Duration dr = new Duration(6, 5, 4, 3, 2, 1, .33, true);
    // The output is: -P6Y5M4DT3H2M1.33S
    Console.WriteLine("The complete duration is: " + dr.ToString());
    // Get only the year and month part as string
    string dr1 = dr.ToYearMonthString();
    Console.WriteLine("The YEARMONTH part is: " + dr1);
    // Get the number of years in duration
    Console.WriteLine("Years: " + dr.Years);
    // Get the number of months in duration
    Console.WriteLine("Months: " + dr.Months);
}
```

14.8.4 Altova.Xml.Meta.Attribute

This class enables you to access schema information about classes generated from attributes. Note that this class is not meant to provide dynamic information about particular instances of an attribute in an XML document. Instead, it enables you to obtain programmatically information about a particular attribute defined in the XML schema.

Properties

	Name	Description
	SimpleType DataType	Returns the type of the attribute content.
	string LocalName	Returns the local name of the attribute.
	string NamespaceURI	Returns the namespace URI of the attribute.
2	XmlQualifiedName QualifiedName	Returns the qualified name of the attribute.
2	<pre>bool Required()</pre>	Returns true if the attribute is required.

14.8.5 Altova.Xml.Meta.ComplexType

This class enables you to access schema information about classes generated from complex types. Note that this class is not meant to provide dynamic information about particular instances of a complex type in an XML document. Instead, it enables you to obtain programmatically information about a particular complex type defined in the XML schema.

Properties

	Name	Description
	Attribute[] Attributes	Returns a list of all attributes.
	ComplexType BaseType	Returns the base type of this type or null if no base type exists.
M	SimpleType ContentType	Returns the simple type of the content.
	Element[] Elements	Returns a list of all elements.
	string LocalName	Returns the local name of the type.
	string NamespaceURI	Returns the namespace URI of the type.
2	XmlQualifiedName QualifiedName	Returns the qualified name of this type.

Methods

	Name	Description
=0	ComplexType BaseType	Returns the base type of this type.
=	bool Equals(obj)	Checks if two info objects refer to the same type, based on qualified name comparison. Returns true if the type has the same qualified name.
=≬	Attribute FindAttribute(string localName, string namespaceURI)	Finds the attribute with the specified local name and namespace URI.
≡Ŵ	Element FindElement(string localName, string namespaceURI)	Finds the element with the specified local name and namespace URI.

14.8.6 Altova.Xml.Meta.Element

This class enables you to access information about classes generated from schema elements. Note that this class is not meant to provide dynamic information about particular instances of an element in an XML document. Instead, it enables you to obtain programmatically information about a particular element defined in the XML schema.

Properties

	Name	Description	
M	ComplexType DataType	Returns the type of the element. Note that this is always a complex type even if declared as simple in the original schema. Use the ContentType property of the returned object to get the simple content type.	
	string LocalName	Returns the local name of the element.	
	int MaxOccurs	Returns the maxOccurs value defined in the schema.	
	int MinOccurs	Returns the minOccurs value defined in the schema.	
	string NamespaceURI	Returns the namespace URI of the element.	
	XmlQualifiedName QualifiedName	Returns the qualified name of the element.	

14.8.7 Altova.Xml.Meta.SimpleType

This class enables you to access schema information about classes generated from simple types. Note that this class is not meant to provide dynamic information about particular instances of simple types in an XML document. Instead, it enables you to obtain programmatically information about a particular simple type defined in the XML schema.

Properties

	Name	Description	
	SimpleType BaseType	Returns the base type of this type.	
	<pre>string[] Enumerations</pre>	Returns a list of all enumeration facets.	
	<pre>int FractionDigits</pre>	Returns the value of this facet.	
	int Length	Returns the value of this facet.	
	string LocalName	Returns the local name of the type.	
	string MaxExclusive	Returns the value of this facet.	
	string MaxInclusive	Returns the value of this facet.	
	int MaxLength	Returns the value of this facet.	
	string MinExclusive	Returns the value of this facet.	
	string MinInclusive	Returns the value of this facet.	
*	int MinLength	Returns the value of this facet.	

	Name	Description	
	string NamespaceURI	Returns the namespace URI of the type.	
	<pre>string[] Patterns</pre>	Returns the pattern facets, or null if no patterns are specified.	
2	XmlQualifiedName QualifiedName	Returns the qualified name of this type.	
2	int TotalDigits	Returns the value of this facet.	
	WhitespaceType Whitespace	Returns the whitespace normalization facet.	

14.8.8 [YourSchema].[Doc]

When code is generated from an XML Schema, the generated code provides a document class with the same name as the schema. This class contains all possible root elements as members, as well as the members listed below. Note that, in the method names below, "Doc" stands for the name of the generated document class itself.

	Name	Description	
=0	<pre>static Doc CreateDocument()</pre>	Creates a new, empty XML document.	
≡∳	static Doc CreateDocument(string encoding)	Creates a new, empty XML document, with encoding of type "encoding".	
- =	<pre>static void DeclareAllNamespacesFromSche ma(Altova.Xml.ElementType node)</pre>	Declares all namespaces from the XML Schema on the eleme supplied as argument (typically, the XML root element). Calling this method is useful if your schema has multiple namespace declarations, each mapped to a prefix, and you would like to declare all of them on the element supplied as argument.	
=	static Doc LoadFromBinary(byte [] binary)	Loads an XML document from a byte array.	
=	static Doc LoadFromFile(string filename)	Loads an XML document from a file.	
.≡∳	static Doc LoadFromString(string xmlstring)	Loads an XML document from a string.	
≡∳	byte [] SaveToBinary(bool prettyPrint)	Saves an XML document to a byte array, with optional "pretty- print" formatting.	
.≡∳	<pre>byte[] SaveToBinary(bool prettyPrint, string encoding)</pre>	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding.	

	Name	Description	
.≡♦	<pre>byte[] SaveToBinary(bool prettyPrint, string encoding, bool bBigEndian, bool bBOM)</pre>	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding, byte order, and BOM (Byte Order Mark).	
=	<pre>void SaveToFile(string fileName, bool prettyPrint)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting.	
#♦	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting. When <code>omitXmlDecl</code> is true, the XML declaration will not be written.	
#♦	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. When <code>omitXmlDecl</code> is true, the XML declaration will not be written.	
₩	<pre>void SaveToFile(string fileName, bool prettyPrint, string encoding, string lineend)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding, and line ending character(s).	
	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, string lineend)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding, and line ending character(s). When omitXmlDecl is true, the XML declaration will not be written.	
	<pre>void SaveToFile(string fileName, bool prettyPrint, bool omitXmlDecl, string encoding, bool bBigEndian, bool bBOM, string lineend)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding, byte order, BOM (Byte Order Mark), and line ending character(s). When <code>omitXmlDec:</code> is true, the XML declaration will not be written.	
щ ф	<pre>void SaveToFileWithLineEnd(string fileName, bool prettyPrint, bool omitXmlDecl, string lineend)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, and line ending character(s). When <code>omitXmlDecl</code> is true, the XML declaration will not be written.	
=	string SaveToString(bool prettyPrint)	Saves an XML document to a file, with optional "pretty-print" formatting.	
≡Ŵ	<pre>string SaveToString(bool prettyPrint, bool omitXmlDecl)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting. When <code>omitXmlDecl</code> is true, the XML declaration will not be written.	
=\$	void SetDTDLocation(string dtdLocation)	Adds a DOCTYPE declaration with the specified system ID. A root element must already exist.	
≡∳	void SetSchemaLocation(string schemaLocation)	Adds an xsi:schemaLocation or xsi:noNamespaceSchemaLocation attribute to the root element. A root element must already exist.	

14.8.9 [YourSchema].[ElementType]

This class provides methods for manipulating XML elements from your schema. Methods of this class can be called on elements, not on the XML document itself. Note that, in order to call methods of this class, you don't need to instantiate the class directly. Any element created using the **Append()** or **AppendWithPrefix()** methods is of [ElementType] type.

	Name	Descript	tion
=∳	void DeclareNamespace(string prefix, string nsURI)	This met prefix an supplied value sup empty, th declarati For exan element	hod takes two arguments that are both of string type: the ad the namespace URI that you want to use. The prefix as argument will be mapped to the namespace URI pplied as argument. If the prefix supplied as argument is he method creates or overrides the default namespace on in the element. nple, let's assume that the XML document has an XML called "purchase". If you call
		purcha "http:	<pre>se.DeclareNamespace("ord", //OrderTypes");</pre>
		then the	XML document becomes
		<purch< th=""><th><pre>mase xmlns:ord="http://OrderTypes" /></pre></th></purch<>	<pre>mase xmlns:ord="http://OrderTypes" /></pre>
		Another	example, if you call:
		purcha "http:	<pre>se.DeclareNamespace("", //OrderTypes");</pre>
		then the	XML document becomes
		<purch< th=""><th><pre>mase xmlns="http://OrderTypes" /></pre></th></purch<>	<pre>mase xmlns="http://OrderTypes" /></pre>
		Note:	The declared namespace is used when appending subsequent child elements or attributes, according to the following rules:
		1.	If the child namespace is the default, then use empty
		2.	If the child namespace is equal to the parent one, then use the parent prefix
		3.	Otherwise, search for nearest prefix from parent to top, using the lookup algorithm described in section "B.2: Namespace Prefix Lookup" at https://www.w3.org/TR/2002/WD-DOM-Level-3-Core- 20021022/namespaces-algorithms.html.

Name	Descri	Description	
	4.	If there is no prefix for element namespace found, then use empty prefix.	

14.8.10 [YourSchemaType].MemberAttribute

When code is generated from an XML schema, a class is created for each member attribute of a type. In the descriptions below, "AttributeType" stands for the type of the member attribute itself.

Methods

	Name	Description
=0	bool Exists()	Returns true if the attribute exists.
=0	void Remove()	Removes the attribute from its parent element.

Properties

	Name	Description
2	int EnumerationValue	Generated for enumeration types only. Sets or gets the attribute value using one of the constants generated for the possible values. Returns Invalid if the value does not match any of the enumerated values in the schema.
2	Altova.Xml.Meta.Attribute Info	Returns an object for querying schema information (see <u>Altova.Xml.Meta.Attribute</u>).
*	AttributeType Value	Sets or gets the attribute value.

14.8.11 [YourSchemaType].MemberElement

When code is generated from an XML schema, a class with the following members is created for each member element of a type. The class implements the standard System.Collections.IEnumerable interface, so it can be used with the foreach statement.

In the descriptions below, "MemberType" stands for the type of the member element itself.

	Name	Description
=0	MemberType Append()	Creates a new element and appends it to its parent.
=\$	MemberType AppendWithPrefix(string prefix)	Creates a new element having the prefix supplied as argument, and appends it to its parent. For an example, see Example: Purchase Order.

	Name	Description
=0	MemberType At(int index)	Returns the member element specified by the index.
=0	System.Collections.IEnumerator GetEnumerator()	Returns an object for iterating instances of the member element.
=0	void Remove()	Deletes all occurrences of the element from its parent.
=0	void RemoveAt(int index)	Deletes the occurrence of the element specified by the index.

Properties

	Name	Description
2	int Count	Returns the count of elements.
*	int EnumerationValue	Generated for enumeration types only. Sets or gets the element value using one of the constants generated for the possible values. Returns Invalid if the value does not match any of the enumerated values in the schema.
2	bool Exists	Returns true if at least one element exists.
2	MemberType First	Returns the first instance of the member element.
*	Altova.Xml.Meta.Element Info	Returns an object for querying schema information (see <u>Altova.Xml.Meta.Element</u>).
2	MemberType Last	Returns the last instance of the member element.
2	MemberType this [int index]	Returns the member element specified by the index.
2	MemberType Value	Sets or gets the element content (only generated if element can have mixed or simple content).

14.9 Reference to Generated Classes (Java)

This chapter includes a description of Java classes generated with MapForce from a DTD or XML schema (see <u>Generating Code from XML Schemas or DTDs</u>). You can integrate these classes into your code to read, modify, and write XML documents.

Note: The generated code does include other supporting classes, which are not listed here and are subject to modification.

14.9.1 com.altova.types.DateTime

This class enables you to process XML attributes or elements that have date and time types, such as xs:dateTime.

Cor	ารtrเ	Jcto	rs

	Name	Description	
• •	<pre>public DateTime()</pre>	Initializes a new instance of the DateTime class to an empty value.	
• •	public DateTime(DateTime newvalue)	Initializes a new instance of the DateTime class to the DateTime value supplied as argument.	
• ^c	<pre>public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, int newoffsetTZ)</pre>	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, second, the fractional part of the second, and timezone supplied as arguments. The fractional part of the second newpartsecond must be between 0 and 1. The timezone offset newoffsetTZ can be either positive or negative and is expressed in minutes.	
• ^c	<pre>public DateTime(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond)</pre>	Initializes a new instance of the DateTime class to the year, month, day, hour, minute, second, and the fractional part of a second supplied as arguments.	
• •	<pre>public DateTime(int newyear, int newmonth, int newday)</pre>	Initializes a new instance of the DateTime class to the year, month, and day supplied as arguments.	
• •	public DateTime(Calendar newvalue)	Initializes a new instance of the DateTime class to the java.util.Calendar value supplied as argument.	

	Name	Description
● ^S	<pre>static DateTime now()</pre>	Returns the current time as a DateTime object.
• ^{\$}	<pre>static DateTime parse(String s)</pre>	Returns a DateTime object parsed from the string value supplied as argument. For example, the following sample string values

	Name	Description	
		would be converted successfully to a DateTime object:	
		2015-11-24T12:54:47.969+01:00 2015-11-24T12:54:47 2015-11-24	
۲	<pre>int getDay()</pre>	Returns the day of the current DateTime instance.	
۲	<pre>int getHour()</pre>	Returns the hour of the current DateTime instance.	
۲	<pre>int getMillisecond()</pre>	Returns the millisecond of the current DateTime instance, as an integer value.	
۲	<pre>int getMinute()</pre>	Returns the minute of the current DateTime instance.	
۲	<pre>int getMonth()</pre>	Returns the month of the current DateTime instance.	
•	double getPartSecond()	Returns the fractional part of the second of the current DateTime instance, as a double value. The return value is greater than zero and smaller than one, for example:	
۲	<pre>int getSecond()</pre>	Returns the second of the current DateTime instance.	
•	<pre>int getTimezoneOffset()</pre>	Returns the timezone offset, in minutes, of the current DateTime instance. For example, the timezone "UTC-01:00" would be returned as:	
•	Calendar getValue()	Returns the current DateTime instance as a	
		java.util.Calendar Value .	
•	<pre>int getWeekday()</pre>	Returns the day in week of the current DateTime instance. Values range from 0 through 6, where 0 is Monday (ISO-8601).	
۲	<pre>int getYear()</pre>	Returns the year of the current DateTime instance.	
۰	int hasTimezone()	Returns information about the timezone of the current DateTime instance. Possible return values are:CalendarBase.TZ_MISSINA timezone offset is not defined.GCalendarBase.TZ_UTCThe timezone is UTC.CalendarBase.TZ_OFFSETA timezone offset has been defined.	
۲	void setDay(int nDay)	Sets the day of the current DateTime instance to the value supplied as argument.	
۲	void setHasTimezone(int nHasTZ)	Sets the timezone information of the current DateTime instance to the value supplied as argument. This method can be used to	

	Name	Description	
		strip the timezone information or set the timezone to UTC (Coordinated Universal Time). Valid values for the nHasTZ argument:	
		CalendarBase.TZ_MISSIN Set the timezone offset to undefined.	
		CalendarBase.TZ_UTC Set the timezone to UTC.	
		CalendarBase.TZ_OFFSET If the current object has a timezone offset, leave it unchanged.	
۲	<pre>void setHour(int nHour)</pre>	Sets the hour of the current DateTime instance to the value supplied as argument.	
۲	void setMinute(int nMinute)	Sets the minute of the current DateTime instance to the value supplied as argument.	
۲	<pre>void setMonth(int nMonth)</pre>	Sets the month of the current DateTime instance to the value supplied as argument.	
۲	<pre>void setPartSecond(double nPartSecond)</pre>	Sets the fractional part of the second of the current DateTime instance to the value supplied as argument.	
۲	<pre>void setSecond(int nSecond)</pre>	Sets the second of the current DateTime instance to the value supplied as argument.	
۲	<pre>void setTimezoneOffset(int nOffsetTZ)</pre>	Sets the timezone offset of the current DateTime instance to the value supplied as argument. The value nOffsetTZ must be an integer (positive or negative) and must be expressed in minutes.	
۲	<pre>void setYear(int nYear)</pre>	Sets the year of the current DateTime instance to the value supplied as argument.	
۲	String toString()	Returns the string representation of the current DateTime instance, for example:	
		2015-11-24T15:50:56.968+01:00	

Examples

Before using the following code listings in your program, ensure the Altova types are imported:

```
import com.altova.types.*;
```

The following code listing illustrates various ways to create ${\tt DateTime}$ objects:

```
protected static void DateTimeExample1()
{
    // Initialize a new instance of the DateTime class to the current time
    DateTime dt = new DateTime(DateTime.now());
```

```
System.out.println("DateTime created from current date and time: " + dt.toString());
// Initialize a new instance of the DateTime class by supplying the parts
DateTime dt1 = new DateTime(2015, 11, 23, 14, 30, 24, .459);
System.out.println("DateTime from parts (no timezone): " + dt1.toString());
// Initialize a new instance of the DateTime class by supplying the parts
DateTime dt2 = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
System.out.println("DateTime from parts (with negative timezone): " +
dt2.toString());
// Initialize a new instance of the DateTime class by parsing a string value
DateTime dt3 = DateTime.parse("2015-11-24T12:54:47.969+01:00");
System.out.println("DateTime parsed from string: " + dt3.toString());
```

The following code listing illustrates getting values from DateTime objects:

```
protected static void DateTimeExample2()
      // Initialize a new instance of the DateTime class to the current time
      DateTime dt = new DateTime(DateTime.now());
      // Output the formatted year, month, and day of this DateTime instance
      String str1 = String.format("Year: %d; Month: %d; Day: %d;", dt.getYear(),
dt.getMonth(), dt.getDay());
      System.out.println(str1);
      // Output the formatted hour, minute, and second of this DateTime instance
      String str2 = String.format("Hour: %d; Minute: %d; Second: %d;", dt.getHour(),
dt.getMinute(), dt.getSecond());
      System.out.println(str2);
      // Return the timezone (in minutes) of this DateTime instance
      System.out.println("Timezone: " + dt.getTimezoneOffset());
      // Get the DateTime as a java.util.Calendar value
      java.util.Calendar dt_java = dt.getValue();
      System.out.println("" + dt java.toString());
      // Return the day of week of this DateTime instance
      System.out.println("Weekday: " + dt.getWeekday());
      // Check whether the DateTime instance has a timezone defined
      switch(dt.hasTimezone())
         case CalendarBase.TZ MISSING:
            System.out.println("No timezone.");
            break:
         case CalendarBase.TZ UTC:
            System.out.println("The timezone is UTC.");
            break:
         case CalendarBase.TZ OFFSET:
            System.out.println("This object has a timezone.");
            break:
         default:
            System.out.println("Unable to determine whether a timezone is defined.");
            break;
```

}

The following code listing illustrates changing the timezone offset of a DateTime object:

```
protected static void DateTimeExample3()
{
    // Create a new DateTime object with timezone -0100 UTC
    DateTime dt = new DateTime(2015, 11, 24, 14, 30, 24, .459, -60);
    // Output the value before the change
    System.out.println("Before: " + dt.toString());
    // Change the offset to +0100 UTC
    dt.setTimezoneOffset(60);
    // Output the value after the change
    System.out.println("After: " + dt.toString());
}
```

14.9.2 com.altova.types.Duration

This class enables you to process XML attributes or elements of type xs:duration.

Constructors

	Name	Description
• °	Duration(Duration newvalue)	Initializes a new instance of the Duration class to the Duration object supplied as argument.
• ^c	Duration(int newyear, int newmonth, int newday, int newhour, int newminute, int newsecond, double newpartsecond, boolean newisnegative)	Initializes a new instance of the Duration class to a duration built from parts supplied as arguments.

	Name	Description
• ^{\$}	<pre>static Duration getFromDayTime(int newday, int newhour, int newminute, int newsecond, double newpartsecond)</pre>	Returns a Duration object created from the number of days, hours, minutes, seconds, and fractional second parts supplied as argument.
• ^{\$}	static Duration getFromYearMonth(int newyear, int newmonth)	Returns a Duration object created from the number of years and months supplied as argument.
• ^{\$}	<pre>static Duration parse(String s)</pre>	Returns a Duration object created from the string supplied as argument. For example, the string -P1Y1M1DT1H1M1.333S can be used to create a negative duration of one year, one month,

	Name	Description	
		one day, one hour, one minute, one second, and 0.333 fractional parts of a second. To create a negative duration, append the minus sign (-) to the string.	
• ^{\$}	static Duration parse(String s, ParseType pt)	Returns a Duration object created from the string supplied as argument, using a specific parse format. The parse format can be any of the following:	
		ParseType.DAYTIME	May be used when the string s consists of any of the following: days, hours, minutes, seconds, fractional second parts, for example -P4DT4H4M4.774S.
		ParseType.DURATION	May be used when the string s consists of any of the following: years, months, days, hours, minutes, seconds, fractional second parts, for example P1Y1M1DT1H1M1.333S.
		ParseType.YEARMON TH	May be used when the string s consists of any of the following: years, months. For example: P3Y2M.
۲	<pre>int getDay()</pre>	Returns the number of days in the current Duration instance.	
۰	<pre>long getDayTimeValue()</pre>	Returns the day and time value (in milliseconds) of the current Duration instance. Years and months are ignored.	
۲	<pre>int getHour()</pre>	Returns the number of hours in the current Duration instance.	
۲	<pre>int getMillisecond()</pre>	Returns the number of milliseconds in the current Duration instance.	
۲	<pre>int getMinute()</pre>	Returns the number of minutes in the current Duration instance.	
۲	<pre>int getMonth()</pre>	Returns the number of months in the current Duration instance.	
۲	<pre>double getPartSecond()</pre>	Returns the number of fractional second parts in the current Duration instance.	
۲	<pre>int getSecond()</pre>	Returns the number of seconds in the current Duration instance.	
۲	<pre>int getYear()</pre>	Returns the number of years in the current Duration instance.	
۲	<pre>int getYearMonthValue()</pre>	Returns the year and month value (in months) of the current Duration instance. Days, hours, seconds, and milliseconds are ignored.	

	Name	Description	
۲	boolean isNegative()	Returns Boolean true if the current Duration instance is negative.	
۲	<pre>void setDayTimeValue(long 1)</pre>	Sets the duration to the number of milliseconds supplied as argument, affecting only the day and time part of the duration.	
۰	void setNegative(boolean isnegative)	Converts the current Duration instance to a negative duration.	
۲	<pre>void setYearMonthValue(int 1)</pre>	Sets the duration to the number of months supplied as argument. Only the years and months part of the duration is affected.	
٩	String toString()	Returns the string representation of the current Duration instance, for example:	
۲	<pre>String toYearMonthString()</pre>	Returns the string representation of the YearMonth part of the current Duration instance, for example:	

Examples

Before using the following code listings in your program, ensure the Altova types are imported:

```
import com.altova.types.*;
import com.altova.types.Duration.ParseType;
```

The following code listing illustrates various ways to create Duration objects:

```
protected static void ExampleDuration()
{
   // Create a negative duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1
second,
   // and 0.333 fractional second parts
   Duration dr = new Duration(1, 1, 1, 1, 1, 1, .333, true);
   // Create a duration from an existing Duration object
   Duration dr1 = new Duration(dr);
  // Create a duration of 4 days, 4 hours, 4 minutes, 4 seconds, .774 fractional
second parts
   Duration dr2 = Duration.getFromDayTime(4, 4, 4, 4, .774);
   // Create a duration of 3 years and 2 months
   Duration dr3 = Duration.getFromYearMonth(3, 2);
   // Create a duration from a string
   Duration dr4 = Duration.parse("-P4DT4H4M4.774S");
   // Create a duration from a string, using specific parse formats
   Duration dr5 = Duration.parse("-P1Y1M1DT1H1M1.333S", ParseType.DURATION);
```

}

```
Duration dr6 = Duration.parse("P3Y2M", ParseType.YEARMONTH);
Duration dr7 = Duration.parse("-P4DT4H4M4.774S", ParseType.DAYTIME);
```

The following code listing illustrates getting and setting the value of Duration objects:

```
protected static void DurationExample2()
   // Create a duration of 1 year, 2 month, 3 days, 4 hours, 5 minutes, 6 seconds,
      // and 333 milliseconds
   Duration dr = new Duration(1, 2, 3, 4, 5, 6, .333, false);
   // Output the number of days in this duration
   System.out.println(dr.getDay());
   // Create a positive duration of one year and 333 milliseconds
   Duration dr1 = new Duration(1, 0, 0, 0, 0, 0, .333, false);
   // Output the day and time value in milliseconds
   System.out.println(dr1.getDayTimeValue());
   // Create a positive duration of 1 year, 1 month, 1 day, 1 hour, 1 minute, 1
second,
       // and 333 milliseconds
   Duration dr2 = new Duration(1, 1, 1, 1, 1, 1, .333, false);
   // Output the year and month value in months
   System.out.println(dr2.getYearMonthValue());
   // Create a positive duration of 1 year and 1 month
   Duration dr3 = new Duration(1, 1, 0, 0, 0, 0, 0, false);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Set the DayTime part of duration to 1000 milliseconds
   dr3.setDayTimeValue(1000);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Set the YearMonth part of duration to 1 month
   dr3.setYearMonthValue(1);
   // Output the value
   System.out.println("The duration is now: " + dr3.toString());
   // Output the year and month part of the duration
   System.out.println("The YearMonth part of the duration is: " +
dr3.toYearMonthString());
}
```

14.9.3 com.altova.xml.meta.Attribute

This class enables you to access schema information about classes generated from attributes. Note that this class is not meant to provide dynamic information about particular instances of an attribute in an XML document. Instead, it enables you to obtain programmatically information about a particular attribute defined in the XML schema.

Methods

	Name	Description
۲	SimpleType getDataType()	Returns the type of the attribute content.
۲	String getLocalName()	Returns the local name of the attribute.
۲	String getNamespaceURI()	Returns the namespace URI of the attribute.
۲	boolean isRequired()	Returns true if the attribute is required.

14.9.4 com.altova.xml.meta.ComplexType

This class enables you to access schema information about classes generated from complex types. Note that this class is not meant to provide dynamic information about particular instances of a complex type in an XML document. Instead, it enables you to obtain programmatically information about a particular complex type defined in the XML schema.

Methods

	Name	Description
۲	Attribute findAttribute(String localName, String namespaceURI)	Finds the attribute with the specified local name and namespace URI.
۲	Element findElement(String localName, String namespaceURI)	Finds the element with the specified local name and namespace URI.
۲	<pre>Attribute[] GetAttributes()</pre>	Returns a list of all attributes.
۲	ComplexType getBaseType()	Returns the base type of this type.
۲	SimpleType getContentType()	Returns the simple type of the content.
۲	<pre>Element[] GetElements()</pre>	Returns a list of all elements.
۲	String getLocalName()	Returns the local name of the type.
۲	String getNamespaceURI()	Returns the namespace URI of the type.

14.9.5 com.altova.xml.meta.Element

This class enables you to access information about classes generated from schema elements. Note that this class is not meant to provide dynamic information about particular instances of an element in an XML document. Instead, it enables you to obtain programmatically information about a particular element defined in the XML schema.

Methods

	Name	Description
•	ComplexType getDataType()	Returns the type of the element. Note that this is always a complex type even if declared as simple in the original schema. Use getContentType() of the returned object to get the simple content type.
۲	String getLocalName()	Returns the local name of the element.
•	<pre>int getMaxOccurs()</pre>	Returns the maxOccurs value defined in the schema.
۲	<pre>int getMinOccurs()</pre>	Returns the minOccurs value defined in the schema.
۲	String getNamespaceURI()	Returns the namespace URI of the element.

14.9.6 com.altova.xml.meta.SimpleType

This class enables you to access schema information about classes generated from simple types. Note that this class is not meant to provide dynamic information about particular instances of simple types in an XML document. Instead, it enables you to obtain programmatically information about a particular simple type defined in the XML schema.

	Name	Description
۲	SimpleType getBaseType()	Returns the base type of this type.
۲	<pre>String[] getEnumerations()</pre>	Returns an array of all enumeration facets.
۲	<pre>int getFractionDigits()</pre>	Returns the value of this facet.
۲	<pre>int getLength()</pre>	Returns the value of this facet.
۲	String getLocalName()	Returns the local name of the type.
۲	<pre>String getMaxExclusive()</pre>	Returns the value of this facet.
۲	<pre>String getMaxInclusive()</pre>	Returns the value of this facet.
۲	<pre>int getMaxLength()</pre>	Returns the value of this facet.
۲	<pre>String getMinExclusive()</pre>	Returns the value of this facet.
۲	<pre>String getMinInclusive()</pre>	Returns the value of this facet.
۲	<pre>int getMinLength()</pre>	Returns the value of this facet.
۲	String getNamespaceURI()	Returns the namespace URI of the type.
۲	<pre>String[] getPatterns()</pre>	Returns an array of all pattern facets.

	Name	Description
۲	<pre>int getTotalDigits()</pre>	Returns the value of this facet.
•	<pre>int getWhitespace()</pre>	Returns the value of the whitespace facet, which is one of: com.altova.typeinfo.WhitespaceType.Whitespace_Unknow n com.altova.typeinfo.WhitespaceType.Whitespace_Preser ve com.altova.typeinfo.WhitespaceType.Whitespace_Replac e com.altova.typeinfo.WhitespaceType.Whitespace_Collap se

14.9.7 com.[YourSchema].[Doc]

When code is generated from an XML Schema, the generated code provides a document class with the same name as the schema. This class contains all possible root elements as members, as well as the members listed below. Note that, in the method names below, "Doc" stands for the name of the generated document class itself.

	Name	Description
● ^S	<pre>static Doc createDocument()</pre>	Creates a new, empty XML document.
• ^{\$}	static void declareAllNamespacesFromSche ma(com.altova.xml.ElementTyp e node)	Declares all namespaces from the XML Schema on the element supplied as argument (typically, the XML root element). Calling this method is useful if your schema has multiple namespace declarations, each mapped to a prefix, and you would like to declare all of them on the element supplied as argument.
• ^{\$}	static Doc loadFromBinary(byte[] xml)	Loads an XML document from a byte array.
• ^{\$}	static Doc loadFromFile(String fileName)	Loads an XML document from a file.
• ^{\$}	static Doc loadFromString(String xml)	Loads an XML document from a string.
۰	byte[] saveToBinary(boolean prettyPrint)	Saves an XML document to a byte array, with optional "pretty- print" formatting.
۲	<pre>byte[] saveToBinary(boolean prettyPrint, String encoding)</pre>	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding.
•	byte[] saveToBinary(boolean prettyPrint, String encoding, boolean bigEndian, boolean writeBOM)	Saves an XML document to a byte array, with optional "pretty- print" formatting, with the specified encoding. Byte order and Unicode byte-order mark can be specified for Unicode encodings.

	Name	Description
۲	void saveToFile(String fileName, boolean prettyPrint)	Saves an XML document to a file, with optional "pretty-print" formatting.
۲	<pre>void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with UTF-8 encoding. When omitXmlDecl is true, the XML declaration will not be written.
۰	<pre>void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding)</pre>	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. When <code>omitXmlDecl</code> is true, the XML declaration will not be written.
•	void saveToFile(String fileName, boolean prettyPrint, boolean omitXmlDecl, String encoding, boolean bBigEndian, boolean bBOM)	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. When <code>omitXmlDecl</code> is true, the XML declaration will not be written. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
۲	void saveToFile(String fileName, boolean prettyPrint, String encoding)	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding.
•	void saveToFile(String fileName, boolean prettyPrint, String encoding, boolean bBigEndian, boolean bBOM)	Saves an XML document to a file, with optional "pretty-print" formatting, with the specified encoding. Byte order and Unicode byte-order mark can be specified for Unicode encodings.
۲	String saveToString(boolean prettyPrint)	Saves an XML document to a string, with optional "pretty-print" formatting.
۲	String saveToString(boolean prettyPrint, boolean omitXmlDecl)	Saves an XML document to a string, with optional "pretty-print" formatting. When <code>omitXmlDecl</code> is true, the XML declaration will not be written.
۲	void setSchemaLocation(String schemaLocation)	Adds an xsi:schemaLocation or xsi:noNamespaceSchemaLocation attribute to the root element. A root element must already exist.

14.9.8 com.[YourSchema].[ElementType]

This class provides methods for manipulating XML elements from your schema. Methods of this class can be called on elements, not on the XML document itself. Note that, in order to call methods of this class, you don't need to instantiate the class directly. Any element created using the append() or appendWithPrefix() methods is of [ElementType] type.

	Name	Description
void declareNamespace(String prefix, String nsURI)	This method takes two arguments that are both of string type: the prefix and the namespace URI that you want to use. The prefix supplied as argument will be mapped to the namespace URI value supplied as argument. If the prefix supplied as argument is empty, the method creates or overrides the default namespace declaration in the element. For example, let's assume that the XML document has an XML element called "purchase". If you call	
		<pre>purchase.declareNamespace("ord", "http://OrderTypes");</pre>
		then the XML document becomes
		<purchase xmlns:ord="http://OrderTypes"></purchase>
		Another example, if you call:
		<pre>purchase.declareNamespace("", "http://OrderTypes");</pre>
		then the XML document becomes
		<pre><purchase xmlns="http://OrderTypes"></purchase></pre>
		Note: The declared namespace is used when appending subsequent child elements or attributes, according to the following rules:
		 If the child namespace is the default, then use empty prefix. If the child namespace is equal to the parent one, then use the parent prefix. Otherwise, search for nearest prefix from parent to top, using the lookup algorithm described in section "B.2: Namespace Prefix Lookup" at <u>https://www.w3.org/TR/2002/WD-DOM-Level-3- Core-20021022/namespaces-algorithms.html.</u> If there is no prefix for element namespace found
		then use empty prefix.

14.9.9 com.[YourSchema].[YourSchemaType].MemberAttribute

When code is generated from an XML schema, a class is created for each member attribute of a type. In the descriptions below, "AttributeType" stands for the type of the member attribute itself.

Methods

	Name	Description
۲	boolean exists()	Returns true if the attribute exists.
۲	<pre>int getEnumerationValue()</pre>	Generated for enumeration types only. Returns one of the constants generated for the possible values, or Invalid if the value does not match any of the enumerated values in the schema.
۲	com.altova.xml.meta.Attribute getInfo()	Returns an object for querying schema information (see <pre>com.altova.xml.meta.Attribute</pre>).
۲	AttributeType getValue()	Gets the attribute value.
۲	void remove()	Removes the attribute from its parent element.
۲	void setEnumerationValue(int)	Generated for enumeration types only. Pass one of the constants generated for the possible values to this method to set the value.
۲	void setValue(AttributeType value)	Sets the attribute value.

14.9.10 com.[YourSchema].[YourSchemaType].MemberElement

When code is generated from an XML schema, a class with the following members is created for each member element of a type. In the descriptions below, "MemberType" stands for the type of the member element itself.

	Name	Description
۲	MemberType append()	Creates a new element and appends it to its parent.
٩	MemberType appendWithPrefix(String prefix)	Creates a new element having the prefix supplied as argument, and appends it to its parent. For an example, see Example: Purchase Order.
۲	MemberType at(int index)	Returns the instance of the member element at the specified index.
۲	<pre>int count()</pre>	Returns the count of elements.

	Name	Description
۲	<pre>boolean exists()</pre>	Returns true if at least one element exists.
۲	MemberType first()	Returns the first instance of the member element.
۰	<pre>int getEnumerationValue()</pre>	Generated for enumeration types only. Returns one of the constants generated for the possible values, or Invalid if the value does not match any of the enumerated values in the schema.
۲	com.altova.xml.meta.Element getInfo()	Returns an object for querying schema information (see <pre>com.altova.xml.meta.Element</pre>).
۲	MemberType getValue()	Gets the element content (only generated if element can have simple or mixed content).
۲	java.util.Iterator iterator()	Returns an object for iterating instances of the member element.
۲	MemberType last()	Returns the last instance of the member element.
۲	<pre>void remove()</pre>	Deletes all occurrences of the element from its parent.
۲	void removeAt(int index)	Deletes the occurrence of the element specified by the index.
۲	void setEnumerationValue(int index)	Generated for enumeration types only. Pass one of the constants generated for the possible values to this method to set the value.
۲	void setValue(MemberType value)	Sets the element content (only generated if element can have simple or mixed content).

14.10 Code Generation Tips

Resolving "Out of memory" exceptions during Java compilation

Complex mappings with large schemas can produce a large amount of code, which might cause a java.lang.OutofMemory exception during compilation using Ant. To rectify this:

• Add the environment variable ANT_OPTS, which sets specific Ant options such as the memory to be allocated to the compiler, and set its value as shown below.

Edit User Variable	•••
Variable <u>n</u> ame:	ANT_OPTS
Variable <u>v</u> alue:	-server -Xmx512m -Xms512m
	OK Cancel

• To make sure that the compiler and the generated code run in the same process as Ant, change the fork attribute, in **build.xml**, to false.

You may need to customize the values depending on the amount of memory in your machine and the size of the project you are working with. For more details, see your Java VM documentation.

```
When running the ant jar command, you may get an error message similar to "[...] archive contains more than 65535 entities". To prevent this, it is recommended that you use Ant 1.9 or later, and, in the build.xml file, add zip64mode="as-needed" to the <jar> element.
```

Reserving method names

When customizing code generation using the supplied SPL files, it might be necessary to reserve names to avoid collisions with other symbols. To do this:

- 1. Navigate to the program installation directory, for example, C:\Program Files\Altova\MapForce2021.
- 2. In the **spl** subdirectory, locate the directory corresponding to the programming language, for example, **..\spl\java**.
- 3. Open the **settings.spl** file and insert a new line into the "reserve" section, for example, **reserve** "myReservedWord".
- 4. Regenerate the program code.

14.11 Code Generator Options

To view or change the MapForce settings applicable to code generation:

• On the Tools menu, click **Options**, and then click **Generation**.

C <u>+</u> + Settings Project File:	C <u>#</u> Settings Project File:	
Microsoft® Visual Studio® 2019 XML Library MSXML 6	Microsoft® Visual Studio® 2019	~
Library Type	Wrapper Classes	
Server Execution File Convert all ADO and ODBC Database Conn Ignore Digital Signatures (unsupported by Ma	ections to <u>J</u> DBC apForce Server)	
Generate for MapForce Server version:	most current ~	

The available settings are as follows.

C++ Settings	 Defines the specific compiler settings for the C++ environment, namely: The Visual Studio version (2010, 2013, 2015, 2017, 2019) The XML library (MSXML, Xerces 3.x) Whether static or dynamic libraries must be generated Whether code must be generated with or without MFC support
C# Settings	Select the option Microsoft .NET Core 3.1 or Microsoft .NET 5.0 to generate a Visual Studio solution targeting the respective platforms.

	If you need to target the .NET Framework platform for a specific Visual Studio version, select any of the Microsoft Visual Studio 2010-2019 options —in this case, the generated solution will target the .NET Framework version corresponding to the respective Visual Studio version.
Wrapper Classes	Allows you to generate wrapper classes for XML schemas, see <u>Generating</u> <u>Code from XML Schemas or DTDs</u> . These wrapper classes can be used by custom code that includes the code generated by MapForce.
Server Execution File	These options are applicable when you compile mappings to MapForce Server execution files. They do not affect generation of C#, C++, or Java code. For more information, see <u>Compiling Mappings to MapForce Server</u> <u>Execution Files</u> .

14.12 SPL Reference

This section gives an overview of SPL (Spy Programming Language), the code generator's template language.

It is assumed that you have prior programming experience, and are familiar with operators, functions, variables and classes, as well as the basics of object-oriented programming - which is used heavily in SPL.

The templates used by MapForce are supplied in the ...\MapForce\spl folder. You can use these files as an aid to help you in developing your own templates.

How code generator works

Inputs to the code generator are the template files (.spl) and the object model provided by MapForce. The template files contain SPL instructions for creating files, reading information from the object model and performing calculations, interspersed with literal code fragments in the target programming language.

The template file is interpreted by the code generator and outputs **.cpp**, **.java**, **.cs** source code files, project files, or any other type of file depending on the template. The source code can then be compiled into an executable file that accesses XML data described by the schema file.

SPL files have access to a wide variety of information that is collated from the source schemas. Note that an SPL file is not tied to a specific schema, but allows access to all schemas. Make sure you write your SPL files generically and avoid structures which apply to specific schemas.

Example: Creating a new file in SPL

This is a very basic SPL file. It creates a file named **test.cpp**, and places the include statement within it. The close command completes the template.

```
[create "test.cpp"]
#include "stdafx.h"
[close]
```

14.12.1 Basic SPL structure

An SPL file contains literal text to output, interspersed with code generator instructions.

Code generator instructions are enclosed in square brackets '[' and ']'. Multiple statements can be included in a bracket pair. Additional statements have to be separated by a new line or a colon ':'.

Valid examples are:

```
[\$x = 42
\$x = \$x + 1]
```

or

[\$x = 42: \$x = \$x + 1]

Adding text to files

Text not enclosed by [and], is written directly to the current output file. If there is no current output file, the text is ignored (see <u>Using files</u> how to create an output file).

To output literal square brackets, escape them with a backslash: \[and \]; to output a backslash use \\.

Comments

Comments inside an instruction block always begin with a ' character, and terminate on the next line, or at a block close character].

14.12.2 Declarations

The following statements are evaluated while parsing the SPL template file. They are **not** affected by flow control statements like conditions, loops or subroutines, and are always evaluated exactly once.

These keywords, like all keywords in SPL, are not case sensitive.

Remember that all of these declarations must be inside a block delimited by square brackets.

map ... to ...

map mapname key to value [, key to value]...

This statement adds information to a map. See below for specific uses.

map schemanativetype schematype to typespec

The specified built-in XML Schema type will be mapped to the specified native type or class, using the specified formatter. This setting applies only to code generation for version 2007r3 and higher. Typespec is a native type or class name, followed by a comma, followed by the formatter class instance.

Example:

map schemanativetype "double" to "double,Altova::DoubleFormatter"

map type ... to ...

map type schematype to classname

The specified built-in XML Schema type will be mapped to the specified class. This setting applies only to code generation for version 2007 or lower.

Example:

```
map type "float" to "CSchemaFloat"
```

default ... is ...

default setting is value

This statement allows you to affect how class and member names are derived from the XML Schema. Note that the setting names are case sensitive.

Example:

default "InvalidCharReplacement" is "_"

Setting name	Explanation
ValidFirstCharSet	Allowed characters for starting an identifier
ValidCharSet	Allowed characters for other characters in an identifier
InvalidCharReplacement	The character that will replace all characters in names that are not in the ValidCharSet
AnonTypePrefix	Prefix for names of anonymous types*
AnonTypeSuffix	Suffix for names of anonymous types*
ClassNamePrefix	Prefix for generated class names
ClassNameSuffix	Suffix for generated class names
EnumerationPrefix	Prefix for symbolic constants declared for enumeration values
EnumerationUpperCase	"on" to convert the enumeration constant names to upper case
FallbackName	If a name consists only of characters that are not in ValidCharSet, use this one

* Names of anonymous types are built from AnonTypePrefix + element name + AnonTypeSuffix

reserve

reserve word

Adds the specified word to the list of reserved words. This ensures that it will never be generated as a class or member name.
Example:

reserve "while"

include

includes the specified file as SPL source. This allows you to split your template into multiple files for easier editing and handling.

include filename

Example:

include "Module.cpp"

14.12.3 Variables

Any non-trivial SPL file will require variables. Some variables are <u>predefined</u> by the code generator, and new variables may be created simply by assigning values to them.

The **\$** character is used when **declaring** or **using** a variable, a variable name is always prefixed by **\$**. Variable names are **case sensitive**.

Variables types:

- integer also used as boolean, where 0 is false and everything else is true
- string
- object provided by MapForce
- iterator see <u>foreach</u> statement

Variable types are declared by first assignment:

[\$x = 0]

x is now an integer.

```
[$x = "teststring"]
```

x is now treated as a string.

Strings

String constants are always enclosed in double quotes, like in the example above. \n and \t inside double quotes are interpreted as newline and tab, \" is a literal double quote, and \\ is a backslash. String constants can also span multiple lines.

String concatenation uses the & character:

```
[$BasePath = $outputpath & "/" & $JavaPackageDir]
```

Objects

Objects represent the information contained in the XML schemas, database structures, text files and mappings. Objects have **properties**, which can be accessed using the . operator. It is not possible to create new objects in SPL (they are predefined by the code generator, derived from the input mapping), but it is possible to assign objects to variables.

Example:

```
class [=$class.Name]
```

This example outputs the word "class", followed by a space and the value of the **Name** property of the **\$class** object.

14.12.4 Predefined variables

After a Schema file is analyzed by the code generator, the objects in the table below exist in the Template Engine.

Name	Туре	Description
\$schematype	integer	1 for DTD, 2 for XML Schema
\$TheLibrary	<u>Library</u>	The library derived from the XML Schema or DTD
\$module	string	Name of the source Schema without extension
\$outputpath	string	The output path specified by the user, or the default output path

For C++ generation only:

Name	Туре	Descriptio	on
\$domtype	integer	1 for MSXML, 2 for Xerces	
\$libtype	integer	1 for static LIB, 2 for DLL	
\$mfc	boolean	True if MFC support is enabled	
\$VSVersion	integer	Specifies the Visual Studio version. Valid values:	
		0	No Visual Studio project
		2010	Visual Studio 2010
		2013	Visual Studio 2013
		2015	Visual Studio 2015
1			

Name	Туре	Description	
		2017	Visual Studio 2017
		2019	Visual Studio 2019

For C# generation only:

Name	Туре	Descriptio	on
\$VSVersion	integer	Specifies ⁻	the Visual Studio version. Valid values:
		0	No Visual Studio project
		2010	Visual Studio 2010
		2013	Visual Studio 2013
		2015	Visual Studio 2015
		2017	Visual Studio 2017
		2019	Visual Studio 2019

14.12.5 Creating output files

These statements are used to create output files from the code generation. Remember that all of these statements must be inside a block delimited by square brackets.

create

create filename

creates a new file. The file has to be closed with the **close** statement. All following output is written to the specified file.

Example:

```
[create $outputpath & "/" & $JavaPackageDir & "/" & $application.Name & ".java"]
package [=$JavaPackageName];
public class [=$application.Name]Application {
    ...
}
[close]
```

close

closes the current output file.

=\$variable

writes the value of the specified variable to the current output file.

Example:

```
[\$x = 20+3]
The result of your calculation is [=\$x] - so have a nice day!
```

The file output will be:

The result of your calculation is 23 - so have a nice day!

write

write string

writes the string to the current output file.

Example:

[write "C" & \$name]

This can also be written as:

C[=\$name]

filecopy ... to ...

filecopy source to target

copies the source file to the target file, without any interpretation.

Example:

```
filecopy "java/mapforce/mapforce.png" to $outputpath & "/" & $JavaPackageDir &
"/mapforce.png"
```

14.12.6 Operators

Operators in SPL work like in most other programming languages.

List of SPL operators in descending precedence order:

Access object property

- () Expression grouping
- true boolean constant "true"
- false boolean constant "false"
- & String concatenation
- Sign for negative number
- not Logical negation
- * Multiply
- / Divide
- % Modulo
- + Add
- Subtract
- <= Less than or equal
- < Less than
- >= Greater than or equal
- > Greater than
- = Equal
- <> Not equal
- and Logical conjunction (with short circuit evaluation) or Logical disjunction (with short circuit evaluation)

```
= Assignment
```

14.12.7 Conditions

SPL allows you to use standard "if" statements. The syntax is as follows:

```
if condition
statements
else
statements
```

endif

or, without else:

```
if condition
statements
endif
```

Note: There are no round brackets enclosing the condition.

As in any other programming language, conditions are constructed with logical and comparison operators.

Example:

```
[if $namespace.ContainsPublicClasses and $namespace.Prefix <> ""]
    whatever you want ['inserts whatever you want, in the resulting file]
[endif]
```

Switch

SPL also contains a multiple choice statement.

Syntax:

```
switch $variable
    case X:
        statements
    case Y:
    case Z:
        statements
    default:
        statements
endswitch
```

The case labels must be constants or variables.

The switch statement in SPL does not fall through the cases (as in C), so there is no need for a "break" statement.

14.12.8 Collections and foreach

Collections and iterators

A collection contains multiple objects - like a ordinary array. Iterators solve the problem of storing and incrementing array indexes when accessing objects.

Syntax:

foreach iterator in collection statements

next

Example:

Example 2:

```
[foreach $i in 1 To 3
     Write "// Step " & $i & "\n"
     ' Do some work
next]
```

In the first line:

\$classes is the global object of all generated types. It is a collection of single class objects.

Foreach steps through all the items in \$classes, and executes the code following the instruction, up to the **next** statement, for each of them.

In each iteration, **\$class** is assigned to the next class object. You simply work with the class object instead of using, classes[i]->Name(), as you would in C++.

All collection iterators have the following additional properties:

Index	The current index, starting with 0
IsFirst	true if the current object is the first of the collection (index is 0)
IsLast	true if the current object is the last of the collection
Current	The current object (this is implicit if not specified and can be left out)

Example:

14.12.9 Subroutines

Code generator supports subroutines in the form of procedures or functions.

Features:

- By-value and by-reference passing of values
- Local/global parameters (local within subroutines)
- Local variables
- Recursive invocation (subroutines may call themselves)

14.12.9.1 Subroutine declaration

Subroutines

Syntax example:

```
Sub SimpleSub()
... lines of code
EndSub
```

- Sub is the keyword that denotes the procedure.
- SimpleSub is the name assigned to the subroutine.
- Round parenthesis can contain a parameter list.
- The code block of a subroutine starts immediately after the closing parameter parenthesis.
- EndSub denotes the end of the code block.
- **Note:** Recursive or cascaded subroutine **declaration** is not permitted, i.e. a subroutine may not contain another subroutine.

Parameters

Parameters can also be passed by procedures using the following syntax:

- All parameters must be variables
- Variables must be prefixed by the \$ character
- Local variables are defined in a subroutine
- Global variables are declared explicitly, outside of subroutines
- Multiple parameters are separated by the comma character "," within round parentheses
- Parameters can pass values

Parameters - passing values

Parameters can be passed in two ways, by value and by reference, using the keywords **ByVal** and **ByRef** respectively.

Syntax:

```
' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
] ...
```

- **ByVal** specifies that the parameter is passed by value. Note that most objects can only be passed by reference.
- ByRef specifies that the parameter is passed by reference. This is the default if neither ByVal nor ByRef is specified.

Function return values

To return a value from a subroutine, use the **return** statement. Such a function can be called from within an expression.

Example:

```
' define a function
[Sub MakeQualifiedName( ByVal $namespacePrefix, ByVal $localName )
if $namespacePrefix = ""
  return $localName
else
  return $namespacePrefix & ":" & $localName
endif
EndSub
]
```

14.12.9.2 Subroutine invocation

Use call to invoke a subroutine, followed by the procedure name and parameters, if any.

Call SimpleSub()

or

```
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
```

Function invocation

To invoke a function (any subroutine that contains a **return** statement), simply use its name inside an expression. Do not use the **call** statement to call functions. Example:

\$QName = MakeQualifiedName(\$namespace, "entry")

14.12.9.3 Subroutine example

The following example shows subroutine declaration and invocation.

```
[create $outputpath & $module & "output.txt"
' define sub SimpleSub()
Sub SimpleSub()
]SimpleSub() called
[endsub
' execute sub SimpleSub()
Call SimpleSub()
```

```
$ParamByValue = "Original Value"
]ParamByValue = [=$ParamByValue]
[$ParamByRef = "Original Value"
]ParamByRef = [=$ParamByRef]
' define sub CompleteSub()
[Sub CompleteSub( $param, ByVal $paramByValue, ByRef $paramByRef )
]CompleteSub called.
      param = [=$param]
      paramByValue = [=$paramByValue]
       paramByRef = [=$paramByRef]
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
     Set values inside Sub
1
[$ParamByRef = "Local Variable"
$paramByValue = "new value"
$paramByRef = "new value"
]CompleteSub finished.
[endsub
' run sub CompleteSub()
Call CompleteSub( "FirstParameter", $ParamByValue, $ParamByRef )
ParamByValue=[=$ParamByValue]
ParamByRef=[=$ParamByRef]
Close
]
```

14.12.10 Built in Types

The section describes the properties of the built-in types used in the <u>predefined variables</u> which describe the parsed schema.

14.12.10.1 Library

This object represents the whole library generated from the XML Schema or DTD.

Property	Туре	Description
SchemaNamespaces	Namespace collection	Namespaces in this library
SchemaFilename	string	Name of the XSD or DTD file this library is derived from
SchemaType	integer	1 for DTD, 2 for XML Schema
Guid	string	A globally unique ID
CodeName	string	Generated library name (derived from schema file name)

14.12.10.2 Namespace

One namespace object per XML Schema namespace is generated. Schema components that are not in any namespace are contained in a special namespace object with an empty NamespaceURI. Note that for DTD, namespaces are also derived from attributes whose names begin with "xmlns".

Property	Туре	Description
CodeName	string	Name for generated code (derived from prefix)
LocalName	string	Namespace prefix
NamespaceURI	string	Namespace URI
Types	<u>Type</u> collection	All types contained in this namespace
Library	Library	Library containing this namespace

14.12.10.3 Type

This object represents a complex or simple type. It is used to generate a class in the target language. There is one additional type per library that represents the document, which has all possible root elements as members.

Anonymous types have an empty LocalName.

Property	Туре	Description
CodeName	string	Name for generated code (derived from local name or parent declaration)
LocalName	string	Original name in the schema
Namespace	<u>Namespace</u>	Namespace containing this type
Attributes	Member collection	Attributes contained in this type*
Elements	Member collection	Child elements contained in this type
IsSimpleType	boolean	True for simple types, false for complex types
IsDerived	boolean	True if this type is derived from another type, which is also represented by a Type object
IsDerivedByExtension	boolean	True if this type is derived by extension
IsDerivedByRestriction	boolean	True if this type is derived by restriction
IsDerivedByUnion	boolean	True if this type is derived by union
IsDerivedByList	boolean	True if this type is derived by list
BaseType	Туре	The base type of this type (if IsDerived is true)

Property	Туре	Description
IsDocumentRootType	boolean	True if this type represents the document itself
Library	<u>Library</u>	Library containing this type
IsFinal	boolean	True if declared as final in the schema
IsMixed	boolean	True if this type can have mixed content
IsAbstract	boolean	True if this type is declared as abstract
IsGlobal	boolean	True if this type is declared globally in the schema
IsAnonymous	boolean	True if this type is declared locally in an element

For simple types only:

Property	Туре	Description
IsNativeBound	boolean	True if native type binding exists
NativeBinding	<u>NativeBinding</u>	Native binding for this type
Facets	<u>Facets</u>	Facets of this type
Whitespace	string	Shortcut to the Whitespace facet

* Complex types with text content (these are types with mixed content and complexType with simpleContent) have an additional unnamed attribute member that represents the text content.

14.12.10.4 Member

This object represents an attribute or element in the XML Schema. It is used to create class members of types.

Property	Туре	Description
CodeName	string	Name for generated code (derived from local name or parent declaration)
LocalName	string	Original name in the schema. Empty for the special member representing text content of complex types.
NamespaceURI	string	The namespace URI of this Element/Attribute within XML instance documents/streams.
DeclaringType	<u>Type</u>	Type originally declaring the member (equal to ContainingType for non-inherited members)
ContainingType	<u>Type</u>	Type where this is a member of

Property	Туре	Description
DataType	<u>Type</u>	Data type of this member's content
Library	Library	Library containing this member's DataType
IsAttribute	boolean	True for attributes, false for elements
IsOptional	boolean	True if minOccurs = 0 or optional attribute
IsRequired	boolean	True if minOccurs > 0 or required attribute
IsFixed	boolean	True for fixed attributes, value is in Default property
IsDefault	boolean	True for attributes with default value, value is in Default property
IsNillable	boolean	True for nillable elements
IsUseQualified	boolean	True if NamespaceURI is not empty
MinOccurs	integer	minOccurs, as in schema. 1 for required attributes
MaxOccurs	integer	maxOccurs, as in schema. 0 for prohibited attributes, -1 for unbounded
Default	string	Default value

14.12.10.5 NativeBinding

This object represents the binding of a simple type to a native type in the target programming language, as specified by the "schemanativetype" map.

Property	Туре	Description
ValueType	string	Native type
ValueHandler	string	Formatter class instance

14.12.10.6 Facets

This object represents all facets of a simple type. Inherited facets are merged with the explicitly declared facets. If a Length facet is in effect, MinLength and MaxLength are set to the same value.

Property	Туре	Description
DeclaringType	Туре	Type facets are declared on
Whitespace	string	"preserve", "collapse" or "replace"

Property	Туре	Description
MinLength	integer	Facet value
MaxLength	integer	Facet value
MinInclusive	integer	Facet value
MinExclusive	integer	Facet value
MaxInclusive	integer	Facet value
MaxExclusive	integer	Facet value
TotalDigits	integer	Facet value
FractionDigits	integer	Facet value
List	Facet collection	All facets as list

Facet

This object represents a single facet with its computed value effective for a specific type.

Property	Туре	Description
LocalName	string	Facet name
NamespaceURI	string	Facet namespace
FacetType	string	one of "normalization", "lexicalspace", "valuespace-length", "valuespace-enum" or "valuespace-range"
DeclaringType	<u>Type</u>	Type this facet is declared on
FacetCheckerName	string	Name of facet checker (from schemafacet map)
FacetValue	string or integer	Actual value of this facet

15 The MapForce API

The COM-based API of MapForce enables clients to access the functionality of MapForce from a custom code or application, and automate a wide range of tasks.

The MapForce COM API follows the common specifications for automation servers as set out by Microsoft. MapForce is automatically registered as a COM server object during installation. Once the COM server object is registered, you can invoke it from within applications and scripting languages that have programming support for COM calls. This makes it possible to access the MapForce API not only from development environments using .NET, C++ and Visual Basic, but also from scripting languages like JScript and VBScript.

Note the following:

- If you use the MapForce API to create an application that you intend to distribute to other clients, MapForce must be installed on each client computer. Also, your custom integration code must be deployed to (or your application installed on) each client computer.
- Certain API methods, such as Document.GenerateOutput, require that the MapForce main window is visible, or MapForce (running as a COM server) is embedded within a graphical user interface. If you need to run mappings in a fully unattended manner, across various platforms, consider using MapForce Server (<u>https://www.altova.com/mapforce-server</u>).

15.1 Accessing the API

To access the MapForce COM API, a new instance of the Application object must be created in your application (or script). Once this object is created, you can interact with MapForce by invoking its methods and properties as required (for example, create a new document, open an existing document, generate mapping code, etc).

Prerequisites

To make the MapForce COM object available in your Visual Studio project, add a reference to the MapForce type library (.tlb) file. The following instructions are applicable to Visual Studio 2013, but are similar in other Visual Studio versions:

- 1. On the **Project** menu, click Add Reference.
- 2. Click Browse and select the MapForce.tlb file located in the MapForce installation folder.

A sample MapForce API client in C# is available at: C: \Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\API\C#.

In Java, the MapForce API is available through Java-COM bridge libraries. These libraries are available in the MapForce installation folder: C:\Program Files (x86)\Altova\MapForce2021\JavaAPI (note this path is valid when 32-bit MapForce runs on 64-bit Windows, otherwise adjust the path accordingly).

- AltovaAutomation.dll: a JNI wrapper for Altova automation servers
- AltovaAutomation.jar: Java classes to access Altova automation servers
- MapForceAPI.jar: Java classes that wrap the MapForce automation interface
- MapForceAPI JavaDoc.zip: a Javadoc file containing help documentation for the Java API

To allow access to the MapForce automation server directly from Java code, the libraries above must be in the Java classpath.

A sample MapForce API client in Java is available at: C: \Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\API\Java.

In scripting languages such as JScript or VBScript, the MapForce COM object is accessible through the Microsoft Windows Script Host (see https://msdn.microsoft.com/en-us/library/9bbdkx3k.aspx). Such scripts can be written with a text editor, and do not need compilation, since they are executed by the Windows Script Host packaged with Windows. (To check that the Windows Script Host is running, type wscript.exe/?at the command prompt). Several JScript example files that call the MapForce API are available at: C: \Users (Users

- メモ 32ビホ MapForce では、登録された名前、おさまCOM オブジェクトのプログラム識別子 (ProgId) は以下の通りです: MapForce.Application。64ビホ MapForce では、名前は以下の通りです: MapForce_x64. Application。プログラムの呼び出しは自身のンジストリレイブ、おさなグループ(32-bit おさよ64-bit)内のCLASSES レジストリエトリニアクセスすることは言してくたさい。ですから、標準のコマドプロンプトと上 64-bit Windows のWindows Explorerを使用してスクレフトを実行すると、64-ビホ MapForce.を指す 64-ビホレジストリエトリニアクセスされます。この理由のため、MapForce 32-ビホと64-ビホの両方がインストールされている場合、32-ビット MapForce.を呼び出すために特別な処理が必要にないます。例えば、スクレプトホストがプログラムを呼び出す場合以下を行います。
 - 1. デルケーをC:\Windows\SysWOW64 に変更します。
 - 2. コマボライン上でwscript.exeを入力し実行するスクリプトパンを以下のように入力します。

wscript.exe "C:\Users\...

\Documents\Altova\MapForce2021\MapForceExamples\API\JScript\start.js"

Guidelines

The following guidelines should be considered in your client code:

- Do not hold references to objects in memory longer than you need them. If a user interacts between two calls of your client, then there is no guarantee that these references are still valid.
- Be aware that if your client code crashes, instances of MapForce may still remain in the system. For details on how to avoid error messages, see <u>Error handling</u>.
- Free references explicitly, if using languages such as C++.

Creating the Application object

The syntax to create the starting Application object depends on the programming language, as shown in the examples below:

C#

```
// Create a new instance of MapForce via its automation interface.
MapForceLib.Application objMapForce = new MapForceLib.Application();
```

Java

```
// Start MapForce as COM server.
com.altova.automation.MapForce.Application objMapForce = new Application();
// COM servers start up invisible so we make it visible
objMapForce.setVisible(true);
```

JScript

```
// Access a running instance, or create a new instance of MapForce.
try
{
        objMapForce = WScript.GetObject ("", "MapForce.Application");
        // unhide application if it is a new instance
        objMapForce.Visible = true;
}
catch(err) { WScript.Echo ("Can't access or create MapForce.Application"); }
```

VBA

```
' Create a new instance of MapForce.
Dim objMapForce As Application
Set objMapForce = CreateObject("MapForce.Application")
```

VBScript

```
' Access a running instance, or create a new instance of MapForce.
Set objMapForce = GetObject("MapForce.Application");
```

Visual Basic

Dim objMapForce As MapForceLib.Application = New MapForceLib.Application

15.2 The Object Model

The starting point for every application which uses the MapForce API is the <u>Application</u> object. All other interfaces are accessed through the <u>Application</u> object as the starting point.

The object model of the MapForce API can be represented as follows (each indentation level indicates a child–parent relationship with the level directly above):

Application Options Project ProjectItem <u>Documents</u> **Document MapForceView** Mapping Component Datapoint **Components** Connection **Mappings ErrorMarkers ErrorMarker** AppOutputLines **AppOutputLine** AppOutputLines **AppOutputLineSymbol**

For information about creating an instance the Application object, see <u>Accessing the API</u>. For reference to the objects exposed by the API, see <u>Object Reference</u>.

15.3 Error Handling

The MapForce API returns errors in two different ways. Every API method returns an HRESULT. This return value informs the caller about any malfunctions during the execution of the method. If the call was successful, the return value is equal to s ok. C/C++ programmers generally use HRESULT to detect errors.

Visual Basic, scripting languages, and other high-level development environments do not give the programmer access to the returning HRESULT of a COM call. They use the second error-raising mechanism supported by the MapForce API, the IErrorInfo interface. If an error occurs, the API creates a new object that implements the IErrorInfo interface. The development environment takes this interface and fills its own error-handling mechanism with the provided information.

The following text describes how to deal with errors raised from the MapForce API in different development environments.

Visual Basic

A common way to handle errors in Visual Basic is to define an error handler. This error handler can be set with the On Error statement. Usually the handler displays an error message and does some cleanup to avoid spare references and any kind of resource leaks. Visual Basic fills its own Err object with the information from the IErrorInfo interface.

```
Sub Validate()
  'place variable declarations here
  'set error handler
  On Error GoTo ErrorHandler
  'if generation fails, program execution continues at ErrorHandler:
  objMapForce.ActiveDocument.GenerateXSLT()
  'additional code comes here
  'exit
  Exit Sub
  ErrorHandler:
  MsgBox("Error: " & (Err.Number - vbObjectError) & Chr(13) &
    "Description: " & Err.Description)
End Sub
```

JavaScript

The Microsoft implementation of JavaScript (JScript) provides a try-catch mechanism to deal with errors raised from COM calls. It is very similar to the VisualBasic approach, in that you also declare an error object containing the necessary information.

```
function Generate() {
    // please insert variable declarations here
    try {
        objMapForce.ActiveDocument.GenerateXSLT();
    }
    catch (Error) {
        sError = Error.description;
        nErrorCode = Error.number & 0xffff;
        return false;
    }
    return true;
}
```

C/C++

C/C++ gives you easy access to the HRESULT of the COM call and to the IErrorInterface.

```
HRESULT hr;
// Call GenerateXSLT() from the MapForce API
if(FAILED(hr = ipDocument->GenerateXSLT()))
{
   IErrorInfo *ipErrorInfo = Null;
   if(SUCCEEDED(::GetErrorInfo(0, &ipErrorInfo)))
   {
      BSTR bstrDescr;
      ipErrorInfo->GetDescription(&bstrDescr);
      // handle Error information
      wprintf(L"Error message:\t%s\n",bstrDescr);
      ::SysFreeString(bstrDescr);
      // release Error info
      ipErrorInfo->Release();
   }
}
```

15.4 Example C# Project

After you install MapForce, an example MapForce API client project for C# is available in the directory C: \Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\API.

To compile and run the example, open the solution .sln file in Visual Studio and run **Debug | Start Debugging**.

Note: If you have a 64-bit operating system and are using a 32-bit installation of MapForce, add the **x86** platform in the solution's Configuration Manager and build the sample using this configuration. A new x86 platform (for the active solution in Visual Studio) can be created in the New Solution Platform dialog (**Build | Configuration Manager | Active solution platform | <New...>**).

When you run the example, a Windows form is displayed, containing buttons that invoke basic MapForce operations:

- Start MapForce
- Create a new mapping design
- Open the CompletePO.mfd file from the ...\MapForceExamples folder (note that you may need to adjust the path to point to the \MapForceExamples folder on your machine)
- Generate C# code in a temp directory
- Shut down MapForce

🔜 Form1	_	\times
Start MapForce		
Create new mapping		
Open CompletePO.mfd		
Generate C# code		
OnDocumentOpened Event On/Off		
Shutdown MapForce		

Code listing

The listing is commented for ease of understanding. The code essentially consists of a series of handlers for the buttons in the user interface shown above.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
        public Form1()
        {
            InitializeComponent();
        }
        // An instance of MapForce accessed via its automation interface.
        MapForceLib.Application MapForce;
        // Location of examples installed with MapForce
        String strExamplesFolder;
        private void Form1 Load (object sender, EventArgs e)
        }
        // handler for the "Start MapForce" button
        private void StartMapForce Click(object sender, EventArgs e)
            if (MapForce == null)
            {
                Cursor.Current = Cursors.WaitCursor;
                // if we have no MapForce instance, we create one a nd make it visible.
                MapForce = new MapForceLib.Application();
                MapForce.Visible = true;
                // locate examples installed with MapForce.
                int majorVersionYear = MapForce.MajorVersion + 1998;
                strExamplesFolder = Environment.GetEnvironmentVariable("USERPROFILE") +
 "\\My Documents\\Altova\\MapForce" + Convert.ToString(majorVersionYear) + "\
\MapForceExamples\\";
                Cursor.Current = Cursors.Default;
            }
            else
            {
                // if we have already an MapForce instance running we toggle its
visibility flag.
                MapForce.Visible = !MapForce.Visible;
        }
        // handler for the "Open CompletePO.mfd" button
        private void openCompletePO Click(object sender, EventArgs e)
        {
            if (MapForce == null)
                StartMapForce Click(null, null);
            // Open one of the sample files installed with the product.
            MapForce.OpenDocument(strExamplesFolder + "CompletePO.mfd");
        }
```

```
// handler for the "Create new mapping" button
        private void newMapping Click (object sender, EventArgs e)
            if (MapForce == null)
                StartMapForce Click(null, null);
            // Create a new mapping
            MapForce.NewMapping();
        }
        // handler for the "Shutdown MapForce" button
        // shut-down application instance by explicitly releasing the COM object.
        private void shutdownMapForce Click(object sender, EventArgs e)
        {
            if (MapForce != null)
            {
                // allow shut-down of MapForce by releasing UI
                MapForce.Visible = false;
                // explicitly release COM object
                try
                    while
(System.Runtime.InteropServices.Marshal.ReleaseComObject(MapForce) > 0) ;
                finally
                 {
                    // avoid later access to this object.
                    MapForce = null;
                }
            }
        }
        // handler for button "Generate C# Code"
        private void generateCppCode Click (object sender, EventArgs e)
        {
            if (MapForce == null)
                listBoxMessages.Items.Add("start MapForce first.");
            // COM errors get returned to C# as exceptions. We use a try/catch block to
handle them.
            try
            {
                MapForceLib.Document doc = MapForce.ActiveDocument;
                listBoxMessages.Items.Add("Active document " + doc.Name);
                doc.GenerateCHashCode();
            catch (Exception ex)
                // The COM call was not successful.
                \ensuremath{{\prime}}\xspace // Probably no application instance has been started or no document is
open.
                MessageBox.Show("COM error: " + ex.Message);
            }
        }
        delegate void addListBoxItem delegate(string sText);
        // called from the UI thread
```

```
private void addListBoxItem(string sText)
        {
            listBoxMessages.Items.Add(sText);
        }
        // wrapper method to allow to call UI controls methods from a worker thread
       void syncWithUIthread(Control ctrl, addListBoxItem delegate methodToInvoke,
String sText)
        {
            // Control.Invoke: Executes on the UI thread, but calling thread waits for
completion before continuing.
            // Control.BeginInvoke: Executes on the UI thread, and calling thread
doesn't wait for completion.
           if (ctrl.InvokeRequired)
               ctrl.BeginInvoke(methodToInvoke, new Object[] { sText });
        }
        // event handler for OnDocumentOpened event
       private void handleOnDocumentOpened (MapForceLib.Document i ipDocument)
            String sText = "";
            if (i ipDocument.Name.Length > 0)
               sText = "Document " + i ipDocument.Name + " was opened!";
            else
               sText = "A new mapping was created.";
            // we need to synchronize the calling thread with the UI thread because
            // the COM events are triggered from a working thread
            addListBoxItem delegate methodToInvoke = new
addListBoxItem delegate(addListBoxItem);
           // call syncWithUIthread with the following arguments:
            // 1 - listBoxMessages - list box control to display messages from COM
events
            // 2 - methodToInvoke - a C# delegate which points to the method which
will be called from the UI thread
                                   - the text to be displayed in the list box
            // 3 - sText
            syncWithUIthread(listBoxMessages, methodToInvoke, sText);
        }
       private void checkBoxEventOnOff CheckedChanged(object sender, EventArgs e)
            if (MapForce != null)
            {
                if (checkBoxEventOnOff.Checked)
                    MapForce.OnDocumentOpened += new
MapForceLib. IApplicationEvents OnDocumentOpenedEventHandler(handleOnDocumentOpened);
                else
                    MapForce.OnDocumentOpened -= new
MapForceLib. IApplicationEvents OnDocumentOpenedEventHandler(handleOnDocumentOpened);
           }
        }
    }
}
```

15.5 Example Java Project

After you install MapForce, an example MapForce API client project for Java is available in the directory C: \Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\API.

You can test the Java example directly from the command line, using the batch file BuildAndRun.bat, or you can compile and run the example project from within Eclipse. See below for instructions on how to use these procedures.

File list

The Java examples folder contains all the files required to run the example project. These files are listed below:

AltovaAutomation.dll	Java-COM bridge: DLL part
AltovaAutomation.jar	Java-COM bridge: Java library part
MapForceAPI.jar	Java classes of the MapForce API
RunMapForce.java	Java example source code
BuildAndRun.bat	Batch file to compile and run example code from the command line prompt. Expects folder where Java Virtual Machine resides as parameter.
.classpath	Eclipse project helper file
.project	Eclipse project file
MapForceAPI_JavaDoc.zip	Javadoc file containing help documentation for the Java API

What the example does

The example starts up MapForce and performs a few operations, including opening and closing documents. When done, MapForce stays open. You must close it manually.

Running the example from the command line

To run the example from the command line, open a command prompt window, go to the Java folder of the API Examples folder (*see above for location*), and then type:

```
buildAndRun.bat "<Path-to-the-Java-bin-folder>"
```

The Java binary folder must be that of a Java Development Kit (JDK) 7 or later installation on your computer.

Press the Return key. The Java source in RunMapForce.java will be compiled and then executed.

Loading the example in Eclipse

Open Eclipse and use the **File | Import... | General | Existing Projects into Workspace** command to add the Eclipse project file (.project) located in the Java folder of the API Examples folder (*see above for location*). The project RunMapForce will then appear in your Package Explorer or Navigator.

Select the project and then the command Run as | Java Application to execute the example.

Note: You can select a class name or method of the Java API and press F1 to get help for that class or method.

Java source code listing

The Java source code in the example file RunMapForce.java is listed below with comments.

```
// access general JAVA-COM bridge classes
import java.util.Iterator;
import com.altova.automation.libs.*;
// access XMLSpy Java-COM bridge
import com.altova.automation.MapForce.*;
import com.altova.automation.MapForce.Enums.ENUMProgrammingLanguage;
/**
 * A simple example that starts the COM server and performs a few operations on it.
 * Feel free to extend.
 */
public class RunMapForce
   public static void main(String[] args)
      // an instance of the application.
      Application mapforce = null;
      // instead of COM error handling use Java exception mechanism.
      try
         // Start MapForce as COM server.
         mapforce = new Application();
         // COM servers start up invisible so we make it visible
         mapforce.setVisible(true);
         // The following lines attach to the application events using a default
implementation
         // for the events and override one of its methods.
         // If you want to override all document events it is better to derive your
listener class
         // from DocumentEvents and implement all methods of this interface.
         mapforce.addListener(new ApplicationEventsDefaultHandler()
         {
            ROverride
            public void onDocumentOpened(Document i ipDoc) throws AutomationException
               String name = i ipDoc.getName();
               if (name.length() > 0)
                  System.out.println("Document " + name + " was opened.");
               else
                  System.out.println("A new mapping was created.");
            }
         });
         // Locate samples installed with the product.
         int majorVersionYear = mapforce.getMajorVersion() + 1998;
         String strExamplesFolder = System.getenv("USERPROFILE") + "\\Documents\
\Altova\\MapForce" + Integer.toString(majorVersionYear) + "\\MapForceExamples\\";
         // create a new MapForce mapping and generate c++ code
```

```
Document newDoc = mapforce.newMapping();
         ErrorMarkers err1 = newDoc.generateCodeEx(ENUMProgrammingLanguage.eCpp);
         display(err1);
         // open CompletePO.mfd and generate c++ code
         Document doc = mapforce.openDocument(strExamplesFolder + "CompletePO.mfd");
         ErrorMarkers err2 = doc.generateCodeEx(ENUMProgrammingLanguage.eCpp);
         display(err2);
        doc.close();
         doc = null;
         System.out.println("Watch MapForce!");
     catch (AutomationException e)
         // e.printStackTrace();
     finally
      {
         // Make sure that MapForce can shut down properly.
        if (mapforce != null)
            mapforce.dispose();
         // Since the COM server was made visible and still is visible, it will keep
running
         // and needs to be closed manually.
         System.out.println("Now close MapForce!");
      }
   }
  public static void display (ErrorMarkers err) throws AutomationException
   {
      Iterator<ErrorMarker> itr = err.iterator();
     if (err.getCount() == 0)
          System.out.print("Code generation completed successfully.\n");
     while (itr.hasNext())
      {
         String sError = "";
         Object element = itr.next();
          if (element instanceof ErrorMarker)
             sError = ((ErrorMarker)element).getText();
          System.out.print("Error text: " + sError + "\n");
      }
   }
```

}

15.6 JScript Examples

After you install MapForce, several JScript example files are available in the directory C: \Users\<username>\Documents\Altova\MapForce2021\MapForceExamples\API.

The example files can be run in one of two ways:

- From the command line: Open a command prompt window and type the name of one of the example scripts (for example, start.js). The Windows Scripting Host that is packaged with Windows will execute the script.
- From Windows Explorer: In Windows Explorer, browse for the JScript file and double-click it. The Windows Scripting Host that is packaged with Windows will execute the script. After the script is executed, the command console gets closed automatically.

The following example files are included:

Start.js	Start Mapforce registered as an automation server or connect to a running instance. You can also view this code listing in <u>Start Application</u> .
DocumentAccess.js	Shows how to open, iterate and close documents. You can also view this code listing in Simple Document Access.
GenerateCode.js	Shows how to invoke code generation using JScript. You can also view this code listing in <u>Generate Code</u> .
Readme.txt	Provides basic help to run the scripts.

This documentation additionally includes a few extra JScript code listings:

- Example: Code Generation
- Example: Mapping Execution
- Example: Project Support

15.6.1 Start Application

The JScript code listing below starts the application and shuts it down. If an instance of the application is already running, the running instance will be returned. To run the script, start it from a command prompt window or from Windows Explorer, see also <u>Accessing the API</u>.

```
// Initialize application's COM object. This will start a new instance of the
application and
// return its main COM object. Depending on COM settings, a the main COM object of an
already
// running application might be returned.
try { objMapForce = WScript.GetObject("", "MapForce.Application"); }
catch(err) {}
```

```
if( typeof( objMapForce ) == "undefined" )
{
  try { objMapForce = WScript.GetObject("", "MapForce x64.Application")
                                                                               }
  catch(err)
   {
     WScript.Echo( "Can't access or create MapForce.Application" );
     WScript.Quit();
  }
}
// if newly started, the application will start without its UI visible. Set it to
visible.
objMapForce.Visible = true;
WScript.Echo(objMapForce.Edition + " has successfully started. ");
objMapForce.Visible = false; // will shutdown application if it has no more COM
connections
//objMapForce.Visible = true; // will keep application running with UI visible
```

15.6.2 Simple Document Access

The JScript listing below shows how to open documents, set a document as the active document, iterate through the open documents, and close documents.

```
// Initialize application's COM object. This will start a new instance of the
application and
// return its main COM object. Depending on COM settings, a the main COM object of an
already
// running application might be returned.
try { objMapForce = WScript.GetObject("", "MapForce.Application");
                                                                 }
catch(err) {}
if( typeof( objMapForce ) == "undefined" )
  try { objMapForce = WScript.GetObject("", "MapForce x64.Application")
                                                                         }
  catch(err)
   {
     WScript.Echo( "Can't access or create MapForce.Application" );
     WScript.Quit();
   }
}
// if newly started, the application will start without its UI visible. Set it to
visible.
objMapForce.Visible = true;
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to
major release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objMapForce.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\
\Documents\\Altova\\MapForce" + majorVersionYear + "\\MapForceExamples\\";
```

```
objMapForce.Documents.OpenDocument(strExampleFolder + "CompletePO.mfd");
objMapForce.Documents.OpenDocument(strExampleFolder + "Altova Hierarchical DB.mfd");
// ***************************** code snippet for "Iteration"
// go through all open documents using a JScript Enumerator
for (var iterDocs = new Enumerator(objMapForce.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
  objName = iterDocs.item().Name;
  WScript.Echo("Document name: " + objName);
}
// go through all open documents using index-based access to the document collection
for (i = objMapForce.Documents.Count; i > 0; i--)
  objMapForce.Documents.Item(i).Close();
// ****************************** code snippet for "Iteration"
//objMapForce.Visible = false; // will shutdown application if it has no more COM
connections
objMapForce.Visible = true; // will keep application running with UI visible
```

The code listed above is available as a sample file (see <u>JScript Examples</u>). To run the script, start it from a command prompt window or from Windows Explorer.

15.6.3 Generate Code

The JScript listing below shows how to open documents, set a document as the active document, iterate through the open documents, and generate C++ code.

```
// Initialize application's COM object. This will start a new instance of the
application and
// return its main COM object. Depending on COM settings, a the main COM object of an
already
// running application might be returned.
try { objMapForce = WScript.GetObject("", "MapForce.Application"); }
catch(err) {}
if( typeof( objMapForce ) == "undefined" )
   try { objMapForce = WScript.GetObject("", "MapForce x64.Application") }
  catch(err)
   {
     WScript.Echo( "Can't access or create MapForce.Application" );
     WScript.Quit();
   }
}
// if newly started, the application will start without its UI visible. Set it to
```

```
visible.
objMapForce.Visible = true;
// ********************************** code snippet for "Simple Document Access"
*******
// Locate examples via USERPROFILE shell variable. The path needs to be adapted to
major release versions.
objWshShell = WScript.CreateObject("WScript.Shell");
majorVersionYear = objMapForce.MajorVersion + 1998
strExampleFolder = objWshShell.ExpandEnvironmentStrings("%USERPROFILE%") + "\
\Documents\\Altova\\MapForce" + majorVersionYear + "\\MapForceExamples\\";
objMapForce.Documents.OpenDocument(strExampleFolder + "CompletePO.mfd");
//objMapForce.Documents.OpenDocument(strExampleFolder + "Altova Hierarchical DB.mfd");
objMapForce.Documents.NewDocument();
// ****************************** code snippet for "Simple Document Access"
 *****
// ****************************** code snippet for "Iteration"
objText = "";
// go through all open documents using a JScript Enumerator and generate c++ code
for (var iterDocs = new Enumerator(objMapForce.Documents); !iterDocs.atEnd();
iterDocs.moveNext())
{
  objText += "Generated c++ code result for document " + iterDocs.item().Name + " :
n":
  objErrorMarkers = iterDocs.item().generateCodeEx(1); // ENUMProgrammingLanguage.eCpp
= 1
  bSuccess = true;
  for (var iterErrorMarkers = new
Enumerator(objErrorMarkers); !iterErrorMarkers.atEnd(); iterErrorMarkers.moveNext())
  {
     bSuccess = false;
     objText += "\t" + iterErrorMarkers.item().Text + "\n";
  }
  if (bSuccess)
      objText += "\tCode generation completed successfully.\n";
  objText += "\n";
}
WScript.Echo(objText);
// go through all open documents using index-based access to the document collection
for (i = objMapForce.Documents.Count; i > 0; i--)
  objMapForce.Documents.Item(i).Close();
// ****************************** code snippet for "Iteration"
//objMapForce.Visible = false; // will shutdown application if it has no more COM
connections
objMapForce.Visible = true; // will keep application running with UI visible
```

The code listed above is available as a sample file (see <u>JScript Examples</u>). To run the script, start it from a command prompt window or from Windows Explorer.

15.6.4 Generate Code (Alternative)

The following JScript example shows how to load an existing document and generate different kinds of mapping code for it.

```
// ----- begin JScript example ------
// Generate Code for existing mapping.
// works with Windows scripting host.
// ----- helper function ------
function Exit(strErrorText)
  WScript.Echo(strErrorText);
  WScript.Quit(-1);
}
function ERROR(strText, objErr)
{
  if (objErr != null)
     Exit ("ERROR: (" + (objErr.number & 0xfff) + ")" + objErr.description + " - " +
strText);
  else
     Exit ("ERROR: " + strText);
// -----
// ----- MAIN -----
// ---- create the Shell and FileSystemObject of the windows scripting
try
{
  objWshShell = WScript.CreateObject("WScript.Shell");
  objFS0 = WScript.CreateObject("Scripting.FileSystemObject");
1
catch(err)
   { Exit("Can't create WScript.Shell object"); }
// ----- open MapForce or access running instance and make it visible
try
{
  objMapForce = WScript.GetObject ("", "MapForce.Application");
  objMapForce.Visible = true; // remove this line to perform background
processing
catch(err) { WScript.Echo ("Can't access or create MapForce.Application"); }
// ----- open an existing mapping. adapt this to your needs!
objMapForce.OpenDocument(objFSO.GetAbsolutePathName ("Test.mfd"));
// ----- access the mapping to have access to the code generation methods
var objDoc = objMapForce.ActiveDocument;
// ----- set the code generation output properties and call the code generation
methods.
```

```
// ----- adapt the output directories to your needs
try
{
   // ----- code generation uses some of these options
  var objOptions = objMapForce.Options;
   // ----- generate XSLT --
  objOptions.XSLTDefaultOutputDirectory = "C:\\test\\TestCOMServer\\XSLT";
   objDoc.GenerateXSLT();
   // ----- generate Java Code -----
  objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\Java";
  objDoc.GenerateJavaCode();
  // ----- generate CPP Code, use same cpp code options as the last time -----
  objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\CPP";
  objDoc.GenerateCppCode();
  // ----- generate C# Code, use options C# code options as the last time -----
  objOptions.CodeDefaultOutputDirectory = "C:\\test\\TestCOMServer\\CHash";
  objDoc.GenerateCHashCode();
catch (err)
   { ERROR ("while generating XSL or program code", err); }
// hide MapForce to allow it to shut down
objMapForce.Visible = false;
// ----- end example -----
```

15.6.5 Run a Mapping

The following JScript example shows how to load an existing document with a simple mapping, access its components, set input- and output-instance file names and execute the mapping.

```
/*
  This sample file performs the following operations:
  Load existing MapForce mapping document.
  Find source and target component.
  Set input and output instance filenames.
  Execute the transformation.
  Works with Windows scripting host.
 */
 // ---- general helpers ------
function Exit( message )
 {
   WScript.Echo( message );
   WScript.Quit(-1);
 }
 function ERROR( message, err )
 {
```

```
if( err != null )
     Exit( "ERROR: (" + (err.number & 0xfff) + ") " + err.description + " - " +
message );
  else
     Exit( "ERROR: " + message );
}
// ---- MapForce constants -----
var eComponentUsageKind Unknown
                                   = 0;
var eComponentUsageKind_Instance = 1;
                                   = 2;
var eComponentUsageKind Input
var ecomponentUsageKind_Input = 2;
var eComponentUsageKind_Output = 3;
// ---- MapForce helpers ------
// Searches in the specified mapping for a component by name and returns it.
// If not found, throws an error.
function FindComponent( mapping, component name )
{
   var components = mapping.Components;
   for( var i = 0 ; i < components.Count ; ++i )</pre>
      var component = components.Item( i + 1 );
      if( component.Name == component name )
         return component;
   throw new Error( "Cannot find component with name " + component_name );
}
// Browses components in a mapping and returns the first one found acting as
// source component (i.e. having connections on its right side).
function GetFirstSourceComponent( mapping )
{
   var components = mapping.Components;
   for( var i = 0 ; i < components.Count ; ++i )</pre>
      var component = components.Item( i + 1 );
      if( component.UsageKind == eComponentUsageKind Instance &&
         component.HasOutgoingConnections )
      {
         return component;
      }
   throw new Error( "Cannot find a source component" );
}
// Browses components in a mapping and returns the first one found acting as
// target component (i.e. having connections on its left side).
function GetFirstTargetComponent( mapping )
   var components = mapping.Components;
   for( var i = 0 ; i < components.Count ; ++i )</pre>
   {
```

```
var component = components.Item( i + 1 );
      if ( component.UsageKind == eComponentUsageKind Instance &&
         component.HasIncomingConnections )
         return component;
      }
   throw new Error( "Cannot find a target component" );
}
function IndentTextLines( s )
ł
   return "\t" + s.replace( /\n/g, "\n\t" );
}
function GetAppoutputLineFullText( oAppoutputLine )
ł
   var s = oAppoutputLine.GetLineText();
   var oAppoutputChildLines = oAppoutputLine.ChildLines;
   var i;
   for( i = 0 ; i < oAppoutputChildLines.Count ; ++i )</pre>
   {
      oAppoutputChildLine = oAppoutputChildLines.Item( i + 1 );
      sChilds = GetAppoutputLineFullText( oAppoutputChildLine );
      s += "\n" + IndentTextLines( sChilds );
   }
   return s;
}
// Create a nicely formatted string from AppOutputLines
function GetResultMessagesString( oAppoutputLines )
{
   var s1 = "Transformation result messages:\n";
   var oAppoutputLine;
   var i;
   for( i = 0 ; i < oAppoutputLines.Count ; ++i )</pre>
   {
      oAppoutputLine = oAppoutputLines.Item( i + 1 );
      s1 += GetAppoutputLineFullText( oAppoutputLine );
      s1 += "\n";
   }
   return s1;
}
// ---- MAIN -----
var wshShell;
var fso;
var mapforce;
// create the Shell and FileSystemObject of the windows scripting system
try
```
```
{
  wshShell = WScript.CreateObject( "WScript.Shell" );
   fso = WScript.CreateObject( "Scripting.FileSystemObject" );
catch( err )
   { ERROR( "Can't create windows scripting objects", err ); }
// open MapForce or access currently running instance
try
  mapforce = WScript.GetObject( "", "MapForce.Application" );
catch( err )
   { ERROR( "Can't access or create MapForce.Application", err ); }
try
   // Make MapForce UI visible. This is an API requirement for output generation.
 mapforce.Visible = true;
  // open an existing mapping.
  // **** adjust the examples path to your needs ! *************
  var sMapForceExamplesPath = fso.BuildPath(
            wshShell.SpecialFolders( "MyDocuments" ),
            "Altova\\MapForce2021\\MapForceExamples" );
  var sDocFilename = fso.BuildPath( sMapForceExamplesPath, "PersonList.mfd" );
  var doc = mapforce.OpenDocument( sDocFilename );
  // Find existing components by name in the main mapping.
   // Note, the names of components may not be unique as a schema component's name
  // is derived from its schema file name.
  var source component = FindComponent( doc.MainMapping, "Employees" );
  var target component = FindComponent( doc.MainMapping, "PersonList" );
   // If you do not know the names of the components for some reason, you could
  // use the following functions instead of FindComponent.
   //var source component = GetFirstSourceComponent( doc.MainMapping );
  //var target component = GetFirstTargetComponent( doc.MainMapping );
   // specify the desired input and output files.
  source component.InputInstanceFile = fso.BuildPath( sMapForceExamplesPath,
"Employees.xml" );
  target component.OutputInstanceFile = fso.BuildPath( sMapForceExamplesPath,
"test transformation results.xml" );
   // Perform the transformation.
   // You can use doc.GenerateOutput() if you do not need result messages.
   // If you have a mapping with more than one target component and you want
   // to execute the transformation only for one specific target component,
  // call target component.GenerateOutput() instead.
  var result messages = doc.GenerateOutputEx();
  var summary info =
         "Transformation performed from " + source component.InputInstanceFile + "\n" +
         "to " + target component.OutputInstanceFile + "\n\n" +
         GetResultMessagesString( result messages );
  WScript.Echo( summary info );
}
catch( err )
ł
```

```
ERROR( "Failure", err );
}
```

15.6.6 Project Tasks

The following JScript example shows how to use the MapForce API to automate tasks pertaining to MapForce projects. Before running the example, make sure to edit the variable strSamplePath so that it points to the MapForceExamples folder of your MapForce installation.

To successfully run all operations in this example below, you will need the Enterprise edition of MapForce. If you have the Professional edition, comment out the lines that insert the WebService project.

```
var objMapForce = null;
var objWshShell = null;
var objFSO = null;
// !!! adapt the following path to your needs. !!!
var strSamplePath = "C:\\Users\\<username>\\Documents\\Altova\\MapForce2021\
\MapForceExamples\\";
function Exit(strErrorText)
{
  WScript.Echo(strErrorText);
  WScript.Quit(-1);
}
function ERROR(strText, objErr)
{
  if (objErr != null)
     Exit ("ERROR: (" + (objErr.number & 0xffff) + ")" + objErr.description + " - " +
strText);
  else
     Exit ("ERROR: " + strText);
}
function CreateGlobalObjects ()
{
   // the Shell and FileSystemObject of the windows scripting host often useful
  try
   {
     objWshShell = WScript.CreateObject("WScript.Shell");
     objFS0 = WScript.CreateObject("Scripting.FileSystemObject");
  catch(err)
     { Exit("Can't create WScript.Shell object"); }
  // create the MapForce connection
   // if there is a running instance of MapForce (that never had a connection) - use it
   // otherwise, we automatically create a new instance
  try
   {
     objMapForce = WScript.GetObject("", "MapForce.Application");
   }
```

```
catch(err)
   {
      { Exit("Can't access or create MapForce.Application"); }
  }
}
// --
// print project tree items and their properties recursively.
function PrintProjectTree( objProjectItemIter, strTab )
{
  while ( ! objProjectItemIter.atEnd() )
   {
      // get current project item
      objItem = objProjectItemIter.item();
      try
      {
         // ---- print common properties
         strGlobalText += strTab + "[" + objItem.Kind + "]" + objItem.Name + "\n";
         // ---- print code generation properties, if available
         try
         {
            if ( objItem.CodeGenSettings UseDefault )
               strGlobalText += strTab + " Use default code generation settings\n";
            else
               strGlobalText += strTab + " code generation language is " +
                                                   objItem.CodeGenSettings Language +
                                " output folder is " +
objItem.CodeGenSettings OutputFolder + "\n";
         }
         catch( err ) {}
         // ---- print WSDL settings, if available
         try
         {
            strGlobalText += strTab + " WSDL File is " + objItem.WSDLFile +
                             " Qualified Name is " + objItem.QualifiedName + "\n";
         }
         catch( err ) {}
      1
      catch ( ex )
         { strGlobalText += strTab + "[" + objItem.Kind + "]\n" }
      // ---- recurse
      PrintProjectTree( new Enumerator( objItem ), strTab + ' ');
     objProjectItemIter.moveNext();
  }
}
// --
// Load example project installed with MapForce.
function LoadSampleProject()
{
  // close open project
  objProject = objMapForce.ActiveProject;
  if ( objProject != null )
```

```
objProject.Close();
```

```
// open sample project and iterate through it.
  objProject = objMapForce.OpenProject(strSamplePath + "MapForceExamples.mfp");
   // dump properties of all project items
   strGlobalText = '';
  PrintProjectTree( new Enumerator (objProject), ' ')
  WScript.Echo( strGlobalText );
  objProject.Close();
}
// --
                           _____
// Create a new project with some folders, mappings and a
// Web service project.
// ---
                              _____
function CreateNewProject()
{
   try
   {
      // create new project and specify file to store it.
     objProject = objMapForce.NewProject(strSamplePath + "Sample.mfp");
     // create a simple folder structure
     objProject.CreateFolder( "New Folder 1");
     objFolder1 = objProject.Item(1);
     objFolder1.CreateFolder( "New Folder 2");
     objFolder2 = ( new Enumerator( objFolder1 ) ).item(); // an alternative to
Item(0)
      // add two different mappings to folder structure
     objFolder1.AddFile( strSamplePath + "DB Altova SQLXML.mfd");
     objMapForce.Documents.OpenDocument(strSamplePath + "InspectionReport.mfd");
     objFolder2.AddActiveFile();
      // override code generation settings for this folder
     objFolder2.CodeGenSettings UseDefault = false;
     objFolder2.CodeGenSettings OutputFolder = strSamplePath + "SampleOutput"
     objFolder2.CodeGenSettings Language = 1;
                                                  //C++
      // insert Web service project based on a wsdl file from the installed examples
     objProject.InsertWebService( strSamplePath + "TimeService/TimeService.wsdl",
                                 "{http://www.Nanonull.com/TimeService/}TimeService",
                          "TimeServiceSoap",
                          true );
     objProject.Save();
     if ( ! objProject.Saved )
        WScript.Echo("problem occurred when saving project");
     // dump project tree
      strGlobalText = '';
     PrintProjectTree( new Enumerator (objProject), ' ' )
     WScript.Echo( strGlobalText );
  catch (err)
   { ERROR("while creating new project", err ); }
}
// --
// Generate code for a project's sub-tree. Mix default code
```

```
// generation parameters and overloaded parameters.
// --
function GenerateCodeForNewProject()
{
   // since the Web service project contains only initial mappings,
   // we generate code only for our custom folder.
  // code generation parameters from project are used for Folder1,
  // whereas Folder2 provides overwritten values.
  objFolder = objProject.Item(1);
objFolder1.GenerateCode();
}
CreateGlobalObjects();
objMapForce.Visible = true;
LoadSampleProject();
CreateNewProject();
GenerateCodeForNewProject();
\ensuremath{{//}} uncomment to shut down application when script ends
// objMapForce.Visible = false;
```

15.7 Object Reference

This section provides reference to the objects of the MapForce COM API. The objects are described in a generic manner, since the API may be used with virtually any language that supports calling a COM object. For language-specific examples, see:

- Example C# Project
- Example Java Project
- JScript Examples

The API reference contains two main sections, each describing the interfaces and the enumeration types used in the API, respectively. The enumeration values contain both the string name and a numeric value. If your scripting environment does not support enumerations, use the number-values instead.

In .NET, for every interface of the MapForce COM automation interface, a .NET class exists with the same name. Also, COM types will be converted to the appropriate .NET type. For example, a type such as Long in the COM API would appear as System.Int32 in .NET.

In Java, note the following syntax variations:

- **Classes and class names**. For every interface of the MapForce automation interface, a Java class exists with the name of the interface.
- Method names. Method names on the Java interface are the same as used on the COM interfaces, but start with a small letter to conform to Java naming conventions. To access COM properties, Java methods that prefix the property name with get and set can be used. If a property does not support write-access, no setter method is available. For example, for the Name property of the Document interface, the Java methods getName and setName are available.
- **Enumerations**. For every enumeration defined in the automation interface, a Java enumeration is defined with the same name and values.
- Events and event handlers. For every interface in the automation interface that supports events, a Java interface with the same name plus 'Event' is available. To simplify the overloading of single events, a Java class with default implementations for all events is provided. The name of this Java class is the name of the event interface plus 'DefaultHandler'. For example:

Application // Java class to access the application ApplicationEvents // Events interface for the application ApplicationEventsDefaultHandler // Default handler for "ApplicationEvents"

15.7.1 Interfaces

15.7.1.1 Application

The Application interface is the interface to a MapForce application object. It represents the main access point for the MapForce application itself. This interface is the starting point to do any further operations with MapForce or to retrieve or create other MapForce related automation objects. For information about creating an instance the Application object, see <u>Accessing the API</u>.

Properties to navigate the object model:

- Application
- Parent
- Options
- Project
- Documents

Application status:

- Visible
- Name
- Quit
- Status
- WindowHandle

MapForce designs:

- NewDocument
- OpenDocument
- OpenURL
- ActiveDocument

MapForce projects:

- NewProject
- OpenProject
- ActiveProject

MapForce code generation:

• HighlightSerializedMarker

Global resources:

- GlobalResourceConfig
- GlobalResourceFile

Version information:

- Edition
- IsAPISupported
- MajorVersion
- MinorVersion

Properties

Name	Description	
ActiveDocument	Read-only. Returns the automation object of the currently active document. This property returns the same as Documents.ActiveDocument.	
ActiveProject	Read-only. Returns the automation object of the currently active project.	

Name	Description		
Application	Read-only. Retrieves the application's top-level object.		
Documents	Read-only. Returns a collection of all currently open documents.		
Edition	Read-only. Returns the edition of the application, e.g. "Altova MapForce Enterprise Edition" for the enterprise edition.		
<u>GlobalResourceConfig</u>	Gets or sets the name of the active global resource configuration file. By default, the file is called GlobalResources.xml .		
	The configuration file can be renamed and saved to any location. You can therefore have multiple Global Resources XML files. However, only one of these Global Resources XML File can be active, per application, at one time, and only the definitions contained in this file will be available to the application.		
<u>GlobalResourceFile</u>	Gets or sets the global resource definition file. By default, the file is called GlobalResources.xml.		
<u>IsAPISupported</u>	Read-only. Returns true if the API is supported in this version of MapForce.		
<u>LibraryImports</u>	Read-only. Gets a collection of imported libraries. In the MapForce graphical user interface, these correspond to entries from the Manage Libraries window, added at application level.		
<u>MajorVersion</u>	Read-only. Gets the major version number of MapForce. The version is calculated starting from 1998, and is incremented by one every year. For example, the major version is "18" for the release 2016.		
MinorVersion	Read-only. The minor version number of the product, e.g. 2 for 2006 R2 SP1.		
Name	Read-only. The name of the application.		
Options	Read-only. This property gives access to options that configure the generation of code.		
Parent	Read-only. The parent object according to the object model.		
ServicePackVersion	Read-only.		

Name	Description		
	The service pack version number of the product, e.g. 1 for 2016 R2 SP1.		
<u>Status</u>	Read-only. The status of the application. It is one of the values of the ENUMApplicationStatus enumeration.		
Visible	 True if MapForce is displayed on the screen (though it might be covered by other applications or be iconized). False if MapForce is hidden. The default value for MapForce when automatically started due to a request from the automation server Application is false. In all other cases, the property is initialized to true. An application instance that is visible is said to be controlled by the user (and possibly by clients connected via the automation interface). It will only shut down due to an explicit user request. To shut down an application instance, set its visibility to false and clear all references to this instance within your program. The application instance will shut down automatically when no further COM clients are holding references to it. 		
<u>WindowHandle</u>	Read-only. Retrieves the application's Window Handle.		

Methods

Name	Description		
<u>HighlightSerializedMarker</u>	Use this method to highlight a location in a mapping file that has been previously serialized. If the corresponding document has not already been loaded, it will be loaded first. See Document.GenerateCodeEx for a method to retrieve a serialized marker.		
<u>NewDocument</u>	Creates a new empty document. The newly opened document becomes the ActiveDocument. This method is a shortened form of Documents.NewDocument.		
NewProject	Creates a new empty project. The current project is closed. The new project is accessible under ActiveProject.		
<u>NewWebServiceProject</u>	Creates a new empty Web Service project. The new project is accessible under ActiveProject. This method is available in MapForce Enterprise Edition only.		
<u>OpenDocument</u>	Loads a previously saved document file and continues working on it. The newly opened document becomes the ActiveDocument. This method is a shorter form of Documents.OpenDocument.		

Name	Description	
<u>OpenProject</u>	Opens an existing Mapforce project (*.mfp). The current project is closed. The newly opened project is accessible under ActiveProject.	
<u>OpenURL</u>	Loads a previously saved document file from an URL location. Allows user name and password to be supplied.	
Quit	Disconnects from MapForce to allow the application to shutdown. Calling this method is optional since MapForce keeps track of all external COM connections and automatically recognizes a disconnection. For more information on automatic shutdown see the Visible property.	

Events

Name	Description	
<u>OnDocumentOpened</u>	This event is triggered when an existing or new document is opened. The corresponding close event is Document.OnDocumentClosed.	
<u>OnProjectOpened</u>	This event is triggered when an existing or new project is loaded into the application. The corresponding close event is Project.OnProjectClosed.	
<u>OnShutdown</u>	This event is triggered when the application is shutting down.	

15.7.1.1.1 Properties

15.7.1.1.1.1 ActiveDocument

Returns the automation object of the currently active document. This property returns the same as Documents.ActiveDocument.

Signature

ActiveDocument : Document

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.2 ActiveProject

Returns the automation object of the currently active project.

Signature

```
ActiveProject : Project
```

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.3 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.4 Documents

Returns a collection of all currently open documents.

Signature

Documents : Documents

Error code	Description
1000	The application object is no longer valid.

Error code	Description
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.5 Edition

Returns the edition of the application, e.g. "Altova MapForce Enterprise Edition" for the enterprise edition.

Signature

Edition : String		

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.6 GlobalResourceConfig

Gets or sets the name of the active global resource configuration file. By default, the file is called **GlobalResources.xml**.

The configuration file can be renamed and saved to any location. You can therefore have multiple Global Resources XML files. However, only one of these Global Resources XML File can be active, per application, at one time, and only the definitions contained in this file will be available to the application.

Signature

GlobalResourceConfig : String

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.7 GlobalResourceFile

Gets or sets the global resource definition file. By default, the file is called GlobalResources.xml.

Signature

GlobalResourceFile : String

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.8 IsAPISupported

Returns true if the API is supported in this version of MapForce.

Signature

IsAPISupported : Boolean

Errors

Error code	Description
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.9 LibraryImports

Gets a collection of imported libraries. In the MapForce graphical user interface, these correspond to entries from the **Manage Libraries** window, added at application level.

Signature

LibraryImports : LibraryImports

Error code	Description
1200	The object is no longer valid.

Error code	Description
1201	Invalid address for the return parameter was specified.

15.7.1.1.1.10 MajorVersion

Gets the major version number of MapForce. The version is calculated starting from 1998, and is incremented by one every year. For example, the major version is "18" for the release 2016.

Signature

MajorVersion : Long	
---------------------	--

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.11 MinorVersion

The minor version number of the product, e.g. 2 for 2006 R2 SP1.

Signature

MinorVersion : Long

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.12 Name

The name of the application.

Signature

```
Name : String
```

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.13 Options

This property gives access to options that configure the generation of code.

Signature

Options : Options

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.14 Parent

The parent object according to the object model.

Signature

```
Parent : Application
```

Error code	Description
1000	The application object is no longer valid.

Error code	Description	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.1.15 ServicePackVersion

The service pack version number of the product, e.g. 1 for 2016 R2 SP1.

Signature

ServicePackVersion : Long

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.1.16 Status

The status of the application. It is one of the values of the ENUMApplicationStatus enumeration.

Signature

Status : <u>ENUMApplicationStatus</u>

Errors

Error code	Description
1001	Invalid address for the return parameter was specified.

15.7.1.1.1.17 Visible

True if MapForce is displayed on the screen (though it might be covered by other applications or be iconized).

False if MapForce is hidden. The default value for MapForce when automatically started due to a request from the automation server Application is false. In all other cases, the property is initialized to **true**.

An application instance that is visible is said to be controlled by the user (and possibly by clients connected via the automation interface). It will only shut down due to an explicit user request. To shut down an application instance, set its visibility to false and clear all references to this instance within your program. The application instance will shut down automatically when no further COM clients are holding references to it.

Signature

Visible : Boolean

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.1.18 WindowHandle

Retrieves the application's Window Handle.

Signature

|--|

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.2 Methods

15.7.1.1.2.1 HighlightSerializedMarker

Use this method to highlight a location in a mapping file that has been previously serialized. If the corresponding document has not already been loaded, it will be loaded first. See Document.GenerateCodeEx for a method to retrieve a serialized marker.

Signature

```
HighlightSerializedMarker(in i_strSerializedMarker:String) -> Void
```

Parameters

Name	Туре	Description
i_strSerializedMarker	String	The ErrorMarker object to highlight. Use ErrorMaker.Serialized to obtain this value.

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	
1007	The string passed in i_strSerializedMarker is not recognized as a serialized MapForce marker.	
1008	The marker points to a location that is no longer valid.	

15.7.1.1.2.2 NewDocument

Creates a new empty document. The newly opened document becomes the <code>ActiveDocument</code>. This method is a shortened form of <code>Documents.NewDocument</code>.

Signature

NewDocument() -> Document

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.2.3 NewProject

Creates a new empty project. The current project is closed. The new project is accessible under ActiveProject.

Signature

NewProject() -> Project

Errors

Error code	Description	
1000	The application object is no longer valid.	
1001	Invalid address for the return parameter was specified.	

15.7.1.1.2.4 NewWebServiceProject

Creates a new empty Web Service project. The new project is accessible under ActiveProject. This method is available in MapForce Enterprise Edition only.

Signature

NewWebServiceProject() -> Project

Errors

Error code	Description	
1004	Error creating new project.	
1005	Wrong edition of MapForce.	

15.7.1.1.2.5 OpenDocument

Loads a previously saved document file and continues working on it. The newly opened document becomes the ActiveDocument. This method is a shorter form of Documents.OpenDocument.

Signature

```
OpenDocument(in i_strFileName:String) -> Document
```

Parameters

Name	Туре	Description
i_strFileName	String	The path of the document to open.

Error code	Description
1000	The application object is no longer valid.

Error code	Description
1001	Invalid address for the return parameter was specified.

15.7.1.1.2.6 *OpenProject*

Opens an existing Mapforce project (*.mfp). The current project is closed. The newly opened project is accessible under ActiveProject.

Signature

```
OpenProject(in i_strFileName:String) -> Project
```

Parameters

Name	Туре	Description
i_strFileName	String	The path of the project to open.

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.
1002	The supplied filename is not valid.

15.7.1.1.2.7 OpenURL

Loads a previously saved document file from an URL location. Allows user name and password to be supplied.

Signature

OpenURL(in strURL:String, in strUser:String, in strPassword:String) -> Void

Parameters

Name	Туре	Description
strURL	String	The URL from which the document should be loaded.
strUser	String	The username required to access the URL.

Name	Туре	Description
strPassword	String	The password required to access the URL.

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.
1002	The supplied URL is not valid.
1006	Error while opening the URL file.

15.7.1.1.2.8 Quit

Disconnects from MapForce to allow the application to shutdown. Calling this method is optional since MapForce keeps track of all external COM connections and automatically recognizes a disconnection. For more information on automatic shutdown see the <code>Visible</code> property.

Signature

Quit() -> Void

Errors

Error code	Description
1000	The application object is no longer valid.
1001	Invalid address for the return parameter was specified.

15.7.1.1.3 Events

15.7.1.1.3.1 OnDocumentOpened

This event is triggered when an existing or new document is opened. The corresponding close event is Document.OnDocumentClosed.

Signature

OnDocumentOpened(in i_ipDocument:Document) : Void

15.7.1.1.3.2 OnProjectOpened

This event is triggered when an existing or new project is loaded into the application. The corresponding close event is Project.OnProjectClosed.

Signature

```
OnProjectOpened(in i_ipProject:Project) : Void
```

15.7.1.1.3.3 OnShutdown

This event is triggered when the application is shutting down.

Signature

OnShutdown : Void

15.7.1.2 AppOutputLine

Represents a message line. In contrast to ErrorMarker, its structure is more detailed and can contain a collection of child lines, therefore forming a tree of message lines.

Properties to navigate the object model:

- Application
- Parent

Line access:

- GetLineSeverity
- GetLineSymbol
- GetLineText
- GetLineTextEx
- GetLineTextWithChildren
- GetLineTextWithChildrenEx

A single AppOutputLine consists of one or more sub-lines. Sub-line access:

• GetLineCount

A sub-line consists of one or more cells. Cell access:

- GetCellCountInLine
- GetCellIcon
- GetCellSymbol
- GetCellText
- GetCellTextDecoration
- GetIsCellText

Below an AppOutputLine there can be zero, one, or more child lines which themselves are of type AppOutputLine, which thus form a tree structure.

Child lines access:

• ChildLines

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>ChildLines</u>	Read-only. Returns a collection of the current line's direct child lines.
<u>Parent</u>	Read-only. The parent object according to the object model.

Methods

Name	Description
GetCellCountInLine	Gets the number of cells in the sub-line indicated by nLine in the current AppOutputLine.
<u>GetCellIcon</u>	Gets the icon of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine
<u>GetCellSymbol</u>	Gets the symbol of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine
<u>GetCellText</u>	Gets the text of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine
<u>GetCellTextDecoration</u>	Gets the decoration of the text cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine. It can be one of the ENUMAppOutputLine_TextDecoration values.
<u>GetIsCellText</u>	Returns true if the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine is a text cell.
GetLineCount	Gets the number of sub-lines the current line consists of.
GetLineSeverity	Gets the severity of the line. It can be one of the ENUMAppOutputLine_Severity values.
<u>GetLineSymbol</u>	Gets the symbol assigned to the whole line.
<u>GetLineText</u>	Gets the contents of the line as text.
GetLineTextEx	Gets the contents of the line as text using the specified part and line separators.

Name	Description
<u>GetLineTextWithChildren</u>	Gets the contents of the line including all child and descendant lines as text.
<u>GetLineTextWithChildrenEx</u>	Gets the contents of the line including all child and descendant lines as text using the specified part, line, tab and item separators.

15.7.1.2.1 Properties

15.7.1.2.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.1.2 ChildLines

Returns a collection of the current line's direct child lines.

Signature

ChildLines : AppOutputLines

Error code	Description	
4100	The object is no longer valid.	
4101	Invalid address for the return parameter was specified.	

15.7.1.2.1.3 Parent

The parent object according to the object model.

Signature

```
Parent : AppOutputLines
```

Errors

Error code	Description	
4100	The object is no longer valid.	
4101	Invalid address for the return parameter was specified.	

15.7.1.2.2 Methods

15.7.1.2.2.1 GetCellCountInLine

Gets the number of cells in the sub-line indicated by <code>nLine</code> in the current <code>AppOutputLine</code>.

Signature

GetCellCountInLine(in nLine:Long) -> Long

Parameters

Name	Туре	Description
nLine	Long	Specifies the zero-based index of the line.

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.2 GetCellIcon (obsolete)

Gets the icon of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine

Signature

GetCellIcon(in nLine:Long, in nCell:Long) -> Long

Parameters

Name	Туре	Description
nLine	Long	
nCell	Long	

Errors

Error code	Description	
4100	The object is no longer valid.	
4101	Invalid address for the return parameter was specified.	

15.7.1.2.2.3 GetCellSymbol

Gets the symbol of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine

Signature

```
GetCellSymbol(in nLine:Long, in nCell:Long) -> AppOutputLineSymbol
```

Parameters

Name	Туре	Description
nLine	Long	Specifies the zero-based index of the line.
nCell	Long	Specifies the zero-based index of the cell.

Error code	Description
4100	The object is no longer valid.

Error code	Description
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.4 GetCellText

Gets the text of the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine

Signature

```
GetCellText(in nLine:Long, in nCell:Long) -> String
```

Parameters

Name	Туре	Description
nLine	Long	Specifies the zero-based index of the line.
nCell	Long	Specifies the zero-based index of the cell.

Errors

Error code	Description	
4100	The object is no longer valid.	
4101	Invalid address for the return parameter was specified.	

15.7.1.2.2.5 GetCellTextDecoration

Gets the decoration of the text cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine. It can be one of the ENUMAppOutputLine_TextDecoration values.

Signature

GetCellTextDecoration(in nLine:Long, in nCell:Long) -> Long

Parameters

Name	Туре	Description
nLine	Long	Specifies the zero-based index of the line.

Name	Туре	Description
nCell	Long	Specifies the zero-based index of the cell.

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.6 GetIsCellText

Returns true if the cell indicated by nCell in the current AppOutputLine's sub-line indicated by nLine is a text cell.

Signature

```
GetIsCellText(in nLine:Long, in nCell:Long) -> Boolean
```

Parameters

Name	Туре	Description
nLine	Long	Specifies the zero-based index of the line.
nCell	Long	Specifies the zero-based index of the cell.

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.7 GetLineCount

Gets the number of sub-lines the current line consists of.

Signature

```
GetLineCount() -> Long
```

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.8 GetLineSeverity

Gets the severity of the line. It can be one of the ENUMAppOutputLine_Severity values.

Signature

GetLineSeverity() -> Long

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.9 GetLineSymbol

Gets the symbol assigned to the whole line.

Signature

GetLineSymbol() -> AppOutputLineSymbol

Error code	Description
4100	The object is no longer valid.

Error code	Description
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.10 GetLineText

Gets the contents of the line as text.

Signature

GetLineText() -> String

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.11 GetLineTextEx

Gets the contents of the line as text using the specified part and line separators.

Signature

```
GetLineTextEx(in psTextPartSeperator:String, in psLineSeperator:String) -> String
```

Parameters

Name	Туре	Description
psTextPartSeperator	String	
psLineSeperator	String	

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.12 GetLineTextWithChildren

Gets the contents of the line including all child and descendant lines as text.

Signature

```
GetLineTextWithChildren() -> String
```

Errors

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.2.2.13 GetLineTextWithChildrenEx

Gets the contents of the line including all child and descendant lines as text using the specified part, line, tab and item separators.

Signature

```
GetLineTextWithChildrenEx(in psPartSep:String, in psLineSep:String, in psTabSep:String,
in psItemSep:String) -> String
```

Parameters

Name	Туре	Description
psPartSep	String	
psLineSep	String	
psTabSep	String	
psltemSep	String	

Error code	Description
4100	The object is no longer valid.
4101	Invalid address for the return parameter was specified.

15.7.1.3 AppOutputLines

Represents a collection of AppOutputLine message lines.

Properties to navigate the object model:

- Application
- Parent

Iterating through the collection:

- Count
- Item

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Count	Read-only. Retrieves the number of lines in the collection.
<u>Item</u>	Read-only. Retrieves the line at index ${\rm n}$ from the collection. Indices start with 1.
Parent	Read-only. The parent object according to the object model.

15.7.1.3.1 Properties

15.7.1.3.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Error code	Description
4000	The object is no longer valid.
4001	Invalid address for the return parameter was specified.

15.7.1.3.1.2 Count

Retrieves the number of lines in the collection.

Signature

Count : Integer

Errors

Error code	Description
4000	The object is no longer valid.
4001	Invalid address for the return parameter was specified.

15.7.1.3.1.3 Item

Retrieves the line at index \ensuremath{n} from the collection. Indices start with 1.

Signature

Item(in n:Integer) : AppOutputLine

Errors

Error code	Description
4000	The object is no longer valid.
4001	Invalid address for the return parameter was specified.

15.7.1.3.1.4 Parent

The parent object according to the object model.

Signature

Parent : AppOutputLine

Error code	Description
4000	The object is no longer valid.

Error code	Description
4001	Invalid address for the return parameter was specified.

15.7.1.4 AppOutputLineSymbol

An AppOutputLineSymbol represents a link in an AppOutputLine message line which can be clicked in the MapForce Messages window. It is applied to a cell of an AppOutputLine or to the whole line itself.

Properties to navigate the object model:

- Application
- Parent

Access to AppOutputLineSymbol methods:

- GetSymbolHREF
- GetSymbolID
- IsSymbolHREF

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Parent	Read-only. The parent object according to the object model.

Methods

Name	Description
GetSymbolHREF	If the symbol is of type URL, returns the URL as a string.
<u>GetSymbolID</u>	Gets the ID of the symbol.
IsSymbolHREF	Returns true if the symbol is of kind URL.

15.7.1.4.1 Properties

15.7.1.4.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description
4200	The object is no longer valid.
4201	Invalid address for the return parameter was specified.

15.7.1.4.1.2 Parent

The parent object according to the object model.

Signature

Parent : Application

Error code	Description
4200	The object is no longer valid.
4201	Invalid address for the return parameter was specified.

15.7.1.4.2 Methods

15.7.1.4.2.1 GetSymbolHREF

If the symbol is of type URL, returns the URL as a string.

Signature

GetSymbolHREF() -> String

Errors

Error code	Description
4200	The object is no longer valid.
4201	Invalid address for the return parameter was specified.

15.7.1.4.2.2 GetSymbolID

Gets the ID of the symbol.

Signature

GetSymbolID() -> Long

Errors

Error code	Description
4200	The object is no longer valid.
4201	Invalid address for the return parameter was specified.

15.7.1.4.2.3 IsSymbolHREF

Returns true if the symbol is of kind URL.

Signature

IsSymbolHREF() -> Boolean
Error code	Description	
4200	The object is no longer valid.	
4201	Invalid address for the return parameter was specified.	

15.7.1.5 Component

A Component represents a MapForce component.

To navigate the control, use the Application and Parent properties.

Component properties:

- HasIncomingConnections
- HasOutgoingConnections
- CanChangeInputInstanceFile
- CanChangeOutputInstanceFile
- ComponentName.
- ID
- IsParameterInputRequired
- IsParameterSequence
- Name
- Preview
- Schema
- SubType
- Туре

Instance related properties:

- InputInstanceFile
- OutputInstanceFile

Datapoints:

• GetRootDatapoint

Execution:

• GenerateOutput

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>CanChangeInputInstanceFile</u>	Read-only.

Name	Description
	Indicates if the input instance file name can be changed.
	Returns false if the component has a filename node and this node has a connection on its left (input) side, otherwise returns true. If the component does not have a filename node, false is returned.
<u>CanChangeOutputInstanceFile</u>	Read-only. Indicates if the output instance file name can be changed.
	Returns false if the component has a filename node and this node has a connection on its left (input) side, otherwise returns true. If the component does not have a filename node, false is returned.
ComponentName	Gets or sets the component's name.
<u>HasIncomingConnections</u>	Read-only. Indicates if the component has any incoming connections (on its left side) not including the filename node. An incoming connection on the filename node does not have any effect on the returned value.
<u>HasOutgoingConnections</u>	Read-only. Indicates if the component has any outgoing connections (on its right side).
ID	Read-only. Retrieves the component ID.
<u>InputInstanceFile</u>	Gets or sets the component's input instance file.
<u>IsParameterInputRequired</u>	Gets or sets, if the input parameter component requires an ingoing connection on the function call component of the user- defined function this input parameter component is in. This property works only for components, which are input parameter components.
<u>IsParameterSequence</u>	Gets or sets, if the input or output parameter component supports sequences. This property works only for components, which are input or output parameter components.
Name	Read-only. Gets the component's name.
OutputInstanceFile	Gets or sets the component's output instance file.
	Trying to access the OutputInstanceFile of a component via the API does not return any data if the "File" connector of the component has been connected to another item in the mapping.
Parent	Read-only. The parent object according to the object model.

Name	Description
Preview	Gets or sets if the component is the current preview component.
	This property works only for components, which are target components in the document's main mapping. Only one target component in the main mapping can be the preview component at any time.
	When setting this property, it is only possible to set it to true. This then will also implicitly set the Preview property of all other components to false.
	If there is just a single target component in the main mapping, it is also the preview component.
<u>Schema</u>	Read-only. Retrieves the component's schema file name.
<u>SubType</u>	Read-only. Retrieves the component's subtype.
<u>Туре</u>	Read-only. Retrieves the component's type.
UsageKind	Read-only. Retrieves the component's usage kind.

Methods

Name	Description
<u>GenerateOutput</u>	Generates the output file(s) defined in the mapping for the current component only, using a MapForce internal mapping language. The name(s) of the output file(s) are defined as property of the current component which is the output item in the mapping for this generation process.
<u>GetRootDatapoint</u>	Gets a root datapoint on the left (input) or right (output) side of a component. To access children and descendants, the Datapoint object provides further methods.

15.7.1.5.1 Properties

15.7.1.5.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.5.1.2 CanChangeInputInstanceFile

Indicates if the input instance file name can be changed.

Returns false if the component has a filename node and this node has a connection on its left (input) side, otherwise returns true. If the component does not have a filename node, false is returned.

Signature

```
CanChangeInputInstanceFile : Boolean
```

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.5.1.3 CanChangeOutputInstanceFile

Indicates if the output instance file name can be changed.

Returns false if the component has a filename node and this node has a connection on its left (input) side, otherwise returns true.

If the component does not have a filename node, false is returned.

Signature

CanChangeOutputInstanceFile : Boolean

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.5.1.4 ComponentName

Gets or sets the component's name.

Signature

|--|

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1246	The component does not support setting its name.
1247	Invalid component name.

15.7.1.5.1.5 HasIncomingConnections

Indicates if the component has any incoming connections (on its left side) not including the filename node. An incoming connection on the filename node does not have any effect on the returned value.

Signature

HasIncomingConnections : Boolean

Error code	Description
1200	The object is no longer valid.

Error code	Description
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.6 HasOutgoingConnections

Indicates if the component has any outgoing connections (on its right side).

Signature

HasOutgoingConnections : Boolean

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.7 ID

Retrieves the component ID.

Signature

ID : Long

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.8 InputInstanceFile

Gets or sets the component's input instance file.

Signature

InputInstanceFile : String

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.9 IsParameterInputRequired

Gets or sets, if the input parameter component requires an ingoing connection on the function call component of the user-defined function this input parameter component is in. This property works only for components, which are input parameter components.

Signature

IsParameterInputRequired : Boolean

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1232	This operation works only for an input parameter component.
1240	Changing the document not allowed. It is read-only.

15.7.1.5.1.10 IsParameterSequence

Gets or sets, if the input or output parameter component supports sequences. This property works only for components, which are input or output parameter components.

Signature

IsParameterSequence : Boolean

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1233	This operation works only for an input or output parameter component.
1240	Changing the document not allowed. It is read-only.

15.7.1.5.1.11 Name

Gets the component's name.

Signature

Name : String

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.12 OutputInstanceFile

Gets or sets the component's output instance file.

Trying to access the <code>OutputInstanceFile</code> of a component via the API does not return any data if the "File" connector of the component has been connected to another item in the mapping.

Signature

OutputInstanceFile : String

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.13 Parent

The parent object according to the object model.

Signature

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.14 Preview

Gets or sets if the component is the current preview component.

This property works only for components, which are target components in the document's main mapping. Only one target component in the main mapping can be the preview component at any time.

When setting this property, it is only possible to set it to true. This then will also implicitly set the Preview property of all other components to false.

If there is just a single target component in the main mapping, it is also the preview component.

Signature

```
Preview : Boolean
```

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1234	Only a target component in the main mapping can be set as preview component.
1235	A component cannot be set as non-preview component. Set another component as preview component instead.

15.7.1.5.1.15 Schema

Retrieves the component's schema file name.

Signature

Schema : String

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.16 SubType

Retrieves the component's subtype.

Signature

SubType : <u>ENUMComponentSubType</u>

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.1.17 Туре

Retrieves the component's type.

Signature

Type : <u>ENUMComponentType</u>

15.7.1.5.1.18 UsageKind

Retrieves the component's usage kind.

Signature

UsageKind : <u>ENUMComponentUsageKind</u>

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.5.2 Methods

15.7.1.5.2.1 GenerateOutput

Generates the output file(s) defined in the mapping for the current component only, using a MapForce internal mapping language. The name(s) of the output file(s) are defined as property of the current component which is the output item in the mapping for this generation process.

Signature

```
GenerateOutput(out pbError:Boolean) -> AppOutputLines
```

Parameters

Name	Туре	Description
pbError	Boolean	This is an output-only parameter. You will receive a value only if the calling language supports output parameters. If not, the value you pass here will remain unchanged when the function has finished.

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

Error code	Description
1248	Generating output is only supported when the graphical user interface is visible.

15.7.1.5.2.2 GetRootDatapoint

Gets a root datapoint on the left (input) or right (output) side of a component. To access children and descendants, the Datapoint object provides further methods.

Signature

```
GetRootDatapoint(in side:ENUMComponentDatapointSide, in strNamespace:String, in
strLocalName:String, in strParameterName:String) -> Datapoint
```

Parameters

Name	Туре	Description
side	<u>ENUMComponentDatapointSide</u>	The side parameter indicates if an input, or output, datapoint of a component is to be retrieved.
strNamespace	String	The specified namespace and local name, indicate the specific name of the node whose datapoint is to be retrieved. For components with structural information such as schema components, you will have to provide the namespace together with the local name, or you can just pass an empty string for the namespace. File-based components like the schema component contain a special node on their root, the filename node. There, GetRootDatapoint can only find the filename node. You will have to pass namespace "http://www.altova.com/mapf orce" and local name "FileInstance" to retrieve a datapoint of this node.
strLocalName	String	See above.
strParameterName	String	The specified parameter name should be an empty string unless the component in question is a function call component. Since a user-defined function might

Name	Туре	Description
		contain input or output parameters of the same structure, the function call component calling this user- defined function can have more than one root node with an identical namespace and local name.
		They will then differ only by their parameter names, which are in fact the names of the according parameter components in the user-defined function mapping itself.
		It is not mandatory to specify the parameter name, though. In that case, the method will return the first root datapoint matching the specified namespace and local name.

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1248	Datapoint not found.

15.7.1.6 Components

Represents a collection of Component objects.

To navigate the object model, use the following properties:

- Application
- Parent

To iterate through the collection:

- Count
- Item

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Count	Read-only. Retrieves the number of components in the collection.
Item	Read-only. Retrieves the component at index ${\rm n}$ from the collection. Indices start with 1.
Parent	Read-only. The parent object according to the object model.

15.7.1.6.1 Properties

15.7.1.6.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.6.1.2 Count

Retrieves the number of components in the collection.

Signature

Count : Integer

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.6.1.3 Item

Retrieves the component at index n from the collection. Indices start with 1.

Signature

Item(in n:Integer) : Component

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.6.1.4 Parent

The parent object according to the object model.

Signature

Parent : Mapping

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.7 Connection

A Connection object represents a connector between two components.

To navigate the object model, use the following properties:

- Application
- Parent

To get or set the connection's type, use ConnectionType.

Properties

Name	Description	
Application	Read-only. Retrieves the application's top-level object.	
<u>ConnectionType</u>	Gets or sets the connection's type.	
Parent	Read-only. The parent object according to the object model.	

15.7.1.7.1 Properties

15.7.1.7.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description	
2100	The object is no longer valid.	
2101	Invalid address for the return parameter was specified.	

15.7.1.7.1.2 ConnectionType

Gets or sets the connection's type.

Signature

ConnectionType : <u>ENUMConnectionType</u>

Error code	Description	
2100	The object is no longer valid.	
2101	Invalid address for the return parameter was specified.	
2102	Changing the document is not allowed. It is read-only.	
2103	Failed changing connection type.	

15.7.1.7.1.3 Parent

The parent object according to the object model.

Signature

Parent : Mapping

Errors

Error code	Description	
2100	The object is no longer valid.	
2101	Invalid address for the return parameter was specified.	

15.7.1.8 Datapoint

A Datapoint object represents an input or output icon of a component.

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Parent	Read-only. The parent object according to the object model.

Methods

Name	Description
GetChild	Scans for a direct child datapoint of the current datapoint, by namespace and local name.
	If a schema component has elements that contain mixed content, each displays an additional child node, the so-called text() node. To retrieve a datapoint of a text() node, you will have to pass an empty string in strNamespace as well as "#text" in strLocalName and eSearchDatapointElement in searchFlags.

15.7.1.8.1 Properties

15.7.1.8.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description	
2000	The object is no longer valid.	
2001	Invalid address for the return parameter was specified.	

15.7.1.8.1.2 Parent

The parent object according to the object model.

Signature

```
Parent : Component
```

Error code	Description
2000	The object is no longer valid.

Error code	Description
2001	Invalid address for the return parameter was specified.

15.7.1.8.2 Methods

15.7.1.8.2.1 GetChild

Scans for a direct child datapoint of the current datapoint, by namespace and local name.

If a schema component has elements that contain mixed content, each displays an additional child node, the so-called text() node. To retrieve a datapoint of a text() node, you will have to pass an empty string in strNamespace as well as "#text" in strLocalName and eSearchDatapointElement in searchFlags.

Signature

```
GetChild(in strNamespace:String, in strLocalName:String, in
searchFlags:ENUMSearchDatapointFlags) -> Datapoint
```

Parameters

Name	Туре	Description
strNamespace	String	The namespace of the direct child datapoint.
strLocalName	String	The name of the direct child datapoint.
searchFlags	<u>ENUMSearchDatapointFlags</u>	Search flags can be passed as combination of values (combined using binary OR) of the ENUMSearchDatapointFlags enumeration.

Error code	Description
2000	The object is no longer valid.
2001	Invalid address for the return parameter was specified.
2002	Datapoint not found.

15.7.1.9 Document

A Document object represents a MapForce document (a loaded MFD file). A document contains a main mapping and zero or more local user-defined-function mappings.

To navigate the object model, use the following properties:

- Application
- Parent

For file handling, use:

- Activate
- Close
- FullName
- Name
- Path
- Saved
- Save
- SaveAs

For mapping handling, use:

- MainMapping
- Mappings
- CreateUserDefinedFunction

For component handling, use:

• FindComponentByID

For code generation, use:

- OutputSettings_ApplicationName
- JavaSettings BasePackageName
- GenerateCHashCode
- GenerateCodeEx
- GenerateCppCode
- GenerateJavaCode
- GenerateXQuery
- GenerateXSLT
- GenerateXSLT2
- GenerateXSLT3
- HighlightSerializedMarker

For mapping execution, use:

- GenerateOutput
- GenerateOutputEx

View access:

• MapForceView

Obsolete:

• OutputSettings_Encoding

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>FullName</u>	Path and name of the document file.
JavaSettings_BasePackageName	Sets or retrieves the base package name used when generating Java code. In the MapForce graphical user interface, this setting is available in the Mapping Settings dialog box (right-click the mapping and select Mapping Settings from the context menu).
<u>LibraryImports</u>	Read-only. Gets a collection of imported libraries. In the MapForce graphical user interface, these correspond to entries from the Manage Libraries window, added at document level.
<u>MainMapping</u>	Read-only. Retrieves the main mapping of the document.
<u>MapForceView</u>	Read-only. This property gives access to functionality specific to the MapForce view.
<u>Mappings</u>	Read-only. Returns a collection of mappings contained in the document.
Name	Read-only. Name of the document file without file path.
OutputSettings_ApplicationName	Sets or retrieves the application name available in the Mapping Settings dialog box (To display this dialog box in MapForce, right-click the mapping and select Mapping Settings from the context menu).
OutputSettings_Encoding	This property is no longer supported. Mapping output encoding settings do no longer exist. Components have individual output encoding settings.
Parent	Read-only. The parent object according to the object model.
Path	Read-only. Path of the document file without name.
Saved	Read-only. True if the document was not modified since the last save operation, false otherwise.

Methods

Name	Description
Activate	Makes this document the active document.
<u>Close</u>	Closes the document without saving.
<u>CreateUserDefinedFunction</u>	Creates a user defined function in the current document.
<u>FindComponentByID</u>	Searches in the whole document, also all its mappings, for the component with the specified id.
<u>GenerateCHashCode</u>	Generates C# code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateCodeEx</u>	Generates code that will perform the mapping. The parameter i_nLanguage specifies the target language. The method returns an object that can be used to enumerate all messages created by the code generator. These are the same messages that get displayed in the Messages window of MapForce.
<u>GenerateCppCode</u>	Generates C++ code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateJavaCode</u>	Generates Java code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateOutput</u>	Generates all output files defined in the mapping using a MapForce internal mapping language. The names of the output files are defined as properties of the output items in the mapping.
	Note: This method can only be used when the MapForce (running as a COM server) main window is visible, or is embedded with a graphical user interface. If the method is called while MapForce is not visible, then an error will occur.
GenerateOutputEx	Generates all output files defined in the mapping using a MapForce internal mapping language. The names of the output files are defined as properties of the output items in the mapping. This method is identical to GenerateOutput except for its return value containing the resulting messages, warnings and errors arranged as trees of AppOutputLines.
	Note: This method can only be used when the MapForce (running as a COM server) main window is visible, or is embedded with a graphical user interface. If the method is called while MapForce is not visible, then an error will occur.

Name	Description
<u>GenerateXQuery</u>	Generates mapping code as XQuery. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateXSLT</u>	Generates mapping code as XSLT. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateXSLT2</u>	Generates mapping code as XSLT2. Uses the properties defined in Application.Options to configure code generation.
<u>GenerateXSLT3</u>	Generates XSLT 3.0 mapping code. Uses the properties defined in Application.Options to configure code generation.
<u>HighlightSerializedMarker</u>	Use this method to highlight a location in a mapping file that has been previously serialized. If the corresponding document is not already loaded, it will be loaded first. See GenerateCodeEx for a method to retrieve a serialized marker.
Save	Saves the document to the file defined by Document.FullName.
SaveAs	Saves the document to the specified file name, and sets Document.FullName to this value if the save operation was successful.

Events

Name	Description
<u>OnDocumentClosed</u>	This event is triggered when a document is closed. The document object passed into the event handler should not be accessed. The corresponding open event is Application.OnDocumentOpened.
<u>OnModifiedFlagChanged</u>	This event is triggered when a document's modification status changes.

15.7.1.9.1 Properties

15.7.1.9.1.1 Application

Retrieves the application's top-level object.

Signature

|--|

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.2 FullName

Path and name of the document file.

Signature

FullName : String

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.3 JavaSettings_BasePackageName

Sets or retrieves the base package name used when generating Java code. In the MapForce graphical user interface, this setting is available in the **Mapping Settings** dialog box (right-click the mapping and select **Mapping Settings** from the context menu).

Signature

JavaSettings_BasePackageName : String

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.4 LibraryImports

Gets a collection of imported libraries. In the MapForce graphical user interface, these correspond to entries from the **Manage Libraries** window, added at document level.

Signature

LibraryImports : LibraryImports

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.5 MainMapping

Retrieves the main mapping of the document.

Signature

```
MainMapping : Mapping
```

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.6 MapForceView

This property gives access to functionality specific to the MapForce view.

Signature

MapForceView : MapForceView

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.7 Mappings

Returns a collection of mappings contained in the document.

Signature

Mappings : Mappings

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.8 Name

Name of the document file without file path.

Signature

Name : String

Error code	Description
1200	The object is no longer valid.

Error code	Description
1201	Invalid address for the return parameter was specified.

15.7.1.9.1.9 OutputSettings_ApplicationName

Sets or retrieves the application name available in the **Mapping Settings** dialog box (To display this dialog box in MapForce, right-click the mapping and select **Mapping Settings** from the context menu).

Signature

OutputSettings_ApplicationName : String

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.9.1.10 OutputSettings_Encoding (obsolete)

This property is no longer supported. Mapping output encoding settings do no longer exist. Components have individual output encoding settings.

Signature

OutputSettings_Encoding : String

15.7.1.9.1.11 Parent

The parent object according to the object model.

Signature

Parent : Documents

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.9.1.12 Path

Path of the document file without name.

Signature

Path : String

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.9.1.13 Saved

True if the document was not modified since the last save operation, false otherwise.

Signature

Saved : Boolean

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.9.2 Methods

15.7.1.9.2.1 Activate

Makes this document the active document.

Signature

Activate() -> Void

Error code	Description
1200	The object is no longer valid.

15.7.1.9.2.2 Close

Closes the document without saving.

Signature

Close() -> Void

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.9.2.3 CreateUserDefinedFunction

Creates a user defined function in the current document.

Signature

```
CreateUserDefinedFunction(in strFunctionName:String, in strLibraryName:String, in strSyntax:String, in strDetails:String, in bInlinedUse:Boolean) -> Mapping
```

Parameters

Name	Туре	Description
strFunctionName	String	The name of the function.
strLibraryName	String	The name of the library to which this function belongs.
strSyntax	String	A string that describes the syntax of this function (this is for information purpose only).
strDetails	String	A description of this function.

Name	Туре	Description
bInlinedUse	Boolean	Boolean flag that specifies if the function has inlined use.

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	
1208	Failed creating user-defined function.	
1209	Changing the document not allowed. It is read-only.	

15.7.1.9.2.4 FindComponentByID

Searches in the whole document, also all its mappings, for the component with the specified id.

Signature

FindComponentByID(in nID:Unsigned Long) -> Component

Parameters

Name	Туре	Description
nID	Unsigned Long	The ID of the component to search for.

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.2.5 GenerateCHashCode

Generates C# code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.

Signature

GenerateCHashCode() -> Void

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1205	Error during code generation.

15.7.1.9.2.6 GenerateCodeEx

Generates code that will perform the mapping. The parameter **i_nLanguage** specifies the target language. The method returns an object that can be used to enumerate all messages created by the code generator. These are the same messages that get displayed in the Messages window of MapForce.

Signature

GenerateCodeEx(in i_nLanguage:ENUMProgrammingLanguage) -> ErrorMarkers

Parameters

Name	Туре	Description
i_nLanguage	ENUMProgrammingLanguage	Specifies the target code generation language.

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1205	Error during code generation.

15.7.1.9.2.7 GenerateCppCode

Generates C++ code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.

Signature

GenerateCppCode() -> Void

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1205	Error during code generation.

15.7.1.9.2.8 GenerateJavaCode

Generates Java code that will perform the mapping. Uses the properties defined in Application.Options to configure code generation.

Signature

GenerateJavaCode() -> **Void**

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1205	Error during code generation.

15.7.1.9.2.9 GenerateOutput

Generates all output files defined in the mapping using a MapForce internal mapping language. The names of the output files are defined as properties of the output items in the mapping.

Note: This method can only be used when the MapForce (running as a COM server) main window is visible, or is embedded with a graphical user interface. If the method is called while MapForce is not visible, then an error will occur.

Signature

GenerateOutput() -> Void

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1206	Error during execution of mapping algorithm.
1210	Generating output is only supported when the graphical user interface is visible.

15.7.1.9.2.10 GenerateOutputEx

Generates all output files defined in the mapping using a MapForce internal mapping language. The names of the output files are defined as properties of the output items in the mapping. This method is identical to GenerateOutput except for its return value containing the resulting messages, warnings and errors arranged as trees of AppOutputLines.

Note: This method can only be used when the MapForce (running as a COM server) main window is visible, or is embedded with a graphical user interface. If the method is called while MapForce is not visible, then an error will occur.

Signature

GenerateOutputEx() -> AppOutputLines

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1206	Error during execution of mapping algorithm.
1210	Generating output is only supported when the graphical user interface is visible.

15.7.1.9.2.11 GenerateXQuery

Generates mapping code as XQuery. Uses the properties defined in ${\tt Application.Options}$ to configure code generation.

Signature

GenerateXQuery() -> Void

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1204	Error during XSLT/XSLT2/XSLT3/XQuery code generation.

15.7.1.9.2.12 GenerateXSLT

Generates mapping code as XSLT. Uses the properties defined in Application.Options to configure code generation.

Signature

GenerateXSLT() -> Void

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1204	Error during XSLT/XSLT2/XSLT3/XQuery code generation.

15.7.1.9.2.13 GenerateXSLT2

Generates mapping code as XSLT2. Uses the properties defined in Application.Options to configure code generation.

Signature

```
GenerateXSLT2() -> Void
```

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1204	Error during XSLT/XSLT2/XSLT3/XQuery code generation.

15.7.1.9.2.14 GenerateXSLT3

Generates XSLT 3.0 mapping code. Uses the properties defined in Application.Options to configure code generation.

Signature

```
GenerateXSLT3() -> Void
```

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1204	Error during XSLT/XSLT2/XSLT3/XQuery code generation.

15.7.1.9.2.15 HighlightSerializedMarker

Use this method to highlight a location in a mapping file that has been previously serialized. If the corresponding document is not already loaded, it will be loaded first. See GenerateCodeEx for a method to retrieve a serialized marker.

Signature

```
HighlightSerializedMarker(in i_strSerializedMarker:String) -> Void
```

Parameters

Name	Туре	Description
i_strSerializedMarker	String	The ErrorMarker object to highlight. Use ErrorMaker.Serialized to obtain this value.

Error code	Description
1000	The object is no longer valid.
1001	Invalid address for the return parameter was specified.
1007	The string passed in i_strSerializedMarker is not recognized a serialized MapForce marker.

Error code	Description
1008	The marker points to a location that is no longer valid.

15.7.1.9.2.16 Save

Saves the document to the file defined by Document.FullName.

Signature

Save() -> Void

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.9.2.17 SaveAs

Saves the document to the specified file name, and sets Document.FullName to this value if the save operation was successful.

Signature

```
SaveAs(in i_strFileName:String) -> Void
```

Parameters

Name	Туре	Description
i_strFileName	String	Specifies the path where to save the document.

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
15.7.1.9.3 Events

15.7.1.9.3.1 OnDocumentClosed

This event is triggered when a document is closed. The document object passed into the event handler should not be accessed. The corresponding open event is Application.OnDocumentOpened.

Signature

```
OnDocumentClosed(in i ipDocument:Document) : Void
```

15.7.1.9.3.2 OnModifiedFlagChanged

This event is triggered when a document's modification status changes.

Signature

```
OnModifiedFlagChanged(in i_blsModified:Boolean) : Void
```

15.7.1.10 Documents

Represents a collection of Document objects.

Properties to navigate the object model:

- Application
- Parent

Open and create mappings:

- OpenDocument
- NewDocument

Iterating through the collection:

- Count
- Item
- ActiveDocument

Properties

Name	Description	
ActiveDocument	Read-only. Retrieves the active document. If no document is open, null is returned.	

Name	Description	
Application	Read-only. Retrieves the application's top-level object.	
Count	Read-only. Retrieves the number of documents in the collection.	
<u>Item</u>	Read-only. Retrieves the document at index ${\rm n}$ from the collection. Indices start with 1.	
Parent	Read-only. The parent object according to the object model.	

Methods

Name	Description	
NewDocument	Creates a new document, adds it to the end of the collection, and makes it the active document.	
<u>OpenDocument</u>	Opens an existing mapping document (*.mfd). Adds the newly opened document to the end of the collection and makes it the active document.	

15.7.1.10.1 Properties

15.7.1.10.1.1 ActiveDocument

Retrieves the active document. If no document is open, null is returned.

Signature

ActiveDocument : Document

Error code	Description	
1600	The object is no longer valid.	
1601	Invalid address for the return parameter was specified.	

15.7.1.10.1.2 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description	
1600	The object is no longer valid.	
1601	Invalid address for the return parameter was specified.	

15.7.1.10.1.3 Count

Retrieves the number of documents in the collection.

Signature

Count : Integer

Errors

Error code	Description	
1600	The object is no longer valid.	
1601	Invalid address for the return parameter was specified.	

15.7.1.10.1.4 Item

Retrieves the document at index n from the collection. Indices start with 1.

Signature

Item(in n:Integer) : Document

Error code	Description
1600	The object is no longer valid.

Error code	Description
1601	Invalid address for the return parameter was specified.

15.7.1.10.1.5 Parent

The parent object according to the object model.

Signature

Parent : Application	
----------------------	--

Errors

Error code	Description	
1600	The object is no longer valid.	
1601	Invalid address for the return parameter was specified.	

15.7.1.10.2 Methods

15.7.1.10.2.1 NewDocument

Creates a new document, adds it to the end of the collection, and makes it the active document.

Signature

NewDocument() -> Document

Errors

Error code	Description	
1600	The object is no longer valid.	
1601	Invalid address for the return parameter was specified.	

15.7.1.10.2.2 OpenDocument

Opens an existing mapping document (*.mfd). Adds the newly opened document to the end of the collection and makes it the active document.

Signature

```
OpenDocument(in strPath:String) -> Document
```

Parameters

Name	Туре	Description
strPath	String	The path of the mapping file.

Errors

Error code	Description
1600	The object is no longer valid.
1601	Invalid address for the return parameter was specified.

15.7.1.11 ErrorMarker

Represents a simple message line. Unlike AppOutputLine objects, error markers do not have a hierarchical structure.

Properties to navigate the object model:

- Application
- Parent

Access to message information:

- DocumentFileName
- ErrorLevel
- Highlight
- Serialization
- Text

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>DocumentFileName</u>	Read-only. Retrieves the name of the mapping file that the error marker is associated with.
ErrorLevel	Read-only. Retrieves the severity of the error.

Name	Description
Parent	Read-only. The parent object according to the object model.
<u>Serialization</u>	Read-only. Serialize error marker into a string. Use this string in calls to Application.HighlightSerializedMarker Or Document.HighlightSerializedMarker to highlight the marked item in the mapping. The string can be persisted and used in other instantiations of MapForce or its Control.
Text	Read-only. Retrieves the message text.

Methods

Name	Description
Highlight	Highlights the item that the error marker is associated with. If the corresponding document is not open, it will be opened.

15.7.1.11.1 Properties

15.7.1.11.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.1.2 DocumentFileName

Retrieves the name of the mapping file that the error marker is associated with.

Signature

DocumentFileName : String

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.1.3 ErrorLevel

Retrieves the severity of the error.

Signature

ErrorLevel : <u>ENUMCodeGenErrorLevel</u>

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.1.4 Parent

The parent object according to the object model.

Signature

Parent : ErrorMarkers

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.1.5 Serialization

Serialize error marker into a string. Use this string in calls to Application.HighlightSerializedMarker or Document.HighlightSerializedMarker to highlight the marked item in the mapping. The string can be persisted and used in other instantiations of MapForce or its Control.

Signature

Serialization : String

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.1.6 Text

Retrieves the message text.

Signature

Text : String

Errors

Error code	Description
1900	The object is no longer valid.
1901	Invalid address for the return parameter was specified.

15.7.1.11.2 Methods

15.7.1.11.2.1 Highlight

Highlights the item that the error marker is associated with. If the corresponding document is not open, it will be opened.

Signature

Highlight() -> Void

Error code	Description
1900	The object is no longer valid.

Error code	Description	
1901	Invalid address for the return parameter was specified.	
1008	The marker points to a location that is no longer valid.	

15.7.1.12 ErrorMarkers

Represents a collection of ErrorMarker objects.

Properties to navigate the object model:

- Application
- Parent

Iterating through the collection:

- Count
- Item

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Count	Read-only. Retrieves the number of error markers in the collection.
Item	Read-only. Retrieves the error marker at index ${\tt n}$ from the collection. Indices start with 1.
Parent	Read-only. The parent object according to the object model.

15.7.1.12.1 Properties

15.7.1.12.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application	
---------------------------	--

Errors

Error code	Description	
1800	The object is no longer valid.	
1801	Invalid address for the return parameter was specified.	

15.7.1.12.1.2 Count

Retrieves the number of error markers in the collection.

Signature

Count : Integer

Errors

Error code	Description	
1800	The object is no longer valid.	
1801	Invalid address for the return parameter was specified.	

15.7.1.12.1.3 Item

Retrieves the error marker at index n from the collection. Indices start with 1.

Signature

Item(in n:Integer) : ErrorMarker

Errors

Error code	Description	
1800	The object is no longer valid.	
1801	Invalid address for the return parameter was specified.	

15.7.1.12.1.4 Parent

The parent object according to the object model.

Signature

Parent : Application

Errors

Error code	Description
1800	The object is no longer valid.
1801	Invalid address for the return parameter was specified.

15.7.1.13 LibraryImport

A LibraryImport represents an imported library file (an entry from the Manage Libraries window).

Properties

Name	Description
Application	Read-only. Retrieves the application's top level object.
<u>Parent</u>	Read-only. Retrieves the parent object, according to the object model.
Path	Read-only. Gets the path of the imported library.
<u>SaveRelativePath</u>	When you save the document, this property specifies whether the library path should be saved as absolute or relative. When true , the path of the library will be relative to the document. When false , the library path will be absolute.
	absolute or relative, since the path may have been changed

Name	Description
	(either from the user interface or via API) since the document was loaded from the .mfd file.
	If you set this property (either via API or via user interface), the "Manage Libraries" window does immediately show the correct path's state in the user interface. Internally, however, the Path of the ImportedLibrary object will not be changed until the document is saved.
	Libraries imported globally cannot be saved with a relative path. Only libraries imported at document level can.

15.7.1.13.1 Properties

15.7.1.13.1.1 Application

Retrieves the application's top level object.

Signature

Application : Application

Errors

Error code	Description	
2500	The object is no longer valid.	
2501	Invalid address for the return parameter was specified.	

15.7.1.13.1.2 Parent

Retrieves the parent object, according to the object model.

Signature

Parent : LibraryImports

Error code	Description
2500	The object is no longer valid.

Error code	Description
2501	Invalid address for the return parameter was specified.

15.7.1.13.1.3 Path

Gets the path of the imported library.

Signature

Path : String

Errors

Error code	Description	
2500	The object is no longer valid.	
2501	Invalid address for the return parameter was specified.	

15.7.1.13.1.4 SaveRelativePath

When you save the document, this property specifies whether the library path should be saved as absolute or relative. When **true**, the path of the library will be relative to the document. When **false**, the library path will be absolute.

Do not rely on this property to determine whether the path is absolute or relative, since the path may have been changed (either from the user interface or via API) since the document was loaded from the .mfd file.

If you set this property (either via API or via user interface), the "Manage Libraries" window does immediately show the correct path's state in the user interface. Internally, however, the Path of the ImportedLibrary object will not be changed until the document is saved.

Libraries imported globally cannot be saved with a relative path. Only libraries imported at document level can.

Signature

SaveRelativePath : Boolean

Error code	Description
2500	The object is no longer valid.

Error code	Description	
2501	Invalid address for the return parameter was specified.	
2502	Global imports cannot be saved with a relative path.	

15.7.1.14 LibraryImports

Represents a collection of imported libraries (LibraryImport objects). Use the Application and Parent properties to navigate the object model. Use the Count and Item properties to iterate through the collection. You can get this collection as follows:

- Locally (at document level), through the Document.LibraryImports property
- Globally (at application level), through the Application.LibraryImports property.

If you get the $\tt LibraryImports$ collection from the application object, the <code>Parent</code> property of the collection will be null.

Properties

Name	Description	
Application	Read-only. Gets the application's top level object.	
Count	Read-only. Gets the count of LibraryImport objects in this collection.	
<u>Item</u>	Read-only. Retrieves a library entry at index <i>n</i> from this collection. The index is 1-based.	
<u>Parent</u>	Read-only. Gets the parent document for local library imports. If you get the LibraryImports collection from the application object, the Parent property of the collection will be null.	

Methods

Name	Description	
<u>Add</u>	Adds a new library to this LibraryImports object. The new library will have the path supplied by the i_strFileName parameter.	
Find	Returns a library reference given the path to the library file.	
Remove	Removes a library reference from the Manage Libraries window.	

15.7.1.14.1 Properties

15.7.1.14.1.1 Application

Gets the application's top level object.

Signature

Application : Application

Errors

Error code	Description	
2400	The object is no longer valid.	
2401	Invalid address for the return parameter was specified.	

15.7.1.14.1.2 Count

Gets the count of LibraryImport objects in this collection.

Signature

Count : Integer

Errors

Error code	Description	
2400	The object is no longer valid.	
2401	Invalid address for the return parameter was specified.	

15.7.1.14.1.3 Item

Retrieves a library entry at index n from this collection. The index is 1-based.

Signature

Item(in n:Integer) : LibraryImport

Errors

Error code	Description	
2400	The object is no longer valid.	
2401	Invalid address for the return parameter was specified.	

15.7.1.14.1.4 Parent

Gets the parent document for local library imports. If you get the LibraryImports collection from the application object, the Parent property of the collection will be null.

Signature

Parent : Document

Errors

Error code	Description	
2400	The object is no longer valid.	
2401	Invalid address for the return parameter was specified.	

15.7.1.14.2 Methods

15.7.1.14.2.1 Add

Adds a new library to this LibraryImports object. The new library will have the path supplied by the **i_strFileName** parameter.

Signature

```
Add(in i_strFileName:String) -> LibraryImport
```

Parameters

Name	Туре	Description
i_strFileName	String	Specifies the path of the library file. This path can be either absolute or relative to the mapping, depending on the state in which it was passed to the

Name	Туре	Description
		object. When the document is saved, the path will be made relative if the LibraryImport.SaveRelativeP ath flag is true ; otherwise, it will be made absolute.

Errors

Error code	Description
2400	The object is no longer valid.
2401	Invalid address for the return parameter was specified.
2402	Adding library file failed.

15.7.1.14.2.2 Find

Returns a library reference given the path to the library file.

Signature

```
Find(in i_strFileName:String) -> LibraryImport
```

Parameters

Name	Туре	Description
i_strFileName	String	The path of the library file to search for. For locally imported libraries, you can specify either the absolute or the relative path to the library file (unlike the Remove method, which requires the exact path).
		For globally imported libraries, the path must always be absolute (since globally imported libraries cannot have a relative path).

Error code	Description
2400	The object is no longer valid.

Error code	Description
2401	Invalid address for the return parameter was specified.

15.7.1.14.2.3 Remove

Removes a library reference from the Manage Libraries window.

Signature

Remove(in i_strFileName:String) -> Void

Parameters

Name	Туре	Description
i_strFileName	String	The path of the library file to remove. Note that the path must reflect exactly the current (most recent) state of the LibraryImport object. Remember that the path may be either relative or absolute, and it may have changed if you saved the document, depending on the LibraryImport.SaveRelativePath flag. Therefore, if the LibraryImport object currently contains a relative path, then you should supply a relative path as value of this parameter. Otherwise, the library is not found and the Remove method fails.
		The above applies only for locally imported libraries. For globally imported libraries, the path must always be absolute (since globally imported libraries cannot have a relative path)

Error code	Description
2400	The object is no longer valid.

15.7.1.15 MapForceView

Represents the current view in the MapForce Mapping tab for a document. A document has exactly one MapForceView which displays the currently active mapping.

Properties to navigate the object model:

- Application
- Parent

View activation and view properties:

- Active
- ShowItemTypes
- ShowLibraryInFunctionHeader
- HighlightMyConnections
- HighlightMyConnectionsRecursivly

Mapping related properties:

- ActiveMapping
- ActiveMappingName

Adding items:

- InsertWSDLCall
- InsertXMLFile
- InsertXMLSchema
- InsertXMLSchemaWithSample

Properties

Name	Description
<u>Active</u>	Use this property to query if the mapping view is the active view, or set this view to be the active one.
<u>ActiveMapping</u>	Gets or sets the currently active mapping in the document this MapForceView belongs to.
<u>ActiveMappingName</u>	Gets or sets the currently active mapping by name in the document this MapForceView belongs to.
Application	Read-only. Retrieves the application's top-level object.
<u>HighlightMyConnections</u>	This property defines whether connections from the selected item only should be highlighed.
HighlightMyConnectionsRecursively	This property defines if only the connections coming directly or indirectly from the selected item should be highlighted.
Parent	Read-only. The parent object according to the object model.

Name	Description
<u>ShowItemTypes</u>	This property defines if types of items should be shown in the mapping diagram.
ShowLibraryInFunctionHeader	This property defines whether the name of the function library should be part of function names.

Methods

Name	Description
<u>InsertWSDLCall</u>	Adds a new WSDL call component to the mapping.
<u>InsertXMLFile</u>	MapForceView.InsertXMLFile is obsolete . Use Mapping.InsertXMLFile instead .
<u>InsertXMLSchema</u>	MapForceView.InsertXMLSchema is obsolete. Use Mapping.InsertXMLSchema instead .
<u>InsertXMLSchemaWithSample</u>	MapForceView.InsertXMLSchemaWithSample is obsolete. Use Mapping.InsertXMLFile instead. Notice, Mapping.InsertXMLFile does not require a parameter for passing the root element. The root element is automatically set as the XML file's root element name.

15.7.1.15.1 Properties

15.7.1.15.1.1 Active

Use this property to query if the mapping view is the active view, or set this view to be the active one.

Signature

		_			
Active	:	Boolean			

Error code	Description		
1300	The object is no longer valid.		
1301	Invalid address for the return parameter was specified.		

15.7.1.15.1.2 ActiveMapping

Gets or sets the currently active mapping in the document this MapForceView belongs to.

Signature

ActiveMapping : Mapping

Errors

Error code	Description		
1300	The object is no longer valid.		
1301	Invalid address for the return parameter was specified.		

15.7.1.15.1.3 ActiveMappingName

Gets or sets the currently active mapping by name in the document this MapForceView belongs to.

Signature

ActiveMappingName : String

Errors

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.4 Application

Retrieves the application's top-level object.

Signature

Application : Application

Error code	Description
1300	The object is no longer valid.

Error code	Description	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.5 HighlightMyConnections

This property defines whether connections from the selected item only should be highlighed.

Signature

HighlightMyConnections : Boolean

Errors

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.6 HighlightMyConnectionsRecursively

This property defines if only the connections coming directly or indirectly from the selected item should be highlighted.

Signature

```
HighlightMyConnectionsRecursively : Boolean
```

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.7 Parent

The parent object according to the object model.

Signature

Parent	:	Document

Errors

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.8 ShowItemTypes

This property defines if types of items should be shown in the mapping diagram.

Signature

ShowItemTypes : Boolean

Errors

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.1.9 ShowLibraryInFunctionHeader

This property defines whether the name of the function library should be part of function names.

Signature

ShowLibraryInFunctionHeader : Boolean

Error code	Description
1300	The object is no longer valid.

Error code	Description	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.2 Methods

15.7.1.15.2.1 InsertWSDLCall

Adds a new WSDL call component to the mapping.

Signature

```
InsertWSDLCall(in i_strWSDLFileName:String) -> Void
```

Parameters

Name	Туре	Description
i_strWSDLFileName	String	Specifies the path of the WSDL file to add to the mapping.

Errors

Error code	Description	
1300	The object is no longer valid.	
1301	Invalid address for the return parameter was specified.	

15.7.1.15.2.2 InsertXMLFile (obsolete)

MapForceView.InsertXMLFile is obsolete. Use Mapping.InsertXMLFile instead.

Signature

InsertXMLFile(in i_strFileName:String, in i_strXMLRootName:String) -> Void

Parameters

Name	Туре	Description
i_strFileName	String	
i_strXMLRootName	String	

15.7.1.15.2.3 InsertXMLSchema (obsolete)

MapForceView.InsertXMLSchema is obsolete. Use Mapping.InsertXMLSchema instead.

Signature

InsertXMLSchema(in i_strSchemaFileName:String, in i_strXMLRootName:String) -> Void

Parameters

Name	Туре	Description
i_strSchemaFileName	String	
i_strXMLRootName	String	

15.7.1.15.2.4 InsertXMLSchemaWithSample (obsolete)

MapForceView.InsertXMLSchemaWithSample is obsolete. Use Mapping.InsertXMLFile instead. Notice, Mapping.InsertXMLFile does not require a parameter for passing the root element. The root element is automatically set as the XML file's root element name.

Signature

```
InsertXMLSchemaWithSample(in i_strSchemaFileName:String, in i_strXMLExampleFile:String,
in i_strXMLRootName:String) -> Void
```

Parameters

Name	Туре	Description
i_strSchemaFileName	String	
i_strXMLExampleFile	String	
i_strXMLRootName	String	

15.7.1.16 Mapping

A Mapping object represents a mapping in a document, either the main mapping, or a local user-defined-function mapping.

Properties to navigate the object model:

- Application
- Parent

Mapping properties:

- IsMainMapping
- Name

Components in the mapping:

• Components

Adding items:

- CreateConnection
- InsertFunctionCall
- InsertXMLFile
- InsertXMLSchema
- InsertXMLSchemaInputParameter
- InsertXMLSchemaOutputParameter

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>Components</u>	Read-only. Returns a collection of all components in the current mapping.
<u>IsMainMapping</u>	Read-only. Indicates if the current mapping is the main mapping of the document the mapping is in. True means it is the main mapping. False means it is a user-defined function (UDF).
Name	Read-only. The name of the mapping or user defined-function (UDF).
Parent	Read-only. The parent object according to the object model.

Methods

Name	Description
<u>CreateConnection</u>	Creates a connection between the two supplied datapoints (DatapointFrom & DatapointTo).
	It will fail to do so if the DatapointFrom is not an output-side datapoint, the DatapointTo is not an input-side datapoint, or a connection between these two datapoints already exists.
<u>InsertFunctionCall</u>	Inserts a function call component into the current mapping.

Name	Description
	The specified library and function names indicate the function or user-defined function to be called.
<u>InsertXMLFile</u>	Adds a new XML schema component to the mapping.
	The component's internal structure is determined by the schema referenced in the specified XML file (i_strFileName) or, if the XML file does not reference a schema file, by the separately specified schema file (i_strSchemaFileName).
	If the XML file has a schema file reference, then the parameter i_strSchemaFileName is ignored.
	The root element of the XML file will be used in the component.
	The specified XML file is used as the input sample to evaluate the mapping.
<u>InsertXMLSchema</u>	Adds a new XML schema component to the mapping.
	The component's internal structure is determined by the schema file specified in the first parameter.
	The second parameter defines the root element of this schema if there is more than one candidate.
	If the passed root element is an empty string and more candidates are available, a Select Root Element dialog box will pop up if MapForce is visible. If MapForce is invisible, no dialog box will pop up and only an error is returned.
	No XML input sample is assigned to this component.
<u>InsertXMLSchemaInputParameter</u>	Inserts an XML schema input parameter component into the current mapping.
	The current mapping has to be a user-defined function. Trying to insert it (the schema input parameter) into the main mapping will fail.
InsertXMLSchemaOutputParameter	Inserts an XML schema output parameter component into the current mapping.
	The current mapping has to be a user-defined function. Trying to insert it (the schema output paramter) into the main mapping will fail.

15.7.1.16.1 Properties

15.7.1.16.1.1 Application

Retrieves the application's top-level object.

Signature

	Application	:	Application					
--	-------------	---	-------------	--	--	--	--	--

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.16.1.2 Components

Returns a collection of all components in the current mapping.

Signature

Components : Components

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.16.1.3 IsMainMapping

Indicates if the current mapping is the main mapping of the document the mapping is in.

True means it is the main mapping. **False** means it is a user-defined function (UDF).

Signature

IsMainMapping : Boolean

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.16.1.4 Name

The name of the mapping or user defined-function (UDF).

Signature

Name : String

Errors

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.16.1.5 Parent

The parent object according to the object model.

Signature

Parent : Document

Error code	Description	
1200	The object is no longer valid.	
1201	Invalid address for the return parameter was specified.	

15.7.1.16.2 Methods

15.7.1.16.2.1 CreateConnection

Creates a connection between the two supplied datapoints (DatapointFrom & DatapointTo).

It will fail to do so if the DatapointFrom is not an output-side datapoint, the DatapointTo is not an input-side datapoint, or a connection between these two datapoints already exists.

Signature

CreateConnection (in DatapointFrom: Datapoint, in DatapointTo: Datapoint) -> Connection

Parameters

Name	Туре	Description
DatapointFrom	Datapoint	The datapoint from which the connection is to be created.
DatapointTo	Datapoint	The destination datapoint.

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.
1241	Failed creating the connection.

15.7.1.16.2.2 InsertFunctionCall

Inserts a function call component into the current mapping.

The specified library and function names indicate the function or user-defined function to be called.

Signature

InsertFunctionCall(in strFunctionName:String, in strLibraryName:String) -> Component

Parameters

Name	Туре	Description
strFunctionName	String	The name of the function to be inserted.
strLibraryName	String	The library name of the function to be inserted

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.
1242	Failed creating function call component.

15.7.1.16.2.3 InsertXMLFile

Adds a new XML schema component to the mapping.

The component's internal structure is determined by the schema referenced in the specified XML file (i_strFileName) or, if the XML file does not reference a schema file, by the separately specified schema file (i_strSchemaFileName).

If the XML file has a schema file reference, then the parameter i strSchemaFileName is ignored.

The root element of the XML file will be used in the component.

The specified XML file is used as the input sample to evaluate the mapping.

Signature

InsertXMLFile(in i_strFileName:String, in i_strSchemaFileName:String) -> Component

Parameters

Name	Туре	Description
i_strFileName	String	The path of the instance XML file to add.
i_strSchemaFileName	String	The path of the XML Schema Definition file to add.

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.
1244	Failed creating component.

15.7.1.16.2.4 InsertXMLSchema

Adds a new XML schema component to the mapping.

The component's internal structure is determined by the schema file specified in the first parameter.

The second parameter defines the root element of this schema if there is more than one candidate.

If the passed root element is an empty string and more candidates are available, a **Select Root Element** dialog box will pop up if MapForce is visible. If MapForce is invisible, no dialog box will pop up and only an error is returned.

No XML input sample is assigned to this component.

Signature

InsertXMLSchema(in i_strSchemaFileName:String, in i_strXMLRootName:String) -> Component

Parameters

Name	Туре	Description
i_strSchemaFileName	String	The path of the XML Schema Definition file to add.
i_strXMLRootName	String	The root element of the schema (applicable when the schema has more than one root element).

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.

Error code	Description
1244	Failed creating component.

15.7.1.16.2.5 InsertXMLSchemaInputParameter

Inserts an XML schema input parameter component into the current mapping.

The current mapping has to be a user-defined function. Trying to insert it (the schema input parameter) into the main mapping will fail.

Signature

```
InsertXMLSchemaInputParameter(in strParamName:String, in strSchemaFileName:String, in
strXMLRootElementName:String) -> Component
```

Parameters

Name	Туре	Description
strParamName	String	The name of the input parameter component to create.
strSchemaFileName	String	The path of the XML Schema Definition file to add.
strXMLRootElementName	String	The root element of the schema (applicable when the schema has more than one root element). If the passed root element is an empty string and more candidates are available, a Select Root Element dialog will pop up if MapForce is visible. If MapForce is invisible, no dialog box will pop up and only an error is returned.

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.
1243	Failed creating parameter component.
1245	This operation is not supported for the main mapping.

15.7.1.16.2.6 InsertXMLSchemaOutputParameter

Inserts an XML schema output parameter component into the current mapping.

The current mapping has to be a user-defined function. Trying to insert it (the schema output paramter) into the main mapping will fail.

Signature

```
InsertXMLSchemaOutputParameter(in strParamName:String, in strSchemaFileName:String, in
strXMLRootElementName:String) -> Component
```

Parameters

Name	Туре	Description
strParamName	String	The name of the output parameter component to create.
strSchemaFileName	String	The path of the XML Schema Definition file to add.
strXMLRootElementName	String	The root element of the schema (applicable when the schema has more than one root element). If the passed root element is an empty string and more candidates are available, a Select Root Element dialog will pop up if MapForce is visible. If MapForce is invisible, no dialog box will pop up and only an error is returned.

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.
1240	Changing the document not allowed. It is read-only.
1243	Failed creating parameter component.
1245	This operation is not supported for the main mapping.

15.7.1.17 Mappings

Represents a collection of Mapping objects.

Properties to navigate the object model:

- Application
- Parent

Iterating through the collection:

- Count
- Item

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
Count	Read-only. Retrieves the number of mappings in the collection.
<u>Item</u>	Read-only. Retrieves the mapping at index ${\rm n}$ from the collection. Indices start with 1.
Parent	Read-only. The parent object according to the object model.

15.7.1.17.1 Properties

15.7.1.17.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.17.1.2 Count

Retrieves the number of mappings in the collection.

Signature

Count : Integer

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.17.1.3 Item

Retrieves the mapping at index n from the collection. Indices start with 1.

Signature

Item(in n:Integer) : Mapping

Errors

Error code	Description
1200	The object is no longer valid.
1201	Invalid address for the return parameter was specified.

15.7.1.17.1.4 Parent

The parent object according to the object model.

Signature

Parent : Document

Error code	Description
1200	The object is no longer valid.
Error code	Description
------------	---
1201	Invalid address for the return parameter was specified.

15.7.1.18 Options

This object gives access to all MapForce options available in the **Tools | Options** dialog.

Properties to navigate the object model:

- Application
- Parent

General options:

- ShowLogoOnPrint
- ShowLogoOnStartup
- UseGradientBackground

Options for code generation:

- DefaultOutputEncoding
- DefaultOutputByteOrder
- DefaultOutputByteOrderMark
- XSLTDefaultOutputDirectory
- CodeDefaultOutputDirectory
- CPPSettings_DOMType
- CPPSettings GenerateVC6ProjectFile
- CppSettings_GenerateVSProjectFile
- CPPSettings_LibraryType
- CPPSettings_UseMFC
- CSharpSettings_ProjectType

Properties

Name	Description
Application	Read-only. Retrieves the application's top-level object.
<u>CodeDefaultOutputDirectory</u>	Specifies the target directory where files generated by Document.GenerateCppCode, Document.GenerateJavaCode and Document.GenerateCHashCode are placed.
CPPSettings_DOMType	Specifies the DOM type used by Document.GenerateCppCode.
<u>CPPSettings_GenerateVC6ProjectFil</u> <u>e</u>	Specifies if VisualC++ 6.0 project files should be generated by Document.GenerateCppCode.
<u>CppSettings_GenerateVSProjectFile</u>	Specifies the version of Visual Studio in which project files should be generated by Document.GenerateCppCode.

Name	Description
<u>CPPSettings LibraryType</u>	Specifies the library type used by Document.GenerateCppCode.
<u>CPPSettings_UseMFC</u>	Specifies if MFC support should be used by C++ code generated by Document.GenerateCppCode.
<u>CSharpSettings_ProjectType</u>	Specifies the type of C# project used by Document.GenerateCHashCode.
<u>DefaultOutputByteOrder</u>	Byte order for the file encoding used for output files.
<u>DefaultOutputByteOrderMark</u>	Indicates if a byte order mark (BOM), is to be included in the file encoding of output files.
<u>DefaultOutputEncoding</u>	File encoding used for output files.
<u>GenerateWrapperClasses</u>	Indicates if wrapper classes are also to be generated when generating code.
JavaSettings_ApacheAxisVersion	This property is obsolete.
Parent	Read-only. The parent object according to the object model.
ShowLogoOnPrint	Show or hide the MapForce logo on printed outputs.
ShowLogoOnStartup	Show or hide the MapForce logo on application startup.
UseGradientBackground	Set or retrieve the background color mode for a mapping window.
XSLTDefaultOutputDirectory	Specifies the target directory where files generated by Document.GenerateXSLT are placed.

15.7.1.18.1 Properties

15.7.1.18.1.1 Application

Retrieves the application's top-level object.

Signature

Application : Application

Error code	Description
1400	The object is no longer valid.

Error code	Description
1401	Invalid address for the return parameter was specified.

15.7.1.18.1.2 CodeDefaultOutputDirectory

Specifies the target directory where files generated by Document.GenerateCppCode, Document.GenerateJavaCode and Document.GenerateCHashCode are placed.

Signature

CodeDefaultOutputDirectory : String

Errors

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.

15.7.1.18.1.3 CPPSettings_DOMType

Specifies the DOM type used by Document.GenerateCppCode.

Signature

CPPSettings_DOMType : <u>ENUMDOMType</u>

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.
1402	The parameter value is out of range.
1403	The parameter value is not available anymore.

15.7.1.18.1.4 CPPSettings_GenerateVC6ProjectFile (obsolete)

Specifies if VisualC++ 6.0 project files should be generated by Document.GenerateCppCode.

Signature

CPPSettings_GenerateVC6ProjectFile : Boolean

Errors

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.
1402	The parameter value is out of range.
1403	The parameter value is not available anymore.

15.7.1.18.1.5 CppSettings_GenerateVSProjectFile

Specifies the version of Visual Studio in which project files should be generated by ${\tt Document.GenerateCppCode}.$

Signature

CppSettings_GenerateVSProjectFile : ENUMProjectType

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.
1402	The parameter value is out of range.
1403	The parameter value is not available anymore.

15.7.1.18.1.6 CPPSettings_LibraryType

Specifies the library type used by Document.GenerateCppCode.

Signature

CPPSettings_LibraryType : ENUMLibType

Errors

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.

15.7.1.18.1.7 CPPSettings_UseMFC

Specifies if MFC support should be used by C++ code generated by Document.GenerateCppCode.

Signature

CPPSettings_UseMFC : Boolean

Errors

Error code	Description
1400	The object is no longer valid.
1401	Invalid address for the return parameter was specified.

15.7.1.18.1.8 CSharpSettings_ProjectType

Specifies the type of C# project used by Document.GenerateCHashCode.

Signature

CSharpSettings_ProjectType : <u>ENUMProjectType</u>

Error code	Description
1400	The object is no longer valid.

Error code	Description	
1401	Invalid address for the return parameter was specified.	
1402	The parameter value is out of range.	
1403	The parameter value is not available anymore.	

15.7.1.18.1.9 DefaultOutputByteOrder

Byte order for the file encoding used for output files.

Signature

DefaultOutputByteOrder : String

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.10 DefaultOutputByteOrderMark

Indicates if a byte order mark (BOM), is to be included in the file encoding of output files.

Signature

DefaultOutputByteOrderMark : Boolean

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.11 DefaultOutputEncoding

File encoding used for output files.

Signature

DefaultOutputEncoding : String

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.12 GenerateWrapperClasses

Indicates if wrapper classes are also to be generated when generating code.

Signature

GenerateWrapperClasses : Boolean

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.13 JavaSettings_ApacheAxisVersion (obsolete)

This property is obsolete.

Signature

 ${\tt JavaSettings_ApacheAxisVersion}\ :\ \underline{{\tt ENUMApacheAxisVersion}}$

15.7.1.18.1.14 Parent

The parent object according to the object model.

Signature

Parent : Application

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.15 ShowLogoOnPrint

Show or hide the MapForce logo on printed outputs.

Signature

ShowLogoOnPrint : Boolean

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.16 ShowLogoOnStartup

Show or hide the MapForce logo on application startup.

Signature

ShowLogoOnStartup : Boolean

Error code	Description	
1400	The object is no longer valid.	

Error code	Description
1401	Invalid address for the return parameter was specified.

15.7.1.18.1.17 UseGradientBackground

Set or retrieve the background color mode for a mapping window.

Signature

UseGradientBackground : Boolean

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.18.1.18 XSLTDefaultOutputDirectory

Specifies the target directory where files generated by Document.GenerateXSLT are placed.

Signature

XSLTDefaultOutputDirectory : String

Errors

Error code	Description	
1400	The object is no longer valid.	
1401	Invalid address for the return parameter was specified.	

15.7.1.19 Project

A Project object represents a project and its tree of project items in MapForce.

Properties to navigate the object model:

- Application
- Parent

File handling:

- FullName
- Name
- Path
- Saved
- Save
- Close

Project tree navigation:

- Count
- Item
- _NewEnum

Project tree manipulation:

- AddActiveFile
- AddFile
- InsertWebService (Enterprise edition only)
- CreateFolder

Code generation:

- Output_Folder
- Output_Language
- Output_TextEncoding
- Java_BasePackageName
- GenerateCode
- GenerateCodeEx
- GenerateCodeIn
- GenerateCodeInEx

For examples of how to use the properties and methods listed above, see <u>Example: Project Tasks</u>. Note that, in order to perform operations that involve Web services, MapForce Enterprise edition is required.

Properties

Name	Description
<u>NewEnum</u>	Read-only. This property supports language-specific standard enumeration.
Application	Read-only. Retrieves the top-level application object.
<u>Count</u>	Read-only. Retrieves number of children of the project's root item. For examples, see Item or _NewEnum
<u>FullName</u>	Path and name of the project file.
Item	Read-only.

Name	Description
	Returns the child at <i>n</i> position of the project's root. The index is 1-based (the first index is 1). The largest valid index is <code>Count</code> . For an alternative, see <code>_NewEnum</code> .
<u>Java_BasePackageName</u>	Sets or gets the base package name of the Java packages that will be generated. This property is used only when generating Java code.
Name	Read-only. Name of the project file without file path.
<u>Output_Folder</u>	Sets or gets the default output folder used with GenerateCode and GenerateCodeIn. Project items can overwrite this value in their CodeGenSettings_OutputFolder property, when CodeGenSettings_UseDefault is set to false.
<u>Output_Language</u>	Sets or gets the default language for code generation when using GenerateCode. Project items can overwrite this value in their CodeGenSettings_OutputLanguage property, when CodeGenSettings_UseDefault is set to false.
Output_TextEncoding	Sets or gets the text encoding used when generating XML- based code.
Parent	Read-only. The parent object according to the object model.
Path	Read-only. Path of the project file without name.
Saved	Read-only. True if the project was not modified since the last Save operation, false otherwise.

Methods

Name	Description
AddActiveFile	Adds the currently open document to the mapping folder of the project's root.
<u>AddFile</u>	Adds the specified document to the mapping folder of the project's root.
<u>Close</u>	Closes the project without saving.
<u>CreateFolder</u>	Creates a new folder as a child of the project's root item.
<u>GenerateCode</u>	Generates code for all project items of the project. The code language and output location is determined by properties of the project and project items.

Name	Description
GenerateCodeEx	Generates code for all project items of the project. The code language and output location are determined by properties of the project and project items. An object that can be used to iterate through all messages issued by the code generation process is returned. These messages are the same as those shown in the Messages window of MapForce.
<u>GenerateCodeIn</u>	Generates code for all project items of the project in the specified language. The output location is determined by properties of the project and project items.
<u>GenerateCodeInEx</u>	Generates code for all project items of the project in the specified language. The output location is determined by properties of the project and project items. Returns an object that can be used to iterate through all messages issued by the code generation process. These messages are the same as those shown in the Messages window of MapForce.
<u>InsertWebService</u>	Inserts a new Web service project into the project's Web service folder. If i_bGenerateMappings is true, initial mapping documents for all ports get generated automatically.
Save	Saves the project to the file defined by FullName.

Events

Name	Description
<u>OnProjectClosed</u>	This event is triggered when the project is closed. The project object passed into the event handler should not be accessed. The corresponding open event is Application.OnProjectOpened.

15.7.1.19.1 Properties

15.7.1.19.1.1 _*NewEnum*

This property supports language-specific standard enumeration.

Signature

_NewEnum :

Errors

Error code	Description
1500	The object is no longer valid.

Examples

```
// ------
// JScript sample - enumeration of a project's project items.
function AllChildrenOfProjectRoot()
{
    objProject = objMapForce.ActiveProject;
    if ( objProject != null )
    {
    for ( objProjectIter = new Enumerator(objProject); ! objProjectIter.atEnd(); objProject
Iter.moveNext() )
        {
            objProjectItem = objProjectIter.item();
            // do something with project item here
        }
    }
}
```

```
// -----
// JScript sample - iterate all project items, depth first.
function IterateProjectItemsRec(objProjectItemIter)
{
    while ( ! objProjectItemIter.atEnd() )
    {
        objProjectItem = objProjectItemIter.item();
        // do something with project item here
        IterateProjectItemsRec( new Enumerator(objProjectItem) );
        objProjectItemIter.moveNext();
    }
}
function IterateAllProjectItems()
{
```

```
objProject = objMapForce.ActiveProject;
if ( objProject != null )
{
    IterateProjectItemsRec( new Enumerator(objProject) );
}
```

15.7.1.19.1.2 Application

Retrieves the top-level application object.

Signature

}

Application : Application

Errors

Error code	Description
1500	The object is no longer valid.
1501	Invalid address for the return parameter was specified.

15.7.1.19.1.3 Count

Retrieves number of children of the project's root item. For examples, see Item or _NewEnum

Signature

Count : Integer

Error code	Description
1500	The object is no longer valid.

15.7.1.19.1.4 FullName

Path and name of the project file.

Signature

.11Name : String	Name : String
------------------	---------------

Errors

Error code	Description
1500	The object is no longer valid.
1501	Invalid address for the return parameter was specified.

15.7.1.19.1.5 Item

Returns the child at *n* position of the project's root. The index is 1-based (the first index is 1). The largest valid index is Count. For an alternative, see _NewEnum.

Signature

Item(in n:Integer) : ProjectItem

Errors

Error code	Description
1500	The object is no longer valid.

Examples

```
// -----
// JScript code snippet - enumerate children using Count and Item.
for( nItemIndex = 1; nItemIndex <= objProject.Count; nItemIndex++ )
{
    objProjectItem = objProject.Item(nItemIndex);
    // do something with project item here
}</pre>
```

15.7.1.19.1.6 Java_BasePackageName

Sets or gets the base package name of the Java packages that will be generated. This property is used only when generating Java code.

Signature

Java_BasePackageName : String

Errors

Error code	Description
1500	The object is no longer valid.
1501	Invalid package name specified. Invalid address for the return parameter was specified.

15.7.1.19.1.7 Name

Name of the project file without file path.

Signature

Name : String

Errors

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid address for the return parameter was specified.	

15.7.1.19.1.8 Output_Folder

Sets or gets the default output folder used with GenerateCode and GenerateCodeIn. Project items can overwrite this value in their CodeGenSettings_OutputFolder property, when CodeGenSettings_UseDefault is set to false.

Signature

Output Folder : String	utput I	der : Strin

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid folder name specified. Invalid address for the return parameter was specified.	

15.7.1.19.1.9 Output_Language

Sets or gets the default language for code generation when using GenerateCode. Project items can overwrite this value in their CodeGenSettings_OutputLanguage property, when CodeGenSettings_UseDefault is set to false.

Signature

Errors

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid language specified. Invalid address for the return parameter was specified.	

15.7.1.19.1.10 Output_TextEncoding

Sets or gets the text encoding used when generating XML-based code.

Signature

Output_TextEncoding : String

Errors

Error code	Description
1500	The object is no longer valid.
1501	Invalid text encoding specified. Invalid address for the return parameter was specified.

15.7.1.19.1.11 Parent

The parent object according to the object model.

Signature

Parent : Application

Errors

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid address for the return parameter was specified.	

15.7.1.19.1.12 Path

Path of the project file without name.

Signature

Path : String

Errors

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid address for the return parameter was specified.	

15.7.1.19.1.13 Saved

True if the project was not modified since the last save operation, false otherwise.

Signature

Saved : Boolean

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid address for the return parameter was specified.	

15.7.1.19.2 Methods

15.7.1.19.2.1 AddActiveFile

Adds the currently open document to the mapping folder of the project's root.

Signature

AddActiveFile() -> **ProjectItem**

Errors

Error code	Description	
1500	The object is no longer valid.	
1501	Invalid address for the return parameter was specified.	
1503	No active document is available.	
1504	Active documents needs to be given a path name before it can be added to the project.	
1705	Mapping could not be assigned to project. Maybe it is already contained in the target folder.	

15.7.1.19.2.2 AddFile

Adds the specified document to the mapping folder of the project's root.

Signature

```
AddFile(in i_strFileName:String) -> ProjectItem
```

Parameters

Name	Туре	Description
i_strFileName	String	Specifies the path of the document to add.

Error code	Description
1500	The object is no longer valid.

Error code	Description
1501	Invalid address for the return parameter was specified.
1705	Mapping could not be assigned to project. The file does not exist or is not a MapForce mapping. Maybe the file is already assigned to the target folder.

15.7.1.19.2.3 Close

Closes the project without saving.

Signature

Close() -> Void

Errors

Error code	Description
1500	The object is no longer valid.

15.7.1.19.2.4 CreateFolder

Creates a new folder as a child of the project's root item.

Signature

```
CreateFolder(in i_strFolderName:String) -> ProjectItem
```

Parameters

Name	Туре	Description
i_strFolderName	String	The name of the folder to create.

Error code	Description
1500	The object is no longer valid.
1501	Invalid folder name or invalid address for the return parameter was specified.

15.7.1.19.2.5 GenerateCode

Generates code for all project items of the project. The code language and output location is determined by properties of the project and project items.

Signature

GenerateCode() -> **Void**

Errors

Error code	Description
1500	The object is no longer valid.
1706	Error during code generation.

15.7.1.19.2.6 GenerateCodeEx

Generates code for all project items of the project. The code language and output location are determined by properties of the project and project items. An object that can be used to iterate through all messages issued by the code generation process is returned. These messages are the same as those shown in the Messages window of MapForce.

Signature

GenerateCodeEx() -> ErrorMarkers

Errors

Error code	Description
1500	The object is no longer valid.
1501	Invalid address for the return parameter was specified.
1706	Error during code generation.

15.7.1.19.2.7 GenerateCodeIn

Generates code for all project items of the project in the specified language. The output location is determined by properties of the project and project items.

Signature

GenerateCodeIn(in i_nLanguage:ENUMProgrammingLanguage) -> Void

Parameters

Name	Туре	Description
i_nLanguage	<u>ENUMProgrammingLanguage</u>	Specifies the programming language in which code should be generated.

Errors

Error code	Description
1500	The object is no longer valid.
1706	Error during code generation.

15.7.1.19.2.8 GenerateCodeInEx

Generates code for all project items of the project in the specified language. The output location is determined by properties of the project and project items. Returns an object that can be used to iterate through all messages issued by the code generation process. These messages are the same as those shown in the **Messages** window of MapForce.

Signature

GenerateCodeInEx(in i_nLanguage:ENUMProgrammingLanguage) -> ErrorMarkers

Parameters

Name	Туре	Description
i_nLanguage	<u>ENUMProgrammingLanguage</u>	Specifies the programming language in which code should be generated.

Error code	Description
1500	The object is no longer valid.
1501	Invalid address for the return parameter was specified.
1706	Error during code generation.

15.7.1.19.2.9 InsertWebService

Inserts a new Web service project into the project's Web service folder. If **i_bGenerateMappings** is true, initial mapping documents for all ports get generated automatically.

Signature

```
InsertWebService(in i_strWSDLFile:String, in i_strService:String, in i_strPort:String,
in i_bGenerateMappings:Boolean) -> ProjectItem
```

Parameters

Name	Туре	Description
i_strWSDLFile	String	Specifies the path of the WSDL file to add.
i_strService	String	Specifies the name of the Web service to add.
i_strPort	String	Specifies the port of the Web service to add.
i_bGenerateMappings	Boolean	If this parameter is true, initial mapping documents for all ports get generated automatically.

Errors

Error code	Description
1500	The object is no longer valid.
1501	WSDL file can not be found or is invalid. Service or port names are invalid. Invalid address for the return parameter was specified.
1503	Operation not supported by current edition.

15.7.1.19.2.10 Save

Saves the project to the file defined by FullName.

Signature

Save() -> Void

Errors

Error code	Description
1500	The object is no longer valid.
1502	Can't save to file.

15.7.1.19.3 Events

15.7.1.19.3.1 OnProjectClosed

This event is triggered when the project is closed. The project object passed into the event handler should not be accessed. The corresponding open event is <code>Application.OnProjectOpened</code>.

Signature

```
OnProjectClosed(in i_ipProject:Project) : Void
```

15.7.1.20 ProjectItem

A ProjectItem object represents one item in a project tree.

Properties to navigate the object model:

- Application
- Parent

Project tree navigation:

- Count
- Item
- _NewEnum

Project item properties:

- Kind
- Name
- WSDLFile (only available to Web service project items)
- QualifiedName (only available to Web service project items)

Project tree manipulation:

- AddActiveFile (only available to folder items)
- AddFile (only available to folder items)
- CreateFolder (only available to folder items)
- CreateMappingForProject (only available to Web service operations)

The MapForce API

• Remove

Document access:

• Open (only available to mapping items and Web service operations)

Code-generation:

- CodeGenSettings_UseDefault
- CodeGenSettings_OutputFolder
- CodeGenSettings_Language
- GenerateCode
- GenerateCodeEx
- GenerateCodeIn
- GenerateCodeInEx

For examples of how to use the properties and methods listed above, see <u>Example: Project Tasks</u>. For operations with Web services, the MapForce Enterprise edition is required.

Properties

Name	Description
<u>NewEnum</u>	Read-only. This property supports language specific standard enumeration. For examples, see Project.Item or ProjectNewEnum.
Application	Read-only. Retrieves the top-level application object.
CodeGenSettings_Language	Gets or sets the language to be used with GenerateCode or Project.GenerateCode. This property is consulted only if CodeGenSettings_UseDefault is set to false.
CodeGenSettings_OutputFolder	Gets or sets the output directory to be used with GenerateCode, GenerateCodeIn, Project.GenerateCode Or Project.GenerateCodeIn. This property is consulted only if CodeGenSettings_UseDefault is set to false.
<u>CodeGenSettings_UseDefault</u>	Gets or sets whether output directory and code language are used as defined by either (a) the parent folders, or (b) the project root. This property is used with calls to GenerateCode, GenerateCodeIn, Project.GenerateCode and Project.GenerateCodeIn. If this property is set to false, the values of CodeGenSettings_OutputFolder and CodeGenSettings_Language are used to generate code for this project item.
Count	Read-only. Retrieves the number of children of this project item. See also Item. For examples, see Project.Item or ProjectNewEnum.
Item	Read-only.

Name	Description
	Returns the child at n position of this project item. The index is 1-based (the first index is 1). The largest valid index is ProjectItem.Count. For an alternative, see ProjectItemNewEnum. For examples, see Project.Item or ProjectNewEnum.
<u>Kind</u>	Read-only. Retrieves the kind of the project item. Availability of some properties and the applicability of certain methods is restricted to specific kinds of project items. The description of all methods and properties contains information about these restrictions.
<u>Name</u>	Retrieves or sets the name of a project item. The name of most items is read-only. Exceptions are user-created folders, the names of which can be altered after creation.
<u>Parent</u>	Read-only. Retrieves the project that this item is a child of. Has the same effect as Application.ActiveProject.
<u>QualifiedName</u>	Read-only. Retrieves the qualified name of a Web service item.
WSDLFile	Read-only. Retrieves the file name of the WSDL file defining the Web service that hosts the current project item.

Methods

Name	Description
<u>AddActiveFile</u>	Adds the currently active document to this project item if it is a valid child. Otherwise, the document is added to the Mapping Folder of the project's root.
<u>AddFile</u>	Adds the specified document to this project item if it is a valid child. Otherwise, the document is added to the Mapping Folder of the project's root.
<u>CreateFolder</u>	Creates a new folder as a child of this project item.
<u>CreateMappingForProject</u>	Creates an initial mapping document for a Web service operation and saves it to i_strFileName . When using Project.InsertWebService you can set the i_bGenerateMappings flag to let MapForce automatically generate initial mappings for all ports.
<u>GenerateCode</u>	Generates code for this project item and its children. The code language and output location is determined by CodeGenSettings_UseDefault, CodeGenSettings_Language and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code- generation.

Name	Description
<u>GenerateCodeEx</u>	Generates code for this project item and its children. The code language and output location are determined by CodeGenSettings_UseDefault, CodeGenSettings_Language and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code- generation.
<u>GenerateCodeIn</u>	Generates code for the project item and its children in the specified language. The output location is determined by CodeGenSettings_UseDefault and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code-generation.
<u>GenerateCodeInEx</u>	Generates code for the project item and its children in the specified language. The output location is determined by CodeGenSettings_UseDefault and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code-generation. An object that can be used to iterate through all messages issued by the code generation process is returned. These messages are the same as those shown in the Messages window of MapForce.
<u>Open</u>	Opens the project item as a document or makes the corresponding document the active one, if it is already open. The project item must be a MapForce mapping or, for Enterprise edition only, Web service operation.
Remove	Remove this project item and all its children from the project tree.

Events

Name	Description
<u>OnModifiedFlagChanged</u>	Occurs when the ProjectItem's modification status changes.
<u>OnProjectClosed</u>	This event is triggered when the project is closed. The project object passed into the event handler should not be accessed. The corresponding open event is Application.OnProjectOpened.

15.7.1.20.1 Properties

15.7.1.20.1.1 _NewEnum

This property supports language specific standard enumeration. For examples, see Project.Item or Project._NewEnum.

Signature

NewEnum : IUnknown

Errors

Error code	Description
1700	The object is no longer valid.

15.7.1.20.1.2 Application

Retrieves the top-level application object.

Signature

Application : Application

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.

15.7.1.20.1.3 CodeGenSettings_Language

Gets or sets the language to be used with GenerateCode or Project.GenerateCode. This property is consulted only if CodeGenSettings_UseDefault is set to false.

Signature

```
CodeGenSettings_Language : <u>ENUMProgrammingLanguage</u>
```

Error code	Description
1700	The object is no longer valid.
1701	Invalid language or invalid address for the return parameter was specified.

15.7.1.20.1.4 CodeGenSettings_OutputFolder

Gets or sets the output directory to be used with GenerateCode, GenerateCodeIn, Project.GenerateCode or Project.GenerateCodeIn. This property is consulted only if CodeGenSettings_UseDefault is set to false.

Signature

CodeGenSettings_OutputFolder : String

Errors

Error code	Description
1700	The object is no longer valid.
1701	An invalid output folder or an invalid address for the return parameter was specified.

15.7.1.20.1.5 CodeGenSettings_UseDefault

Gets or sets whether output directory and code language are used as defined by either (a) the parent folders, or (b) the project root. This property is used with calls to GenerateCode, GenerateCodeIn, Project.GenerateCode and Project.GenerateCodeIn. If this property is set to false, the values of CodeGenSettings_OutputFolder and CodeGenSettings_Language are used to generate code for this project item.

Signature

CodeGenSettings UseDefault : Boolean

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.

15.7.1.20.1.6 Count

Retrieves the number of children of this project item. See also Item. For examples, see Project.Item or Project._NewEnum.

Signature

Count	:	Integer
-------	---	---------

Errors

Error code	Description
1700	The object is no longer valid.

15.7.1.20.1.7 Item

Returns the child at n position of this project item. The index is 1-based (the first index is 1). The largest valid index is ProjectItem.Count. For an alternative, see ProjectItem.NewEnum. For examples, see Project.Item Or Project.NewEnum.

Signature

Item(in n:Integer) : ProjectItem

Errors

Error code	Description
1700	The object is no longer valid.

15.7.1.20.1.8 Kind

Retrieves the kind of the project item. Availability of some properties and the applicability of certain methods is restricted to specific kinds of project items. The description of all methods and properties contains information about these restrictions.

Signature

Kind : <u>ENUMProjectItemType</u>

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.

15.7.1.20.1.9 Name

Retrieves or sets the name of a project item. The name of most items is read-only. Exceptions are usercreated folders, the names of which can be altered after creation.

Signature

Name : String

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.
1702	Project item does not allow to alter its name.

15.7.1.20.1.10 Parent

Retrieves the project that this item is a child of. Has the same effect as Application.ActiveProject.

Signature

Parent : Project

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.

15.7.1.20.1.11 QualifiedName

Retrieves the qualified name of a Web service item.

Signature

QualifiedName : String

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.

Error code	Description
1702	The project item is not a part of a Web service.

15.7.1.20.1.12 WSDLFile

Retrieves the file name of the WSDL file defining the Web service that hosts the current project item.

Signature

WSDLFile : String	WSDLFile : String	
-------------------	-------------------	--

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.
1702	The project item is not a part of a Web service.

15.7.1.20.2 Methods

15.7.1.20.2.1 AddActiveFile

Adds the currently active document to this project item if it is a valid child. Otherwise, the document is added to the Mapping Folder of the project's root.

Signature

```
AddActiveFile() -> ProjectItem
```

Error code	Description
1700	The object is no longer valid.
1701	The file name is empty. Invalid address for the return parameter was specified.
1703	No active document is available.
1704	Active documents needs to be given a path name before it can be added to the project.

Error code	Description
1705	Mapping could not be assigned to project. The file does not exist or is not a MapForce mapping. Maybe the file is already assigned to the target folder.

15.7.1.20.2.2 AddFile

Adds the specified document to this project item if it is a valid child. Otherwise, the document is added to the Mapping Folder of the project's root.

Signature

```
AddFile(in i_strFilePath:String) -> ProjectItem
```

Parameters

Name	Туре	Description
i_strFilePath	String	The path of the document to add.

Errors

Error code	Description
1700	The object is no longer valid.
1701	The file name is empty. Invalid address for the return parameter was specified.
1705	Mapping could not be assigned to project. The file does not exist or is not a MapForce mapping. Maybe the file is already assigned to the target folder.

15.7.1.20.2.3 CreateFolder

Creates a new folder as a child of this project item.

Signature

CreateFolder(in i_strFolderName:String) -> ProjectItem

Parameters

Name	Туре	Description
i_strFolderName	String	The name of the folder to create.

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid folder name or invalid address for the return parameter was specified.
1702	The project item does not support children.

15.7.1.20.2.4 CreateMappingForProject

Creates an initial mapping document for a Web service operation and saves it to **i_strFileName**. When using Project.InsertWebService you can set the **i_bGenerateMappings** flag to let MapForce automatically generate initial mappings for all ports.

Signature

CreateMappingForProject(in i strFileName:String) -> ProjectItem

Parameters

Name	Туре	Description
i_strFileName	String	Specifies the path where the mapping should be saved.

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.
1707	Cannot create new mapping. The project item does not support auto-creation of initial mappings or a mapping already exists.
1708	Operation not supported in current edition.

15.7.1.20.2.5 GenerateCode

Generates code for this project item and its children. The code language and output location is determined by CodeGenSettings_UseDefault, CodeGenSettings_Language and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code-generation.

Signature

GenerateCode() -> Void

Errors

Error code	Description
1700	The object is no longer valid.
1706	Error during code generation.

15.7.1.20.2.6 GenerateCodeEx

Generates code for this project item and its children. The code language and output location are determined by CodeGenSettings_UseDefault, CodeGenSettings_Language and CodeGenSettings_OutputFolder. Children of this project item can have their own property settings related to code-generation.

Signature

GenerateCodeEx() -> ErrorMarkers

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid address for the return parameter was specified.
1706	Error during code generation.

15.7.1.20.2.7 GenerateCodeIn

Generates code for the project item and its children in the specified language. The output location is determined by <code>CodeGenSettings_UseDefault</code> and <code>CodeGenSettings_OutputFolder</code>. Children of this project item can have their own property settings related to code-generation.

Signature

GenerateCodeIn(in i_nLanguage:ENUMProgrammingLanguage) -> Void

Parameters

Name	Туре	Description
i_nLanguage	ENUMProgrammingLanguage	Specifies the programming language for code generation.

Errors

Error code	Description
1700	The object is no longer valid.
1701	Invalid language specified.
1706	Error during code generation.

15.7.1.20.2.8 GenerateCodeInEx

Generates code for the project item and its children in the specified language. The output location is determined by <code>CodeGenSettings_UseDefault</code> and <code>CodeGenSettings_OutputFolder</code>. Children of this project item can have their own property settings related to code-generation.

An object that can be used to iterate through all messages issued by the code generation process is returned. These messages are the same as those shown in the Messages window of MapForce.

Signature

GenerateCodeInEx(in i_nLanguage:ENUMProgrammingLanguage) -> ErrorMarkers

Parameters

Name	Туре	Description
i_nLanguage	<u>ENUMProgrammingLanguage</u>	Specifies the programming language for code generation.

Error code	Description
1700	The object is no longer valid.
1701	Invalid language specified or invalid address for the return parameter was specified.
1706	Error during code generation.
15.7.1.20.2.9 Open

Opens the project item as a document or makes the corresponding document the active one, if it is already open. The project item must be a MapForce mapping or, for Enterprise edition only, Web service operation.

Signature

Open() -> Document		
--------------------	--	--

Errors

Error code	Description	
1700 The object is no longer valid.		
1701	Invalid address for the return parameter was specified.	
1702	The project item does not refer to a MapForce mapping file.	
1708	Operation not supported in current edition.	

15.7.1.20.2.10 Remove

Remove this project item and all its children from the project tree.

Signature

Remove() -> Void

Errors

Error code	Description
1700	The object is no longer valid.

15.7.1.20.3 Events

15.7.1.20.3.1 OnModifiedFlagChanged

Occurs when the ProjectItem's modification status changes.

Signature

OnModifiedFlagChanged(in i_blsModified:Boolean) : Void

15.7.1.20.3.2 OnProjectClosed

This event is triggered when the project is closed. The project object passed into the event handler should not be accessed. The corresponding open event is Application.OnProjectOpened.

Signature

```
OnProjectClosed(in i_ipProject:Project) : Void
```

15.7.2 Enumerations

15.7.2.1 ENUMApacheAxisVersion (obsolete)

This enumeration type is obsolete.

Members

eApacheAxisVersion_Axis = 1

eApacheAxisVersion_Axis2 = 2

15.7.2.2 ENUMApplicationStatus

Enumeration values to indicate the status of the application.

Members

eApplicationRunning = 0

eApplicationAfterLicenseCheck = 1

eApplicationBeforeLicenseCheck = 2

eApplicationConcurrentLicenseCheckFailed = 3

eApplicationProcessingCommandLine = 4

15.7.2.3 ENUMAppOutputLine_Severity

Enumeration values to identify the severity of an AppOutputLine.

Members

eSeverity_Undefined = -1

- eSeverity_Info = 0
- eSeverity_Warning = 1
- eSeverity_Error = 2
- eSeverity_CriticalError = 3
- eSeverity_Success = 4
- eSeverity_Summary = 5
- eSeverity_Progress = 6
- eSeverity_DataEdit = 7
- eSeverity_ParserInfo = 8
- eSeverity_PossibleInconsistencyWarning = 9
- eSeverity_Message = 10
- eSeverity_Document = 11
- eSeverity_Rest = 12
- eSeverity_NoSelect = 13
- eSeverity_Select = 14
- eSeverity_Autoinsertion = 15
- eSeverity_GlobalResources_DefaultWarning = 16
- eSeverity_XPath_Styles_Changed = 17
- eSeverity_XPath_Styles_Unchanged = 18
- eSeverity_XPath_Styles_Skipped = 19
- eSeverity_XPath_ComboBox_Values_Changed = 20
- eSeverity_XPath_ComboBox_Values_Unchanged = 21
- eSeverity_XPath_ComboBox_Values_Skipped = 22
- eSeverity_XPath_Assertions_Changed = 23
- eSeverity_XPath_Assertions_Unchanged = 24
- eSeverity_XPath_Assertions_Skipped = 25

15.7.2.4 ENUMAppOutputLine_TextDecoration

Enumeration values for the different kinds of text decoration of an AppOutputLine.

Members

eTextDecorationDefault = 0 eTextDecorationBold = 1 eTextDecorationDebugValues = 2 eTextDecorationDB_ObjectName = 3 eTextDecorationDB_ObjectLink = 4 eTextDecorationDB_ObjectKind = 5 eTextDecorationDB_TimeoutValue = 6 eTextDecorationFind_MatchingString = 7 eTextDecorationValidation_Speclink = 8 eTextDecorationValidation_ErrorPosition = 9 eTextDecorationValidation_UnkownParam = 10

15.7.2.5 ENUMCodeGenErrorLevel

Enumeration values to identify severity of code generation messages.

Members

eCodeGenErrorLevel_Information = 0

eCodeGenErrorLevel_Warning = 1

eCodeGenErrorLevel_Error = 2

eCodeGenErrorLevel_Undefined = 3

15.7.2.6 ENUMComponentDatapointSide

Enumeration values to indicate the side of a datapoint on its component. See also Component.GetRootDatapoint.

Members

eDatapointSideInput = 0

eDatapointSideOutput = 1

15.7.2.7 ENUMComponentSubType

Enumeration values to indicate component sub types.

Members

eComponentSubType_None = 0 eComponentSubType_Text_EDI = 1 eComponentSubType_Text_Flex = 2 eComponentSubType_Text_CSVFLF = 3

15.7.2.8 ENUMComponentType

Enumeration values to indicate component types.

Members

- eComponentType_Unknown = 0 eComponentType_XML = 1 eComponentType_DB = 2
- eComponentType Text = 3
- eComponentType Excel = 4
- eComponentType_WSDL = 5
- eComponentType_XBRL = 6
- eComponentType_Input = 7
- eComponentType_JSON = 8

15.7.2.9 ENUMComponentUsageKind

Enumeration values to indicate component usage kind.

Members

eComponentUsageKind_Unknown = 0

eComponentUsageKind_Instance = 1

eComponentUsageKind_Input = 2

eComponentUsageKind_Output = 3

eComponentUsageKind_Variable = 4

eComponentUsageKind_String = 5

15.7.2.10 ENUMConnectionType

Enumeration values to indicate the type of a connection. See also Connection.ConnectionType.

Members

eConnectionTypeTargetDriven = 0 eConnectionTypeSourceDriven = 1 eConnectionTypeCopyAll = 2

15.7.2.11 ENUMDOMType

Enumeration values to specify the DOM type used by generated C++ mapping code.

NOTE: The value <code>eDOMType_xerces</code> is obsolete. <code>eDOMType_xerces3</code> indicates Xerces 3.x usage. Obsolete in this context means that this value is not supported and should not be used.

Members

eDOMType_xerces = 1 (obsolete) eDOMType_xerces3 = 2 eDOMType_msxml6 = 3

15.7.2.12 ENUMLibType

Enumeration values to specify the library type used by the generated C++ mapping code.

Members

eLibType_static = 0 eLibType_dll = 1

15.7.2.13 ENUMProgrammingLanguage

Enumeration values to select a programming language.

Members

eUndefinedLanguage = -1 eJava = 0 eCpp = 1 eCSharp = 2 eXSLT = 3 eXSLT2 = 4 eXQuery = 5 eXSLT3 = 6

15.7.2.14 ENUMProjectItemType

Enumeration to identify the different kinds of project items that can be children of Project or folder-like ProjectItems. See also ProjectItem.Kind.

Members

eProjectItemType_MappingFolder = 0 eProjectItemType_Mapping = 1 eProjectItemType_WebServiceFolder = 2 eProjectItemType_WebServiceRoot = 3 eProjectItemType_WebServiceService = 4 eProjectItemType_WebServicePort = 5 eProjectItemType_WebServiceOperation = 6 eProjectItemType_ExternalFolder = 7 eProjectItemType_LibraryFolder = 8 eProjectItemType_ResourceFolder = 9 eProjectItemType_VirtualFolder = 10 eProjectItemType_Count = 11 eProjectItemType_Invalid = -1

15.7.2.15 ENUMProjectType

Enumeration values to select a project type for generated C# and C++ mapping code.

Members

eVisualStudio2010Project = 6 eVisualStudio2013Project = 7 eVisualStudio2015Project = 8 eVisualStudio2017Project = 9 eVisualStudio2019Project = 10 eDotNetCore3_1 = 11 eDotNet5_0 = 12

15.7.2.16 ENUMSearchDatapointFlags

Enumeration values used as bit-flags; to be used as combination of flags when searching for a datapoint. See also GetChild.

Members

eSearchDatapointElement = 1

eSearchDatapointAttribute = 2

15.7.2.17 ENUMViewMode

Enumeration values to select a MapForce view.

Members

eMapForceView = 0

eXSLView = 1

eOutputView = 2

16 ActiveX Integration

このセクションで説明されている MapForce ユーザーインターフェイスとActiveX コントロール機能を使用するアプリケーションに統合することができます。 ActiveX 技術により C++、C#、VB.NET、HTML などの統合のため ゴ使用される広範囲の言語が有効化されます (HTML 内に統合されている ActiveX コンポーネントは、Microsoft Internet Explorer バージョンと ActiveX をサポートすると共に作動する必要があります)。全てのコンポーネントはフル OLE コントロールです。 Java への統合はラッパークラスにより与えられます。

ActiveX コトロールを、カスタムコードに統合するコよ MapForce Integration Package をインストールしてくたさい(次を参照してくたさい: <u>http://www.altova.com/ide_integration.html)</u>。すべてのMapForce を最初に、そして、MapForce Integration Package そして他の必要条件を言語とプラオフォームコ応じてインストールしてくたさい(次を参照してくたさい: <u>必要</u>条件)。

統合の2つの異なるレベリ間を柔軟に選択することができます:アプリケーションレベルビギュメントレベル

アプリケーションレベルの統合は、(メニュー、ツール・ー、ペインを含む) MapForce のフルインターフェイスを ActiveX エトロールをカスタ ムアプリケーションとして埋め込むことを意味します。例えば、多くのシナリオでは、カスタムアプリケーションを、MapForce グラフィカルユーザ ーインターフェイスを埋め込む1つのフォームから構成することができます。

ドキュメトレベルの統合より、このアプローチより、簡単に実装することができますが、カスタム必要条件に近しを柔軟的にMapForce グラ フィカルユーザーインターフェイスを構成するためには適切ではない可能性があります。ドキュメトレベルの統合は、アプリケーションに MapForce を1つ1つ埋め込むことを意味します。メイン MapForce エトロールのみではなく、メインドキュメトエディターウィドウの実 装も含まれます。また、オプションで、追加ウィドウの実装も含まれます。このアプローチはGUIの構成に柔軟性を与えますが、選択された 言語により ActiveX エトロールとの高度なインタラグションカ必要とされます。

セクションアプリケーションレベルの統合<u>ど、キュメトレベルの統合</u>は、それぞれのレベルの主要なステップに関して説明しています。<u>ActiveX</u> 統合のサンプルセクションは、C#、HTML、および、Java のサンプルを提供します。これらのサンプルを確認することにより、正しい選択が 直くに行うことができます。オブジェクトレフォレンス・セクションは、プロ・ティとメンドにつれえ統合のオンタン使用できるすべての COM オブジェクト トロイル て説明しています。

MapForce を Visual Studio プラブインとて使用するための情報に関しては、次を参照してくたさい: <u>Visual Studio 内の</u> MapForce。

16.1 必要条件

MapForce ActiveX エトロールをカスタムアプリケーションに統合するコよ以下が使用中のエピューターにインストールされている必要が あます:

- MapForce
- MapForce Integration Package は 次からダウロードできます: http://www.altova.com/ide_integration.html

64 ビナ ActiveX コトロールを統合するコよ、MapForce の64 ビナ バージョンとMapForce Integration Package をインスト ールする必要があます。Visual Studio を使用して、Microsoft.NET プラナフォームの下で開発されたアプリケーション 引してよ MapForce の32 ビナ と64 ビナ バージョンとMapForce Integration Package について下で説明されているとおりインストール ます。

メモ Windows XP とWindows Server 2003 で統合パッケージが正確に作動するためには、再頒布可能パッケージを手動でインストール てくたさい、https://www.microsoft.com/en-US/download/details.aspx?id=48145.

Microsoft .NET (C#, VB.NET) with Visual Studio

MapForce ActiveX ゴトロールを、Microsoft.NET の下で開発された32ビットアプリケーションに統合するコよ以下を使用中のコンピューターにインストールしてください

- Microsoft.NET Framework 4.0 おけお降
- Visual Studio 2010/2012/2013/2015/2017/2019
- MapForce 32 ビオとMapForce Integration Package 32 ビオ
- ActiveX コトロールがVisual Studio ツールボックスに追加されている必要があります(次を参照してくたさい: <u>ActiveX コト</u> ロールをソールボックスに追加する).

64 ビナ ActiveX コトロールを統合する場合、次の必要条件が上記に加え満たされる必要があります。

- MapForce 32 ビナとMapForce Integration Package 32 ビナ をんスールてたさい(Visual Studio は32) ビナで作動するため、32 ビナ ActiveX エトロールをVisual Studio デザイナーに与える必要があます。)
- MapForce 64 ビナとMapForce Integration Package 64 ビナ をインストールてくたさい(実際の64 ビナ ActiveX コトロールをランタイムコンスタムアプリケーションに与えます。)
- Visual Studio 内で、64 ビットビルド更新を作成し、この更新を使用してアプリケーションをビルドします。サンプルに関しては、 次を参照してくたさい、サンプルC# ソリューションの実行。

Java

Eclipse 開発環境を使用して、MapForce ActiveX コトロールを Java アプリケーションに統合するコよ、次を使用中のコンピューターにインストールしてくたさい

- Java Runtime Environment(JRE) おは Java Development Kit(JDK) 7 おは以降
- Eclipse
- MapForce and MapForce Integration Package
- メモ MapForce ActiveX エトロールの64 ビナ バージョンを作動するコよ Eclipse の64 ビナ バージョン、および、 MapForce とMapForce Integration Package の64 ビナ バージョン ofを使用してください。

ケライアントコンピューター上でのMapForce 統合とデプロイ

.NET アプリケーションを作成し他のケライアトトコ配布する場合は、ケライアトコンピューター上に次をインストールしてくたさい

- MapForce
- MapForce Integration Package
- カスタムの統合コード、おは、アプリケーション。

16.2 ActiveX コントロールをツールボックスに追加する

Visual Studio を使用して開発されたアプリケーション内で、MapForce ActiveX コトロールを使用するコよ コトロールの最初に Visual Studio ツールドックスに追加される必要がみます。

- 1. Visual Studio のツール」メニューから、「ツールボックスアイテムの選択」をクリックします。
- 2. 「COM コンポーネント」ダブ、から、MapForceControl, MapForceControl ドキュメト、および、MapForceControl プレース市レダーの横のチェックボックスを選択します。

上記のエトロールが使用することができない場合、下のステップに従います

- 1. 「COM コンポーネント」 ダブかび 参照」をクリックします。 MapForceControl.ocx を MapForce インストールから フォル ダーを選択します。 MapForce Integration Package をインストールすることを忘れない でくたさい。 それ以外の場合、この ファイルを使用することはてきません、次を参照してくたさい、 <u>必要条件</u>.
- 2. を更に高いトミッションと共に再起動するようにプロンプトされると、「異なる資格情報下で再起動する」をクリックします。

Microso	ft Visual Studio	×
+	This task requires the application to have elevated permissions.	
	Why is using the Administrator or other account necessary?	
	Restart under different credentials Saves the current changes and then restarts Microsoft Visual Studio. You will be prompted to change your user account.	
	Cancel the task and return to Microsoft Visual Studio	
🕑 V	ew error <u>i</u> nformation Can	icel

上記のステップが完了すると、MapForce ActiveX コトロールは、Visual Studio ツールボックス内で使用できるよう」ないます。

Toolbo	ĸ	- □ ×
Search Toolbox 🖉 -		- م
All Windows Forms		
▷ Com	mon Controls	
▷ Cont	tainers	
🔺 Men	us & Toolbars	
k	Pointer	
È	ContextMenuStrip	
	MenuStrip	
-	StatusStrip	
10¥	ToolStrip	
ToolStripContainer		
🙁 MapForceControl		
\odot	MapForceControl Doc	ument
	MapForceControl Plac	eHolder
⊳ Data		
▷ Components		
Printing		
▷ Dialogs		
WPF Interoperability		
▷ Gene	eral	

- メモ アプリケーションベルの統合では MapForceControl ActiveX エトロールのみが使用されます(次を参照してくたさい、 アプリケーションベルの統合)。 MapForceControl ドキュメント とMapForceControl プレースホルダー エトロール
 - は、ドキュメトレベルの統合のために使用されます(次を参照してくたさい、ドキュメトレベルの統合)。

16.3 アプリケーションレベルの統合

アプリケーションノベルの統合により使用中のアプリケーションのウイドウこMapForceのインターフェイスを埋め込むことができます。この種類の統合により、すべてのメニュー、ツールドー、ステータスバー、ドキュメトウイドウ、および、ヘルパーウイドウを含むMapForceのユ ーザーインターフェイスを使用することができます。アプリケーションのユーザーインターフェイスのカスタム化はMapForceが提供する機能に限られています。これには、ヘルパーウイドウの整理とサイズ調整、及びメニューとソールドーのカスタム化か含まれています。

アプリケーションベルで統合する場合、統合する必要のある唯一のActiveX コトロールは<u>MapForceControl</u>です。 <u>MapForceControlDocument</u> おけは<u>MapForceControlPlaceHolder</u> ActiveX コトロールをインスタンス化、おけよアクセ ス使用としたいでがさい。

初期化、おけよ、MapForceの振る舞いを自動化するけよ、MapForceControl.のために説明されているプロ・ディ、メンド、とイベントを使用してくたさい。MapForce機能への複雑なアクセスのためにMapForceControl.applicationを使用することを考慮してくたさい。

MapForce アプリケーションをHTML ページ、に埋め込むサンプルに関しては、次を参照してくたさい、アプリケーションレベルのHTML 統合。

Visual Studio を使用するC# おけはVB.NET でのMapForce ActiveX コトロールをアプリケーションベルで統合する基本的なアプリケーションを作成するステップは以下のとおりです:

- 1. すべての必要条件が満たされていることを確認してくたさい(次を参照してくたさい: 必要条件).
- 2. 新規のVisual Studio Windows Forms プロジェクトを新規の空のフォーム共に作成してください。
- 3. お行っていない場合、ActiveX エトロールをソールボクスに追加します(次を参照してくたさい、<u>ActiveX エトロールをソー</u>ルボックスに追加する)。
- 4. ルーンボックスから、MapForceControl をデラッグして、新規のフォームゴロップしてくたさい。
- 5. フォームで MapForceControl を選択し、プロ・ディウイドウ内で IntegrationLevel プロ・ディを ICActiveXIntegrationOnApplicationLevel に設定します。

Pr	Properties 🔹 🗖 >			
a	axMapForceControl AxMapForceControlLib.AxMapForceControl			
	💱 🖓 F 🖉			
Ŧ	(ApplicationSettings)			
Ŧ	(DataBindings)			
	(Name)	axMapForceControl		
	AccessibleDescription			
	AccessibleName			
	AccessibleRole	Default		
	AllowDrop	False		
	Anchor	Top, Left		
	Appearance	0		
	BorderStyle	0 - None		
	CausesValidation	True		
	ContextMenuStrip	(none)		
	Dock	None		
	EnableUserPrompts	True		
	GenerateMember	True		
	IntegrationLevel	ICActiveXIntegrationOnApplicationLevel		
Ŧ	Location	280, 8		
	Locked	False		
+	Margin	3, 3, 3, 3		

- 6. ビルド(x86、x64)するプラホフォームに一致するビルドプラホフォームの更新を作成します。ビルドの更新の作成の方法は以下 のとおいです

 - a. Visual Studio 内のノリューションを右クトックし、「更新 マネージャー」を選択します。 b. 「アクティブなソリューションプラットフォーム」から「新規作成...」を選択し、x86 まけはx64 更新を選択します(このサ ンプルでは x86)。

New Solution Platform
Type or select the new <u>p</u> latform:
x86 🗸
Copy <u>s</u> ettings from:
Any CPU 🔹
Create new project platforms
OK Cancel

Visual Studio 内でノリューションをビルドして実行する準備か整うと、ターゲオプラオフォーム(x86、x64)に一致する更新をしようしてビルドすることを確認してくたさい。

16.4 ドキュメントレベルの統合

アプリケーションレベルの統合に比べ、ドキュントレベルの統合複雑ですが、ActiveX コトロール。を使用することに以アプリケーションに MapForce機能を柔軟的に埋め込むことができます。このアプローチにより、コードはMapForce ユーザーインターフェイスの次のパーソこ 部分的にアクセスすることができます。

- ドキュメントの編集ウインドウ
- プロジェクトウィンドウ
- ライブラリウイ・ボウ
- 概要ウイボウ
- メッセージウイボウ

アプリケーションレベルの統合 で説明されているとおり、ActiveX アプリケーションレベルの統合のためには、MapForceControlのみが必要とされる唯一のエトロールです。しかしなから、ActiveX ドキュメノトレベルの統合の場合は、機能 MapForce か次のActiveX エントロールにより与えられます。

- 1. MapForceControl
- 2. <u>MapForceControl ドキュメント</u>
- 3. <u>MapForceControl プレースホルダ</u>

これらのエトロールはMapForceのアプケーション インストールフォルダーで使用することのできる MapForceControl.ocx ファイル により与えられます。 Visual Studio を使用して ActiveX 統合を開発すると、これらのエントロールを Visual Studio ツールボックスご追加する必要があります (次を参照してください: ActiveX エトロールをソールボックスに追加する)。

ドキュメイレベルでのActiveX コイロールのアプリケーションへの統合の基本のステップは以下のとおりです。

- 1. 最初にアプケーション内で MapForceControl をインスタンス化します。「MapForceControl ドキュメント」と 「MapForceControl プレースホルダ」コトロールへのサポートが有効化されるかがにこのコントロールの統合は必須です。 <u>IntegrationLevel</u> プロ、ティを「ICActiveXIntegrationOnDocumentLevel」(おけよ「1」)に設定することは重要です。ユーザイン対してコントロールを非表示にするけよ「Visible」プロ、ティを「False」に設定します。
- メモ ドキュメトレベルで統合する場合、未知の結果が生成される可能性があるため、MapForceControl の開く」メンドを使用しないでくたさい。MapForceControlドキュメント とMapForceControl プレースホルダの対応するメンドを代わり に使用して開いてくたさい。
 - 2. MapForceControl ドキュメント のインスタンスを少なくとも1 つアプリケーション内に作成します。このエトロールは、アプリケー ションにMapForce のドキュメントの編集ウイドウを与え、必要であれば、複数回インスタンス化することができます。

既存のファイルを開くために、メンボ「開く」を使用します。ドキュメントに関連した機能にアクセスするコよ、メンボとプロ・ティ「ドキュメント」を介してアクセスすることのできる「パス」と「保存」を使用します。

- メモ コトロールは読み取り専用モードをサポートしません。プロノティ「読み取り専用」の値は無視されます。
 - 3. 任意で、アプリケーションにアプリケーションで使用することのできるドキュメトウインドウ以外の各追加ウインドウのために MapForceControl プレースホルダゴントロールを追加することができます。

MapForceControl プレースホルダのインスタンスによアプリケーションウイドウに選択して MapForce の追加ウイドウを 埋め込むことができます。ウイドウの種類(例えば、プロジェクトウイドウ はプロ ティ「PlaceholderWindowID」により 定義することができます。有効なウイドウの識別子のためコよ、次を参照してくたさい: MapForceControlPlaceholderWindow.

メモ 各ウイドウ織別子のためゴ MapForceControl プレースホルダ」のみを使用してくたさい。

MapForce プロジェクトウイドウを選択するプレースヤルダーニントロールのために、追加メソドを使用することができます。 「OpenProject」を使用して、MapForce プロジェクトをロードすることができます。MapForce 自動化インターフェイスからプロ・ディプロジェクトとメソバを使用して、他のプロジェクトに関連したオペレーションを操作します。

例えば、Visual Studioを使用したC# おさよVB.NET 内では、MapForce ActiveX エトロールをドキュメントレベルで統合する基本的なアプリケーションの作成ステップは、以下にリストされるものに類似しています。必要に応じて適用は複雑になる可能性があります。しかしながら、ActiveX ドキュメントレベルの統合のさかの必要条件として、下の命令を理解することは重要です。

- 1. 新規の空のフォームと共に新規のVisual Studio Windows Forms プロジェクトを作成します。
- 2. お行っていない場合、ActiveX エトロールをツールボックスに追加します(次を参照してくたさい: <u>ActiveX エトロールをソールボックスに追加する</u>)。
- 3. ツールボックスから MapForceControl をデラックして新規のフォームエラックします。
- 4. **MapForceControl**の「Integration Level」プロテルで「ICActiveXIntegrationOnDocumentLevel」にと「Visible」プロテルでを「False」に設定します。コードから、おは、「プロ・ティ」ウイドウから行うことができます。
- 5. <u>Map ForceControl ドキュメント</u> をソールボックスからフォームコボラッグします。 このエトロールは、使用中のアプリケーション にMapForce のメインウイバーウを与え、ドキュメトのナカにサイズを調整することができます。
- 6. 代わりに、フォームコ つまけは複数のMapForceControl プレースホルダ エトロールを追加することができます(アプリケー ションが必要とする各追加ウィバウの型など。例えば「プロジェクト」ウィバウなど)。このような追加プレースホルダー エトロー ルをメイバキュメト エトロールの下、まけよ 左右に通常配置します。例



- 7. 各「MapForceControl プレースホルダ」コナロールの「PlaceholderWindowID」プロ・デを有効なウイド 「識別子に設定してくたさい。有効な値のパトに関しては、次を参照してくたさい、<u>MapForceControlPlaceholderWindow</u>。
- 8. アプリケーションコマイを以下のように追加します(少なくとも、ドキュメントを開き、保存し、閉じる必要がみます)。

MapForce コマバをクエリする

ドキュメナレベルで統合する場合は、MapForceメニュー・おけまツールレーをアプリケーションで使用することはできません。代われに、必要とするコマイを取得し、状態を確認し、プログラム的に実行することができます。

- 使用することのできるすべてのコマンドを取得するコよ MapForceControl の CommandsList プロ 守 を使用します。
- メニュー構造によ整理されているコマンドを取得するコは、メイン Menu プロンティを使用します。
- 表示される順番に整理されているコマンドを取得するコよ、Toolbars プロ・ティを使用します。
- MapForce コマイを送信するコよ Exec メノイを使用します。
- コマイが現在有効化されているか、おけよ、無効化されているかクロノするけっと、QueryStatus メノンドを使用します。

これによりMapForceコマボをアプケーションのメニューとソール・イニ柔軟的に統合することができます。

MapForceのインストールは、MapForce内で使用されているコマイドラベルを与えます。GIF書式内のアイエンのためにMapForce インストールのフォルダー「<ApplicationFolder>モxamples半ActiveX¹mages」を参照してくたさい。fコマイド名に対応するファ イル名はコマイを参照センタン内にリストされています。

一般的な注意点

MapForce の振る舞いを自動化するコよ、MapForceControl、MapForceControlドキュメントとMapForceControlプレレースホルダ内で説明されているプロンティ、メンド、とイベトを使用してくたい。

MapForce機能への高度なアクセスに関しては、次のプロレテを考慮してくたさい

- <u>MapForceControl.Application</u>
- <u>MapForceControlDocument.Document</u>
- <u>MapForceControlPlaceHolder.Project</u>

これらのプロ ディは MapForce 自動化インターフェイス (MapForceAPI) へのアクセスを提供します

メモ ドキュメトを開くはは、適切なドキュメトコトロールの<u>MapForceControlDocument.Open</u> おは <u>MapForceControlDocument.New</u>を使用してけ違い。 プロジェケを開くはは、MapForce プロジェケ・ウィドウを埋 め込むプレース市りダコトロール上の<u>MapForceControlPlaceHolder.OpenProject</u>を常に使用してけ違い。

異なるプログラング環境内で必要なエントロールをインスタント化しアクセスする方法を説明するサンプルニ関しては、次を参照してくたさい: ActiveX 統合のサンプル。

16.5 ActiveX 統合のサンプル

このセクタンは、異なるエンテナー環境とプログラム言語を使用する MapForce ドキュメトレベルの統合のサンプルを含みます (HTML セクションコよ 追加でアプリケーションレベルの統合のサンプル 含まれます)。全てのサンプルのノースコードは、MapForce インストール内のフォルダー <ApplicationFolder>\Examples\ActiveX で使用することができます。

16.5.1 C#

C# and Visual Studio のよめの基本のActiveX 統合サンプルノリーションは、フォルダー <ApplicationFolder>\Examples\ActiveX\C#内にあります。ソースコードをコンパルする前に、サンプルを実行して、全ての 必要条件が満たされていることを確認してくたさい、次を参照してくたさい、サンプルC#ソリューションの実行)。

16.5.1.1 サンプル C# ソリューションの実行

フォルダー<ApplicationFolder>\Examples\ActiveX\C#内で使用することのできるサンプルVisual Studio ソリューションは MapForce ActiveX コトロールの使用方法について説明しています。このノリューションをビルドして実行する前に、次のステップに注意 してくたさい

ステップ 1: 必要条件をチェックする

Visual Studio 2010 おけお以降は、サンプルンリューションを開く必要かあります。必要条件の完全なリストは、次を参照してくたさい、必要条件。

ステップ 2: 書き込みのパーミッションを保有するディレクトリにサンプルをコピーする

Visual Studio を管理者として作動することを回避するコよ、デフォルトの場所から実行する代われに、書き込みのパーミッションを持つディレクトリークロードをコピーします。

ステップ 3: すべての必要とされるコートロールプロノティをチェックし設定する

サンプルアプリケーションは MapForceControlDocument の1 つのインスタンスとMapForceControlPlaceHolder エトロールの 複数のインスタンスを含んでいます。下のテーブルに表示されているとおりにこれらのエレトロールの次のプロイティが設定されていることを確認し てくけざい

그가마니名	プロパティ	プロ/ 守/値
axMapForceControl	IntegrationLevel	ICActiveXIntegrationOnDocume ntLevel
axMapForceControlLibrary	PlaceholderWindowl D	0

그사머ール名	プロパティ	プロ/ 守/値
axMapForceControlOutput	PlaceholderWindowl D	2
axMapForceControlPreview	PlaceholderWindowl D	1

ActiveX コントロールのプロ ディをビュー、おけよ 設定する方法は 以下のとおしです:

- 1. デザイナーウイズウ内で MDIMain.cs フォームを開きます。
- メモ 64 ビナト Windows では、デザイナーウインドウで開く前に Visual Studio ソレューションを「x86」 にビルドを更新する必要が ある可能性があります。64 ビナトアプリケーションとしてサンプルをビルドする場合は、次を参照してくたさい、必要条件。



2. Visual Studio のドキュメント概要 ウイボウを開きます「表示」メニューから、「他のウィンドウ|ドキュメントの概要」をクリックします)。

Document Outline - MDIMain	×
MDIMain Form	
axMapForceControlOutput AxMapForceControlPlaceH	older
axMapForceControlLibrary AxMapForceControlPlaceHe	older
axMapForceControl AxMapForceControl	
🔺 🖹 mainMenu MainMenu	
🔺 🚰 menultem1 Menultem	
menuFileNew MenuItem	
🗗 menuFileOpen MenuItem	
🗗 menuFileSave MenuItem	
menuFileSaveAs MenuItem	
menuFileClose MenuItem	
menultem2 Menultem	
fileGenCode Menultem	
fileMSettings Menultem	

- 🗗 menultem3 Menultem
- 3. ドキュメント概要 ウィドウ内のActiveX コトロールをクリックし、プロパティウィドウ内の必要とされるプロパティを編集します。 例:

Pr	operties	× 🗆	×
a	MapForceControl AxMapForceC	ControlLib.AxMapForceControl	•
	🛛 🔁 F 🖉		
	(Name)	axMapForceControl	*
	AccessibleDescription		
	AccessibleName		
	AccessibleRole	Default	
	AllowDrop	False	
	Anchor	Top, Left	
	Appearance	0	
	BorderStyle	0 - None	
	CausesValidation	True	
	ContextMenuStrip	(none)	
	Dock	None	
	EnableUserPrompts	True	
	GenerateMember	True	
	IntegrationLevel	ICActiveXIntegrationOnDocumentLevel 🗸	
Ŧ	Location	280, 8	-

IntegrationLevel

ステップ 4: ビルドプラナトフォームを設定する

- ビルド(x86、x64)するプラオフォームに一致するビルドプラオフォームの更新を作成します。ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio 内のノリューションを右クトックし、「更新マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成…」を選択し、x86 まけま、64 更新を選択します(このサンプルでは、x86)。

New Solution Platform
Type or select the new <u>p</u> latform:
x86 🗸
Copy <u>s</u> ettings from:
Any CPU 👻
✓ Create new project platforms
OK Cancel

Visual Studio 内でノリューションをビルドして実行する準備か整うと、ターゲ・トプラ・トフォーム(x86、x64)に一致するように更新をしよう してビルドすることを忘れないでください。それ以外の場合、ランタイムエラーが起こる可能性があります。

作動中のサンプルでは、メインMDIフレームウイドウが表示されています。「ファイル | 開く」を使用してマピングファイル例えば MapForce サンプルフォルダーであるMarketingExpenses.mfd)を開きます。ファイルはロードされ新規ドキュメント 子ウイド ウ内で表示されます:

🖳 MapForceControl			
File Edit Insert Component	Connection	unction Output Debug View Tools Help <u>W</u> indow	
Fore	Ā		
		👷 C:\Users\altova\Documents\ 🗖 🖻 🗾	
avo	result = avo(
count	result = cour		
max	result = maxi		
max-string	result = maxi	😫 ExpReport	
min	result = min(File: ExpReport.xml File/String [>	
min-string	result = min(> 🕞 () expense-report expense-repor	
string-join	result = strin	> = detailed C= "",	
sum	result = sum	currency currency	
		⇒ — = total-sum	
conversion functions		Person	
boolean	result = book	> — () First	
format-date	result = form	> — () Last	
format-dateTime	result = form	>	
format-number	result = form		
format-time	result = form	Mapping DB Query Output 📀 HTML (
number	result = numi		
string	result = strin		
nie path functions			
get-fielder	folder - get i		
get-tolder	filenath = ma		
main-mid-niepath	filepath = mf		
remove_fileevt	result_filenat		
remove-folder	filename = re		
renlace_fileevt	result_filenat		
resolve-filenath	result-filenat	I TABBE SISX	
	readic-neput		
			_
auto-number	result = auto		
	rooun auto		
_ logical functions	7		
Search for function			
Add/Remove Libraries	3		-

Click to enlarge image

ドキュメトをロードした後、メニューを使用してアクティブなドキュメトレス対してコマイドを実行することができます。 コンテキストメニューも使用し、追加ドキュメントをロードすることもできます。「ファイル | 保存」コマイドを使用して変更を保存できます。

16.5.1.2 コマンド情報の取得

MapForceControl は、CommandsList、MainMenu、および Toolbars プロ テイにお MapForce の全てのコマンドへのアクセ スを与えます。フォルダー < ApplicationFolder > Examples ActiveX \C#内にあるサンプルプロジェクトは、MainMenu プロ ティ を使用して MapForce メニュー構造を動的に作成します。

メニューコマボを取得するコードは MDIMain.cs ファイル内のMDIMain メノバでみつけることができます:

public MDIMain()

{

```
// ...
// Get the MainMenu property of the control and create the menu structure from it.
MFLib.MapForceCommand objCommand = this.axMapForceControl.MainMenu;
InsertMenuStructure(mainMenu, objCommand);
```

コードリスト内で、mainMenuはMDIフレームウイドウの既存の静的メニューです。 Visual Studio Designer 内で MDIMain.cs フォームを開くと、このメニューは次の2つのメニューアイテムから構成されていることに気が付きます。「ファイル」とウイドウ。

м	DI	Main.cs [Design] 👳 🗡		
		<mark>₽</mark> MapF <u>F</u> ile	orceControl <u>W</u> indow		
					MapForceX Manager Control

MDIMain.cs

メンド InsertMenuStructure はmainMenu とobjCommand オジンナトを・ラメーターとして取ります。(前者は既存の静的なメ ニューを含み、後者は、MapForce ActiveX エトローリから取得されたフルメニュー構造を含みます)取得された MapForce メニュー構 造は既存の静的 メニューにマージされます。メニュー「ファイル」、「プロジェクト」、および「ウィンドウ」は動的に追加されません。これら のメニューは開かれているドキュメント とアウティブご作業するオュタ、おた、このサンプルの範囲外のコードを必要とするオュタ、これは意図的です。 基本のファイル管理 コマンド (create、open、save、bring into focus)既存の静的メニュー「ファイル」とウンドウにより扱われま す。すべての他のメニューのActiveX エトロールのMainMenu プロ、ティから取得された情報をベースに動的に挿入されます。新規のメ ニューは、「ファイル」の後、「ウィンドウ」の前に挿入されます。例えば、メニューインデックス1 から始まるなど。

メントド InsertMenuStructure は、MapForceCommand オブジェクト内で見つたわたすべてのトップレベルメニュを介して反復し、 新規のメニューをそれぞれご追加します。各トップレベルメニューオ自身の子メニューアイテムを所有するため、メンド InsertMenuCommand への呼び出しは、発生した子メニューアイテムのために実行されます。更に、各子メニューアイテムは自身の子ニュ ーアイテムを持つことができるため、InsertMenuCommand メンドは子メニューアイテムが存在しなくなるまで再帰します。

動的に追加されるコマドは CustomMenultem cs 内で定義されるクラス CustomMenuItem のインスタンスです。このクラスは System.Windows.Forms.MenuItem クラスから生成され、MapForce コマド ID に保存するために追加メンバーが存在します。

public class CustomMenuItem : System.Windows.Forms.MenuItem
{
 public int m_MapForceCmdID;
}

(他のコマドのためのコンテナー以外の)すべての動的に追加されたコマド は、コマドの処理をおこなう同じ イベト・ハドラー AltovaMenuItem_Click を取得します:

```
private void AltovaMenuItem_Click(object sender, EventArgs e)
{
    if(sender.GetType() == System.Type.GetType("MapForceapplication.CustomMenuItem"))
    {
        CustomMenuItem customItem = (CustomMenuItem)sender;
        ProcessCommand(customItem.m_MapForceCmdID);
    }
}
```

} }

コマボが子コマボを持つ場合)他のコマボのかがのエレテナーである場合、イベトハンボラーAltovaSubMenu_Popupを取得します。このいボラーは、各子コマボの状態をクエル、必要に応じて有効化、おけよ、無効化することができます。

これにより各コマイトカ意味のある時にのみ有効化されます。(例えば、「ファイル | 保存」メニューアイテムはアクティブなドキュメントか開かれていない場合無効化されている必要があります)。

メンボ ProcessCommand は MapForceControl 自身およ MapForceControl Document コトロール内ロードされてい るアクティブな MapForce ドキュメト のどちらかに実行を委任します。MapForceControl はオストアプリケーション内で現在アクティブな ドキュメトを知る方法がな サンダン要しないます。

```
private void ProcessCommand(int nID)
{
    MapForceDoc docMapForce = GetCurrentMapForceDoc();
    if(docMapForce != null)
        docMapForce.axMapForceControlDoc.Exec(nID);
    else
        axMapForceControl.Exec(nID);
}
```

16.5.1.3 イベントの処理

MapForce ライブラル内の全てのイベトは、接続ポイントをベースしており、カスタムイベトノンドラーを与えるために、C# 委任メカニズ ムを使用することができます。 MapForce ライブラルのプロ・ティページ上でイベトの完全なリストを見つけることができます。 下のイメージ は、メイン MapForceControl イメージを表示しています:

Properties	• □ ×
axMapForceControl AxMapForceControlLib.AxMapForceControl	•
≣ 🛃 🖓 🗲 🖉	
(DataBindings)	<u></u>
CausesValidationChanged	
ClientSizeChanged	
ContextMenuStripChanged	
ControlAdded	
ControlRemoved	
DockChanged	
Enter	
Leave	
LocationChanged	
MarginChanged	
MouseCaptureChanged	
Move	
OnCloseEditingWindow	
OnComboBoxSelectionUpdate	
OnContextChanged	
OnDocumentOpened	
OnFileChangedAlert	
OnFileExternalChange axMapForceControl_OnFileExternalCh	ange 🗸
OnGetLastActiveEditor	
OnGetPluginManagerControl	-

サンプルプロジェクトはOnFileExternalChange イベトのみをオー、トライドします。C# フレームワークによりC# 委任は作成されます。空のイベトハンドラーをフィルするのみで行うことができます。

例えば、下に示されるいンドラーの実装は、ファイルの再ロードをオフェゴル替え、MapForceControl によりロードされたファイルの外部により変更されたことをユーザーに通知するためにシッセージボックスを表示します。

private void axMapForceControl_OnFileExternalChange(object sender, AxMapForceControlLibDMapForceControlEvents_OnFileExternalChangeEvent e) {
MessageBox.Show("Attention: the file " + e.strPath + " has been changed from outside¥nbut reloading is turned off in the sample application!");
<pre>// This turns off any file reloading: e.varRet = false; }</pre>

16.5.2 HTML

このセクションのコードリストは、アプリケーションレベルビキュメントレベルでのMapForceControlの統合方法を示しています。全てのサン プルのナダのノースコードはMapForce インストールのフォルダー <ApplicationFolder>\Examples\ActiveX\HTML内にあり ます。サンプルは、32ビットアプリケーションとして作動する場合、InternetExplorer内でのみ作動することができます。 メモ 64 ビオ モド上で作動する場合、nternet Explorer 10 おけは以降は、ActiveX コトロールをロードしません。

16.5.2.1 アプリケーションレベルの HTML 統合

このサンプルは、アプケーションベルでのHTMLページへのMapForce コイロールの簡単な統合について説明しています。統合に関し ては、次のセクションで説明されています

- HTML コード内で MapForce Control を開始します。
- ボタンがドキュメノトをロードするように実装し、コード生成タスクを自動化します。 •
- アプリケーションイベトのナックのアクションを定義する。

このサンプルのためのコードは、MapForce インストール内の次の場所で使用することができます:

<ApplicationFolder>\Examples\ActiveX\HTML\MapForceActiveX ApplicationLevel.htm.

コントロールインスタンスを作成する 16.5.2.1.1

HTML Object タグは MapForceControl のインスタンスを作成するために使用されます。Classid は MapForceControl で す。幅と高さがウイドウのサイズを指定します。アプリケーションレベリはデフォリトのため、追加の、ウメーターは必要ありません。

```
<OBJECT id="objMapForceControl"
       Classid="clsid:A38637E9-5759-4456-A167-F01160CC22C1"
       width="800"
       height="500"
       VIEWASTEXT>
```

</OBJECT>

デフォルトのドキュメントを開くボタンを追加する 16.5.2.1.2

一部のタスクの自動化のシンプルなサンプルとして、ページンボタンを追加します

<input type="button" value="Open Marketing Expenses" onclick="BtnOpenMEFile()">

クリックされると、定義済みドキュメントがMapForceControl 内で開かれます。異なるインストール上でサンプルを作動するため、 MapForceControl」マ対して相対的なファイルを検索するためコンソンドを使用します。

<SCRIPT ID=Javahandlers LANGUAGE=javascript>

```
// ------
// open a pre-defined document
function BtnOpenMEFile()
{
```

objMapForceControl.Open("C:\Documents and Settings\username\My Documents\Altova\XMLSpy2021\Examples/MarketingExpenses.mfd");

</SCRIPT>

16.5.2.1.3 コード生成のためにボタンを追加する

アクティブンドキュメトのオメのコード生成は、メニューを使用して使用することができますが、出力の場所をユーザーに問わずにコードが生成されるボタンを作成します。 メノボンゴ前のセクションで使用されナメノバン類似しています。

おとを追加します

<input type="button" value="Generate XSLT" onclick="BtnGenerate(0)">
<input type="button" value="Generate Java" onclick="BtnGenerate(1)">
<input type="button" value="Generate C++" onclick="BtnGenerate(2)">
<input type="button" value="Generate C#" onclick="BtnGenerate(3)">

現在定義されているデフォルトの出力フォルダーのサブフォルダー内にコードを生成するスクリプトを与えます。

```
<SCRIPT ID=Javahandlers LANGUAGE=javascript>
// ------
// generate code for active document into language-specific sub folders of
// the current default output directory. No user interaction necessary.
function BtnGenerate(languageID)
{
      // get top-level object of automation interface
      var objApp = objMapForceControl.Application;
      // get the active document
      var objDocument = objApp.ActiveDocument;
      // retrieve object to set the generation output path
      var objOptions = objApp.Options;
      if ( objDocument == null )
             alert( "no active document found" );
      else
      {
             if (languageID == 0)
             {
                    objOptions.XSLTDefaultOutputDirectory =
objOptions.XSLTDefaultOutputDirectory + "\\XSLTGen";
                   objDocument .GenerateXSLT();
             }
             else if (languageID == 1)
             {
                    objOptions.CodeDefaultOutputDirectory =
objOptions.CodeDefaultOutputDirectory + "/JavaCode";
                    objDocument .GenerateJavaCode();
             }
             else if (languageID == 2)
             {
                    objOptions.CodeDefaultOutputDirectory =
objOptions.CodeDefaultOutputDirectory + "/CPPCode";
                   objDocument .GenerateCppCode();
             }
             else if (languageID == 3)
             {
                    objOptions.CodeDefaultOutputDirectory =
objOptions.CodeDefaultOutputDirectory + "/CSharpCode";
                    objDocument .GenerateCHashCode();
      }
}
</SCRIPT>
```

16.5.2.1.4 カスタムイベントへの接続

サンプルは、プレシッツを表示するためこMapForceControlカスタムイベトのための2つのイベトのコールシンを実装します:

16.5.2.2 ドキュメントレベルの HTML 統合

このサンプルはドキュメトレベルでHTMLページにMapForceコトロールを統合する方法について説明しています。次のトピックについて説明されています。

- HTMLコード内でMapForceControl ActiveX コトロールオブジェクトをインスタンス化します。
- MapForce ファイルの編集を許可する MapForceControlDocumentActiveX コトロールをインスタンス化します。
- MapForceControl プロジェクトウィドウのためのMapForceControlPlaceHolder をインスタンス化します。
- MapForce ヘルトーウィボウをホストするためJMapForceControlPlaceHolder をインスタンス化します。
- 頻繁に使用される MapForce コマンドのためのシンプルなカスタマーソールレーを作成します。
- MapForce のCOM 自動化インターフェイスを使用するポタンを追加します。
- コマドボタンを更新するためのイベトハンドラーを使用します。

このサンプルは MapForce インストールの ApplicationFolder>\Examples\ActiveX\HTML\ フォルダー内のファイル MapForceActiveX ApplicationLevel.htm 内で使用することがあます。

16.5.2.2.1 MapForceControl をインスタンス化する

MapForceControl HTML OBJECT タグは MapForceControl のインスタンスを作成するために使用されます。 クラスID は MapForceControl です。ユーザーインターフェイスのためのエントロール間ジェージャーとして使用される幅と高さよ 0 に設定されていま す。統合レベルはOBJECT タグ内で、デメーターとして指定されています。

```
<OBJECT id="objMapForceXMapForceControl"
Classid="clsid:A38637E9-5759-4456-A167-F01160CC22C1"
width="0"
height="0"
```

```
VIEWASTEXT>
<PARAM NAME="IntegrationLevel" VALUE="1">
</OBJECT>
```

16.5.2.2.2 エディターウィンドウの作成

HTML OBJECT タグは編集ウイドウを埋め込むために使用されます。追加のカスタム・ラメーターは新規のからのマピングを使用してコトロールが初期化されるように指定します。

```
<OBJECT id="objDoc1"
Classid="clsid:DFBB0871-DAFE-4502-BB66-08CEB7DF5255"
width="600"
height="500"
VIEWASTEXT>
<PARAM NAME="NewDocument">
</OBJECT>
```

16.5.2.2.3 プロジェクトウィンドウの作成

HTML OBJECT タグは MapForceControlPlaceHolder ウイドウを作成するかとうます。最初の追加カスタム・ラメー ターは、MapForce プロジェクトウィドウを表示するかかのプレース市以を定義します。2番目の、ラメーターは、MapForce インストー ルで提供されたサンプルプロジェクトのローをロードします(以下の場所で見つけることができます: <yourusername>/MyDocuments フォルダー)。

```
<OBJECT id="objProjectWindow"
    Classid="clsid:FDEC3B04-05F2-427d-988C-F03A85DE53C2"
    width="200"
    height="200"
    VIEWASTEXT>
    <PARAM name="PlaceholderWindowID" value="3">
        <PARAM name="FileName" value="MapForceExamples/MapForceExamples.mfp">
    </OBJECT>
```

16.5.2.2.4 ヘルパーウィンドウのためにプレースホルダを作成する

HTML OBJECT タグは、異なる MapForce ヘルレーナイドウをオストすることのできる MapForce ControlPlaceHolder ActiveX コトロールをインスタンス化するために使用されます。最初から、ヘルレーナイドウは表示されません。サンプルファイルを参照してくたさい。

```
<OBJECT id="objEHWindow"
    Classid="clsid:FDEC3B04-05F2-427d-988C-F03A85DE53C2"
    width="200"
    height="200"
    VIEWASTEXT>
    <PARAM name="PlaceholderWindowID" value="0">
</OBJECT>
```

3つのドタイことは表示される実際のウインドウを切り替えることができます。JavaScript実行ボタンをクリックすることにとり、プロ、ティ PlaceHolderWindowIDを定義されている対応する値に設定します。

16.5.2.2.5 カスタムツールバーの作成

カスタムソールレーは、MapForceコマンドのイメージを持つおタインとり構成されています。コマンド ID 番号は、表示されるおタン要素内で見つけることができます

ボタンをクリックすると、対応するコマンド ID がマネジャーコントロールに送信されます。

16.5.2.2.6 ボタンを追加する

サノプルでは、自動化コードを表示する状況を追加します。

```
<input type="button" value="New File" onclick="BtnNewFile(objDoc1)">
<input type="button" value="Save File" onclick="BtnSaveFile(objDoc1)">
<input type="text" title="Path" id="strPath" width="150">
<input type="button" value="Open MarketingExpenses" onclick="BtnOpenMEFile(objDoc1)">
```

対応する Java Script 修正は以下のよう ないます:

```
<SCRIPT ID="Javahandlers" LANGUAGE="javascript">
// ------
// open a document in the specified document control window.
function BtnOpenMEFile(objDocCtrl)
{
      // do not use MapForceX.Application.OpenDocument(...) to open a document,
     // since then MapForceControl wouldn't know a control window to show
     // the document in. Instead:
     objDocCtrl.OpenDocument("C:\Documents and Settings\username\My Documents\
           Altova\XMLSpy2021\Examples/MarketingExpenses.mfd");
     objDocCtrl.setActive();
}
// -----
// open a new empty document in the specified document control window.
function BtnNewFile(objDocCtrl)
{
     objDocCtrl.OpenDocument("");
     objDocCtrl.setActive();
}
// _____
                        _____
// Saves the current file in the specified document control window.
function BtnSaveFile(objDocCtrl)
{
      if(objDocCtrl.Path.length > 0)
           objDocCtrl.SaveDocument();
     else
      {
           if(strPath.value.length > 0)
            {
                  objDocCtrl.Path = strPath.value;
                  objDocCtrl.SaveDocument();
            }
           else
            {
                  alert("Please set path for the document first!");
                 strPath.focus();
            }
      }
     objDocCtrl.setActive();
</SCRIPT>
```

16.5.2.2.7 ボタンステータスを更新するためのイベントハンドラーを作成する

コマドの可用性はマウスクトック、おけまキーボード操作にと興なる場合が砂ます。MapForceControlのカスタムイベト OnUpdateCmdUIは MapForceコマナドに関連するボタンの有効化された、無効化された状態を更新することができます。コマナドの 有効化、無効化に関わらず、メノボ MapForceControl.QueryStatus はクエルのために使用されます。

<SCRIPT LANGUAGE="javascript">

| funct | tion objMapForceX::OnUpdateCmdUI() | |
|------------------|--|--------|
| {
if (| <pre>document.readyState == "complete") // 'complete'</pre> | |
| { | // undate status of buttons | |
| | GenerateXSLT.disabled = ! (objDoc1.QueryStatus(13617) & 0x02); | // not |
| enabled | GenerateJava.disabled = ! (objDocl.QueryStatus(13587) & 0x02); | // not |
| enabled | GenerateCpp.disabled = ! (objDoc1.QueryStatus(13589) & 0x02); | // not |
| enabled | GenerateCSharp.disabled = ! (objDoc1.OuervStatus(13588) & 0x02); | // not |
| enabled | | ,, |
| | <pre>btnFuncUserDef.disabled = ! (objDoc1.QueryStatus(13633) & 0x02);
btnFuncUserDefSel.disabled = ! (objDoc1.QueryStatus(13634) & 0x02)
btnFuncSettings.disabled = ! (objDoc1.QueryStatus(13632) & 0x02);
btnInsertInput.disabled = ! (objDoc1.QueryStatus(13491) & 0x02);</pre> | ; |
| | <pre>btnGenXSLT.disabled = ! (objDocl.QueryStatus(13617) & 0x02);
btnGenXSLT2.disabled = ! (objDocl.QueryStatus(13618) & 0x02);
btnGenXQuery.disabled = ! (objDocl.QueryStatus(13586) & 0x02);
btnGenCPP.disabled = ! (objDocl.QueryStatus(13589) & 0x02);
btnGenCSharp.disabled = ! (objDocl.QueryStatus(13588) & 0x02);
btnGenJava.disabled = ! (objDocl.QueryStatus(13587) & 0x02);</pre> | |
| }
} | | |
| // se
 | et activity status of simulated toolbar | |

16.5.3 Java

MapForce ActiveX コパーネトにJava コードからアクセスすることができます。Java 統合は下にリストされるライブラノで提供されています。これらのライブラルは、MapForce とMapForce Integration Package の両方をインストールた後、MapForce インストール 内のフォルダー <ApplicationFolder>\Examples\JavaAPI で使用することができます(次を参照してくたさい: 必要条件)。

- AltovaAutomation.dll: Altova 自動サードのためのJNI ラッドー(MapForce の32 ビナ インストー)の場合)
- AltovaAutomation_x64.dll: Altova 自動サードのかのJNI ラッド(MapForce の64 ビオ インストールの場合)
- AltovaAutomation.jar: Altova 自動サーバーにアクセスします

- MapForceActiveX JavaDoc.zip: Javadoc イノターフェイスのためのヘリレプドキュメトを含む Javadoc ファイル

メモ Java ActiveX 統合を使用するコよ、dllと.jar ファイルが Java クラス検索/ Splこ含まれている必要があります。

サンプルJava プロジェクト

サンプルJava プロジェケイは製品のインストールにより提供されます。Java プロジェケトをテストし、使用用途に応じて変更することができます。詳細に関しては、次を参照してください、サンプルJava プロジェケト。

JavaRules にActiveX ゴトロール名をマピングする際のルール

ActiveX エトロールのドキュメントに関しては、次を参照してくたさい、オブンエクトレファレンス。オブシェクトの名前変換は Java の場合、 他の言語と比較して、若干異なります。具体的には、ActiveX エトロールと Java ラッパー間のマメピングのためのルールは、以下のとおり です:

- クラスとクラス名 MapForce ActiveX インターフェイスの全てのエンポーネントのために、コンポーネントの名前を持つ Java クラスが存在します。
- メノボ名

Javadoc イターフェイス上のシンド名は、COM イターフェイス上で使用される名前と同じですが、Java 名前変換に準拠 する小文字で開始します。COM プロ、ディイアクセスするゴは、プロ、ディ名をプレフィックスするgetとsetから始まる Java メ ンド使用することができます。 ふプロ、ディか書き込みをサポートしてい場合、setter メンドを使用することはできません。サンプ ル MapForceControlのIntegrationLevel プロ、ディ、Java メンド getIntegrationLevel と setIntegrationLevel を使用することができます。

• 列挙

ActiveX インターフェイス内で定義される全ての列挙のために、Java 列挙は同じ名前と植を使用して定義されます。

 イベント とイベント・ハンドラー イベントをサポートする自動化インターフェイス内の全てのインターフェイスのかがに、同じ名前と'イベント'を使用した Javadoc イ ンターフェイスを使用することができます。単一のイベントのオーバーロードを簡素化するかがに、全てのイベントのかかのデフォルトの 実装を使用する Java クラスが提供されます。この Java クラスの名前は、イベント インターフェイスの名前と 'DefaultHandler'です。例: MapForceControl:アプリケーションにアクセスするかかの Java クラス MapForceControlEvents:MapForceControl のかのイベント インターフェイス MapForceControlEventsDefaultHandler:MapForceControlEvents のかのデフォルトの いどうー

マピングルールの例外

上記のルールの例外の一部は、以下のとおりです。これらは下にリストされています。

インターフェイス	Java クラス内の変更
MapForceControlDocument, method New	名前を付けて保存済み。 newDocument
MapForceControlDocument, method OpenDocument	削除済み。次を使用してくたさい Open メノ・ド
MapForceControlDocument, method NewDocument	削除済み。次を使用してくたさい newDocument メノノド
MapForceControlDocument, method SaveDocument	削除済み。次を使用してくたさい Save メノバ

このセクション

このセクションは、Java コードからアクセスすることのできる基本のMapForce ActiveX 機能の一部を紹介しています。次のサブセクションに整理されています。

- <u>サンプルJava プロジェクト</u>
- <u>ActiveX ゴトロールの作成</u>
- 基本のイベトハンドリング
- <u>UI アップデートイベントハンドリング</u>
- <u>MapForce マピングテーブルの作成</u>

16.5.3.1 サンプル Java プロジェクト

MapForce インストール ジケージコよ アプリケーション フォルダー: <ApplicationFolder>\Examples\ActiveX\Java\の ActiveX サンプルフォルダー(あるサンプルJava プロジェクトが含まれています。

Java サンプルは、Java. を使用して作成された共有されるデスクトップアプレケーション内にMapForceControl を統合する方法を示しています。

バッチファイルBuildAndRun.batを使用してコマイドラインからテストすることができ、また、Eclipse内からサンプルプロジェクトをエンパイルして実行することもできます。これらのプロシージャの使用方法を参照してくたさい。

ファイルリスト

Java サンプルフォルダーコは、サンプルプロジェクトを作動するために必要な全てのファイルは含まれています。これらのファイルは下にリストされています

.classpath	Eclipse プロジェクトヘルレーファイル
.project	Eclipse プロジェクトファイル
AltovaAutomation.dll	Java-COM ブッジ DLL パト (32ビナインストール用)
AltovaAutomation_x64.dll	Java-COM ブッジ DLL パト(64ビナインストール用)
AltovaAutomation.jar	Java-COM ブリッジ. Java ライブラルドト
BuildAndRun.bat	コンパイルする シチファイル および、コマボラインプロンプトからサンプルコードを 実行します。Java 仮想マシンが ラメータとして存在する箇所にフォルダーか期 待されます。
MapForceActiveX.jar	MapForce ActiveX ゴナロールのJava クラス
MapForceActiveX_JavaDoc.zip	Java API のためのヘルプドキュメントを含む Javadoc ファイル
MapForceContainer.java	Java サンプルノースコード
MapForceContainerEventHandler.java	Java サンプルノースコード

1404 ActiveX Integration

MapForceTable.java	Java サンプルノースコード
--------------------	-----------------

サンプルの内容

サンプルは、AWTフレームウイドウ内でMapForceドキュメトエディターウイドウ、MapForceプロジェクトウイドウ、MapForce ライブラリウイドウおよびMapForce検証ウイドウを表示します。MapForceのために定義されているメインメニューを読み取り、同じ 構造を持つAWTメニューを作成します。このメニューまけコンプロジェクトを使用して、ドキュメントエディター内でファイルを開き作業することが できます。

サンプルを必要に応じて変更することができます。

コードリスト内で次の特定の機能が説明されています

- <u>ActiveX エトロールの作成</u>:自動サーバーとて考えられている MapForce を返します。ませま、作動中の場合、 MapForce をアクティブ化します。
- <u>コトロール内のデータのロード</u>: MapForce にインストールされているサンプルドキュメクトの1つを検索し開きます。
- <u>基本のイベトノンドリング</u>: すべての開かれているドキュメントのビューをテキストビューに変えます。コードも開かれているドキュメント内での反復の方法を示しています。
- <u>メーユー</u>: アクティブなドキュメントを検証し、メッセージボックス内に結果を表示します。コードは出力/ ラメータの使用方法について 説明しています。
- UI アップデートイベトハンドリング: MapForce イベトの処理方法について説明しています。
- <u>MapForce マンピングテーブルの作成</u>: MapForce マンピングテーブルの作成とモーダルアクティブ化にていて説明しています。

サンプルフォルダーへのいるの更新

与えられているサンプルを作動する前に、MapForceContainer.java ファイルを編集します。具体的はよ MapForce サンプルファ イルが使用中のオペレーティングシステム上で保存されている実際のフォルダーを参照する次のパンを確認します:

```
// Locate samples installed with the product.
final String strExamplesFolder = System.getenv( "USERPROFILE" ) + "\\My Documents\
\Altova\\MapForce2021\\MapForceExamples\\";
```

コマンドラインからサンプルを実行する コマ・ドラインからサンプルを作動します:

- 1. すべての必要条件が満たされていることを確認してくたさい(次を参照してくたさい:必要条件).
- 2. コマボプロンプトウイドウを開き、サンプル Java プロジェクト フォルダー コネル て現在のデルクトリを変更し、次を入力します:

buildAndRun.bat "<Path-to-the-Java-bin-folder>"

3. 「Enter」を押します。

MapForceContainer.java 内のJava ソースがゴン イルされ、実行されます。

Eclipse 内でサンプルをコンパイルし実行する サンプルJava プロジェクトをEclipse にインポートする

1. すべての必要条件が満たされていることを確認してくたさい(次を参照してくたさい: 必要条件).
- 2. ファイルメニューから「インポート」をクリックします。
- 3. 「ワークスペースで既存のプロジェクトを開く」を選択し、次の場所にあるEclipse プロジェクトファイルを参照します: <ApplicationFolder>¥Examples¥ActiveX¥Java¥。このフォルダーでは書き込みのアクセスがないせっか、インポートダイア ログボックス上の「ワークプレースにプロジェクトをコピーする」チェックボックスを選択することが奨励されます。

サンプルアプトケーションを作動するコよ、Package Explorer内のプロジェクトを右クトックし、コマド「実行 | Java アプリケーション」 を選択します。

コード内のコメントを使用するとJava API クラスのナメのヘンレンを使用することができ、とE clipse 内で Javadoc ビューを有効化するこは、メニューコマンド「Window |ビューの表示 | JavaDoc」を選択してくたさい。

16.5.3.2 ActiveX コントロールの作成

下のコードリストは、ActiveX コントロールの作成方法を示しています。コンストラクターは、Java ラットーオブジェクトを作成します。カンバ スにより派生したオブジェクトをパネル、まけよ、フレームに追加すると、ラップされたActiveXの生成がトリガーされます。

```
01
     /**
02
     * MapForce manager control - always needed
0.3
04
    public static MapForceControl
                                          mapForceControl = null;
05
06
    /**
    * MapForceDocument editing control
07
08
     */
                                             mapForceDocument = null;
09
    public static MapForceControlDocument
10
11
     * Tool windows - MapForce place-holder controls
12
     */
13
14
    private static MapForceControlPlaceHolder mapForceProjectToolWindow = null;
15
    private static MapForceControlPlaceHolder mapForceValidationToolWindow = null;
16
    private static MapForceControlPlaceHolder mapForceLibraryToolWindow = null;
17
    // Create the MapForce ActiveX control; the parameter determines that we want
18
     // to place document controls and place-holder controls individually.
19
     // It gives us full control over the menu, as well.
2.0
       mapForceControl = new MapForceControl(
        ICActiveXIntegrationLevel.ICActiveXIntegrationOnDocumentLevel.getValue(), false );
21
22
       mapForceDocument = new MapForceControlDocument();
23
       frame.add( mapForceDocument, BorderLayout.CENTER );
2.4
25
26
   // Create a project window and open the sample project in it
       mapForceProjectToolWindow = new MapForceControlPlaceHolder(
27
       MapForceControlPlaceholderWindow.MapForceXProjectWindow.getValue(),
        strExamplesFolder + "MapForceExamples.mfp" ) ;
28
       mapForceProjectToolWindow.setPreferredSize( new Dimension( 200, 200 ) );
```

16.5.3.3 コントロール内のデータのロード

下のコードリストは、ActiveX コトロール内にデーダセロードを表示する方法を表示しています。

1 // Locate samples installed with the product.

16.5.3.4 基本のイベントハンドリング

下のコードリストは、基本のイベトの処理方法について表示しています。MapForceControl のopen メンド、おけよ メニューおけま プロジェクトットを使用してファイルを開こうとすると、onOpenedOrFocused イベントは、アタッチ済みのイベトハンドラーに送信されま す。このイベトのナダの基本の処理により、MapForceDocumentControlのopen メンドを呼び出すことによりファイルを開きます。

```
01
         // Open the Marketing file when button is pressed
02
         btnMarkExp.addActionListener( new ActionListener() {
03
           public void actionPerformed(ActionEvent e) {
04
             try {
0.5
               // Instruct the Document control to open the file - avoid calling the open method of
MapForceControl (see help)
06
               mapForceDocument.open( strExamplesFolder + "MarketingExpenses.mfd" );
07
               mapForceDocument.requestFocusInWindow();
08
             } catch (AutomationException e1) {
09
               el.printStackTrace();
10
             }
11
           }
12
         } );
13
         public void onOpenedOrFocused( String i strFileName, boolean i bOpenWithThisControl,
boolean i_bFileAlreadyOpened ) throws AutomationException
14
    {
       // Handle the New/Open events coming from the Project tree or from the menus
15
       if ( !i bFileAlreadyOpened )
16
17
18
         \prime\prime This is basically an SDI interface, so open the file in the already existing document
control
19
         try {
           MapForceContainer.mapForceDocument.open( i_strFileName );
20
21
           MapForceContainer.mapForceDocument.requestFocusInWindow();
22
         } catch (Exception e) {
23
           e.printStackTrace();
24
         }
25
       }
26
     }
```

16.5.3.5 メニュー

下のコードリストは、メニューアイテムの作成方法を表示しています。各 MapForceCommand オブジェクトは、コマンドのID へのセントされているActionCommandを持つ、対応するMenuItemオブジェクト、を取得します。全てのメニューアイテムことと見成されているアグターンは、(閉じられたメカニズムのを再度解釈するなどの)特定の処理を行う同じ関数によや処理されます。おけま、execメントでを呼び出すことにとり MapForceControl オブジェクトに実行を委任することができます。メニューの作成中についたれる menuMap オブジェクトは、後で使用されます(次のセグションを参照してくたさい、UIアップデートイベント・ハンドレク)。

01 02 // Load the file menu when the button is pressed 03 btnMenu.addActionListener(new ActionListener() { 04 public void actionPerformed(ActionEvent e) {

```
0.5
             try {
06
               // Create the menubar that will be attached to the frame
07
               MenuBar mb = new MenuBar();
08
               // Load the main menu's first item - the File menu
09
               MapForceCommand xmlSpyMenu =
mapForceControl.getMainMenu().getSubCommands().getItem( 0 );
10
               // Create Java menu items from the Commands objects
11
               Menu fileMenu = new Menu();
               handlerObject.fillMenu( fileMenu, xmlSpyMenu.getSubCommands() );
12
13
               fileMenu.setLabel( xmlSpyMenu.getLabel().replace( "&", "" ) );
14
               mb.add( fileMenu );
15
               frame.setMenuBar( mb );
               frame.validate();
16
17
             } catch (AutomationException el) {
18
               e1.printStackTrace();
19
             }
             \ensuremath{/\!/} Disable the button when the action has been performed
20
21
             ((AbstractButton) e.getSource()).setEnabled( false );
22
           }
23
         });
     /**
24
25
      \ast Populates a menu with the commands and submenus contained in an MapForceCommands object
26
      * /
27
     public void fillMenu (Menu newMenu, MapForceCommands mapForceMenu) throws AutomationException
2.8
29
       // For each command/submenu in the mapForceMenu
30
       for ( int i = 0 ; i < mapForceMenu.getCount() ; ++i )</pre>
31
32
         MapForceCommand mapForceCommand = mapForceMenu.getItem( i );
33
         if ( mapForceCommand.getIsSeparator() )
34
          newMenu.addSeparator();
35
         else
36
           MapForceCommands subCommands = mapForceCommand.getSubCommands();
37
38
           // Is it a command (leaf), or a submenu?
           if ( subCommands.isNull() || subCommands.getCount() == 0 )
39
40
41
             // Command -> add it to the menu, set its ActionCommand to its ID and store it in the
menuMap
             MenuItem mi = new MenuItem( mapForceCommand.getLabel().replace( "&", "" ) );
42
             mi.setActionCommand( "" + mapForceCommand.getID() );
43
44
             mi.addActionListener( this );
             newMenu.add( mi );
45
             menuMap.put( mapForceCommand.getID(), mi );
46
47
           }
48
           else
49
          {
50
             // Submenu -> create submenu and repeat recursively
51
             Menu newSubMenu = new Menu();
52
             fillMenu( newSubMenu, subCommands );
53
             newSubMenu.setLabel( mapForceCommand.getLabel().replace( "&", "" ) );
54
             newMenu.add( newSubMenu );
55
           }
56
         }
57
       }
58
    }
59
     /**
      * Action handler for the menu items
60
      * Called when the user selects a menu item; the item's action command corresponds to the
61
command table for MapForce
62
     */
     public void actionPerformed( ActionEvent e )
63
64
     {
65
       try
66
       {
```

```
int iCmd = Integer.parseInt( e.getActionCommand() );
67
68
         // Handle explicitly the Close commands
69
         switch ( iCmd )
70
        {
71
          case 57602:
                             // Close
          case 34050:
                             // Close All
72
73
            MapForceContainer.initMapForceDocument();
74
            break;
75
           default:
76
             MapForceContainer.mapForceControl.exec( iCmd );
77
             break;
78
        }
79
      }
       catch ( Exception ex )
80
81
      {
82
        ex.printStackTrace();
83
       }
84
85
   }
```

16.5.3.6 UI アップデートイベントハンドリング

下のコードリストは、UI-Update イベトハンドラーの作成方法を表示しています。

```
01 /**
02
     * Call-back from the MapForceControl.
03 * Called to enable/disable commands
     */
04
05
    00verride
06
    public void onUpdateCmdUI() throws AutomationException
07
08
      // A command should be enabled if the result of queryStatus contains the Supported (1) and
Enabled (2) flags
      for ( java.util.Map.Entry<Integer, MenuItem> pair : menuMap.entrySet() )
09
10
        pair.getValue().setEnabled( MapForceContainer.mapForceControl.queryStatus( pair.getKey() )
> 2);
11
   }
12 /**
    * Call-back from the MapForceControl.
13
    * Usually called while enabling/disabling commands due to UI updates
14
1.5
     */
16
   00verride
    public boolean onIsActiveEditor( String i_strFilePath ) throws AutomationException
17
18
     {
19
      try {
20
        return
MapForceContainer.mapForceDocument.getDocument().getFullName().equalsIgnoreCase( i_strFilePath );
21
     } catch (Exception e ) {
22
        return false;
23
      }
   }
24
```

16.5.3.7 MapForce マッピングのプロパティのリスト

下のノストは、MapForce内のマピングオブジェクトがテーブルとしてどのようコロードされるか、ませまモーダルの有効化のために準備されるかについてを表示しています。

```
01 //access MapForce Java-COM bridge
02 import com.altova.automation.MapForce.*;
03 import com.altova.automation.MapForce.Component;
04 import com.altova.automation.MapForce.Enums.ENUMComponentUsageKind;
0.5
06 //access AWT and Swing components
07 import java.awt.*;
08 import javax.swing.*;
09 import javax.swing.table.*;
10
11
12 /**
13 * A simple example of a table control loading the structure from a Mapping object.
14 * The class receives an Mapping object, loads its components in a JTable, and prepares
15 * for modal activation.
16 *
17 * Feel free to modify and extend this sample.
18 *
19 * @author Altova GmbH
20 */
21 class MapForceTable extends JDialog
22 {
    /**
23
     * The table control
24
     */
25
26
    private JTable myTable;
27
     /**
2.8
     * Constructor that prepares the modal dialog containing the filled table control
29
30
     * Oparam mapping The data to be displayed in the table
31
      * @param parent Parent frame
32
33
    public MapForceTable( Mapping mapping, Frame parent )
34
35
      // Construct the modal dialog
36
      super( parent, "MapForce component table", true );
37
      // Build up the tree
38
      fillTable( mapping );
39
      // Arrange controls in the dialog
40
       setContentPane( new JScrollPane( myTable ) );
41
    }
42
43
    /**
     * Loads the components of a Mapping object in the table
44
     * @param mapping Source data
45
      */
46
47
    private void fillTable( Mapping mapping)
48
     {
49
       try
50
51
         // count how many Instance components do we have
52
        int size = 0;
53
        for (Component comp : mapping.getComponents())
54
          if ( comp.getUsageKind() == ENUMComponentUsageKind.eComponentUsageKind_Instance )
55
            +tsize:
56
57
         // Prepare data
58
         final String[] columnNames = { "Component", "Has inputs", "Has outputs", "Input file",
"Output file", "Schema" };
59
         final Object[][] data = new Object[size ][ 7 ] ;
60
         int index = 0 ;
61
         for (Component comp : mapping.getComponents())
62
           if ( comp.getUsageKind() == ENUMComponentUsageKind.eComponentUsageKind_Instance )
```

```
63
           {
64
             int i = 0;
65
             data[ index ][ i++ ] = comp.getName() ;
             data[ index ][ i++ ] = new Boolean( comp.getHasIncomingConnections() );
66
             data[ index ][ i++ ] = new Boolean( comp.getHasOutgoingConnections() );
67
68
             data[ index ][ i++ ] = comp.getInputInstanceFile();
69
             data[ index ][ i++ ] = comp.getOutputInstanceFile();
70
             data[ index++ ][ i ] = comp.getSchema() ;
71
           }
72
73
         // Set up table
74
         myTable = new JTable( new AbstractTableModel() {
75
             public String getColumnName(int col) { return columnNames[col]; }
76
             public int getRowCount() { return data.length; }
77
             public int getColumnCount() { return columnNames.length; }
78
             public Object getValueAt(int row, int col) { return data[row][col]; }
79
             public boolean isCellEditable(int row, int col) { return false; }
80
             public Class getColumnClass(int c) { return getValueAt(0, c).getClass(); }
81
         });
82
83
         // Set width
         for( index = 0 ; index < columnNames.length ; ++index )</pre>
84
85
          myTable.getColumnModel().getColumn( index ).setMinWidth( 80 );
86
         myTable.getColumnModel().getColumn( 5 ).setMinWidth( 400 );
87
      }
88
       catch (Exception e)
89
       {
90
         e.printStackTrace();
91
       }
92
    }
93
94 }
```

16.5.4 VB.NET

ソースコードは、VB.NET アプリケーションへのMapForceControl の統合を表しています。これはMapForce インストール内のフォルダ ー <ApplicationFolder>\Examples\ActiveX\VB.NET でみつねことができます。 ソリューションは、3つのカイドウにより以下のように構成されています:

- 1. MainWindow.vb メインドキュメトウイドウミ基本のアプリケーションメニューか含まれています。
- 2. LibraryWindow.vb ライブラノウイドウ。このウイドウのコンテンソは PlaceholderWindowlD プロ 守っが0 に設定されているプレースオルダーコナトロールにより作成されます(この値はコナトロールに対してライブラリウイドウェ表示するように命令します)。

🖳 LibraryWindow 💷 🔳 🗾			
	Properties		• □ ×
	AxMapForceControlPlace	Holder1 AxMapForceControlLib.	AxMapForceContr 🗸
	E 🛃 🖓 🗲 👂		
	⊞ MinimumSize	0, 0	-
	Modifiers	Friend	
	Padding	<u>0, 0, 0</u> , 0	
	PlaceholderWindowlD	0	
	⊞ Size	251, 446	
	TabIndex	0	
× .			
Search for function			
Add/Remove Libraries			

3. OutputWindow.vb - メッセージ(Output) ウイドウ。このウイドウのエンテンソは、PlaceholderWindowlD プロ、ティ がに設定されているプレースヤリダーエトロールにより作成されます(この値はエトロールに対して出カウイドウに表示するよう に命令します)。

	OutputV	Vindow	
	NS X		
• 76/5/4/3/2/			
	Properties		- □ ×
	MapForceValidation AxMapFor	rceControlLib.AxMapForceCo	ontrolPlaceHolder 👻
	₽ ₽ ₽		
	⊞ MinimumSize	0, 0	A
	Modifiers	Friend	
	Padding	0, 0, 0, 0	
	PlaceholderWindowlD	2	-
	⊞ Size	761, 133	
	TabIndex	0	-
	PlaceholderWindowID		

このノリューションをビルドして実行する前に、以下のステップに注意してください

ステップ1:必要条件をチェックする

必要条件のノストレ、関しては、次を参照してくたさい、必要条件

ステップ 2: 書き込みのパーミッションを持つ箇所でサンプルをディレクトリにコピーする

Visual Studio を管理者として作動することを回避するコよ、デフォルトの場所から実行する代われに、書き込みのパーミッションを持つディレクトリンのノースコードをコピーします。

ステップ 3: ビルドプラトフォームを設定する

- ビルド(x86、x64)するプラオフォームに一致するビルドプラオフォームの更新を作成します。ビルドの更新の作成の方法は以下のとおりです:
 - a. Visual Studio 内のノリューションを右クトックし、「更新 マネージャー」を選択します。
 - b. 「アクティブなソリューションプラットフォーム」から、「新規作成…」を選択し、x86 おけまx64 更新を選択します(このサンプルでは、x86)。

New Solution Platform
Type or select the new <u>p</u> latform:
x86 🗸
Copy <u>s</u> ettings from:
Any CPU 👻
Create new project platforms
OK Cancel

Visual Studio 内でノリューションをビノドして実行する準備か整うと、ターゲ・ナプラ・オフォーム(x86、x64)に一致するように更新をしよう してビノドすることを忘れないでください。それ以外の場合、ランタイムエラーが発生する可能性があります。

16.6 コマンド レファレンス

このセクションは、MapForce内で使用することのできる全てのメニューコマドの名前と識別子をリストしています。各サブセクションは MapForce、コマドテーブルの対応するトップレベルメニューからのコマドが以下のとおりに整理されています。

- 「メニューアイテム」列は、コマンドのメニューテキストをMapForce で表示される順番に表示します。コマンドの機能を簡単に識別することができます。
- 「コマボ名」列は、MapForce インストールデルクトリのActiveX、Images フォルダーから同じ名前を持つアイコンを取得 するかめご使用される文字列を指定します。
- 「ID」列はこのコマボを実行おけなアレオるメソボの引数とて与えられる列の数値識別子を表示します。

コマイを実行するコよ <u>MapForceControl.Exec</u> おは<u>MapForceControlDocument.Exec</u> メノイを利用します。コマンドのステータスをクロけるコよ <u>MapForceControl.QueryStatus</u> おは<u>MapForceControlDocument.QueryStatus</u> メノイドを使用します。

インストールされたMapForceのエディションにより、コマンドの一部がサポートされない場合かあります。

16.6.1 「ファイル」メニュー

「ファイル」メニューコお以下のコマドが含まれます

X=1-717L	コマバ名	ID
新規作成	ID_FILE_NEW	57600
開く	ID_FILE_OPEN	57601
保存	ID_FILE_SAVE	57603
名前を付けて保存	ID_FILE_SAVE_AS	57604
全て保存する	ID_FILE_SAVEALL	32377
再中卡	IDC_FILE_RELOAD	32467
閉じる	ID_WINDOW_CLOSE	32453
全て閉じる	ID_WINDOW_CLOSEALL	32454
印刷	ID_FILE_PRINT	57607
	ID_FILE_PRINT_PREVIEW	57609
印刷設定	ID_FILE_PRINT_SETUP	57606
マピングの検証	ID_MAPPING_VALIDATE	32347
マピングの設定	ID_MAPPING_SETTINGS	32396
選択された言語でコードを生成する	ID_FILE_GENERATE_SELECTED_CODE	32362
XSLT 1.0	ID_FILE_GENERATEXSLT	32360

	コマバ名	ID
XSLT 2.0	ID_FILE_GENERATEXSLT2	32361
XQuery	ID_FILE_GENERATEXQUERY	32359
Java	ID_FILE_GENERATEJAVACODE	32358
C# (Sharp)	ID_FILE_GENERATECSCODE	32357
C++	ID_FILE_GENERATECPPCODE	32356
MapForce Server 実行ファイルのエンパル	ID_FILE_CREATE_SERVER_EXECUTION_FILE	32517
FlowForce Server にデプロイする	ID_FILE_DEPLOY_MAPPING	32506
ドキュメトの生成	ID_FILE_GENERATE_DOCUMENTATION	32468
最近使オシアイル	ID_FILE_MRU_FILE1	57616
終了	ID_APP_EXIT	57665

16.6.2 「編集」メニュー

「編集」メニューココントのコマンドか含まれます

メニューアイテム	コマバ名	ID
元に戻す	ID_EDIT_UNDO	57643
や直し	ID_EDIT_REDO	57644
検索	ID_EDIT_FIND	57636
次を検索	ID_EDIT_FINDNEXT	32349
前を検索	ID_EDIT_FINDPREV	32350
切り取り	ID_EDIT_CUT	57635
ヹ ー	ID_EDIT_COPY	57634
貼り付け	ID_EDIT_PASTE	57637
削除	ID_EDIT_CLEAR	57632
全て選択	ID_EDIT_SELECT_ALL	57642

16.6.3 「挿入」メニュー

「挿入」メニューコお以下のコマドが含まれます

メニューアイテム	コマボ名	ID
XML スキーマ/ファイル	ID_INSERT_XSD	32393
7-9~- 7	ID_INSERT_DATABASE	32389
EDI	ID_INSERT_EDI	32390
テキストファイル	ID_INSERT_TXT	32392
ウェブサービス関数	ID_INSERT_WEBSERVICE_FUNCTION	32319
Excel 2007+ ファイル	ID_INSERT_EXCEL	32376
XBRLドキュメント	ID_INSERT_XBRL	32469
JSON スキーマ/ファイル	ID_INSERT_JSON	32531
入力の挿入	ID_FUNCTION_INSERT_INPUT	32383
出力の挿入	ID_FUNCTION_INSERT_OUTPUT	32402
定数	ID_INSERT_CONSTANT	32388
変数	ID_INSERT_VARIABLE	32500
ぶつ	ID_INSERT_JOIN	32581
並 べ 替え Nodes/Rows	ID_INSERT_SORT	32444
ブルター: Nodes/Rows	ID_INSERT_FILTER	32391
SQL-WHERE/ORDER	ID_INSERT_SQLWHERE_CONDITION	32351
Value-Map	ID_INSERT_VALUEMAP	32354
IF-Else 条件	ID_INSRT_CONDITION	32394
例外	ID_INSERT_EXCEPTION	32311

16.6.4 「プロジェクト」メニュー

「プロジェクト」メニューコお以下のコマイが含まれます

メ⊏ュ─アイテム	コマズ名	ID
プロジェクトを再ロードする	ID_PROJECT_RELOAD	32476

メニューアイテム	コマバ名	ID
プレジェクトを閉じる	ID_FILE_CLOSEPROJECT	32355
プレジェクトを保存する	ID_FILE_SAVEPROJECT	32378
プレジェクトレファイルを追加する	ID_PROJECT_ADDFILESTOPROJECT	32420
アクティブなファイルをプロジェクトに追加する	ID_PROJECT_ADDACTIVEFILETOPROJECT	32419
フォルダーの作成	ID_PROJECT_CREATE_FOLDER	32310
マピングを開く	ID_PROJECT_OPEN_MAPPING	32307
オペレーションマパングの作成	ID_PROJECT_CREATE_MAPPING_FOR_OPE RATION	32399
オペレーションマンピングファイルの追加	ID_PROJECT_ADD_MAPPING	32309
ウェブサービスを挿入	ID_PROJECT_INSERT_WEBSERVICE	32306
XMLSpy のファイルを開く	ID_PROJECT_OPEN_IN_XMLSPY	32305
プレジェケト全体のコード生成	ID_PROJECT_GENERATE_ALL	32303
XSLT 1.0	ID_PROJECT_GENERATEXSLTCODE_ENTIR E	32408
XSLT 2.0	ID_PROJECT_GENERATEXSLT2CODE_ENTI RE	32409
XQuery	ID_PROJECT_GENERATEXQUERYCODE_EN TIRE	32410
Java	ID_PROJECT_GENERATEJAVACODE_ENTIR E	32411
C# (Sharp)	ID_PROJECT_GENERATECSCODE_ENTIRE	32412
C++	ID_PROJECT_GENERATECPPCODE_ENTIR E	32413
プロ / ኻィ	ID_PROJECT_PROPERTIES	32404
最近使 オンプロジェクト	ID_FILE_MRU_PROJECT1	32364

16.6.5 「コンポーネント」メニュー

「コンポーネント」メニューココンドのコマンドが含まれます

	コマバ名	ID
ル十要素の変更	ID_COMPONENT_CHANGEROOTELEMENT	32334

メニューアイテム	コマバ名	ID
XMLSpy でスキーマ定義を編集	ID_COMPONENT_EDIT_SCHEMA	32337
FlexText 構成の編集	ID_COMPONENT_EDIT_MFT	32301
データベースオブジェクトの追加/削除/編集	ID_COMPONENT_SELECTTABLES	32346
EDI X12 997 へのマピングを作成	ID_COMPONENT_CREATE_MAPPING_TO_99 7	32483
EDI X12 999 へのマンプを作成	ID_COMPONENT_CREATE_MAPPING_TO_99 9	32484
更新	IDC_COMMAND_REFRESH_COMPONENT	32373
前に入力を複製する	ID_COMPONENT_CREATE_DUPLICATE_ICO N_BEFORE	32503
後に入力を複製する	ID_COMPONENT_CREATE_DUPLICATE_ICO N	32335
複製を削除	ID_COMPONENT_REMOVE_DUPLICATE_ICO N	32339
前にコメントを追加する	ID_COMPONENT_ADD_COMMENT_BEFORE	32518
後にコメントを追加する	ID_COMPONENT_ADD_COMMENT_AFTER	32519
前に処理命令を追加する	ID_COMPONENT_ADD_PI_BEFORE	32520
後に処理命令を追加する	ID_COMPONENT_ADD_PI_AFTER	32521
処理命令を編集する	ID_COMPONENT_EDIT_PI	32524
コメナイ処理命令を削除する	ID_COMPONENT_REMOVE_COMMENT_PI	32522
CDATA セグタンとしてコンテングを書き込む	ID_COMPONENT_TOGGLE_CDATA	32525
データベーステーブルアクション	ID_POPUP_DATABASETABLEACTIONS	32400
Query データベース	ID_QUERY_DATABASE	32341
ツーを左揃えにする	ID_COMPONENT_LEFTALIGNTREE	32338
ツーを右揃えにする	ID_COMPONENT_RIGHTALIGNTREE	32340
プロ/ テティ	ID_COMPONENT_PROPERTIES	32336

16.6.6 「接続」メニュー

「接続」メニューココントのコマンドが含まれます

	コマバ名	ID
マッチした子に自動的に接続	ID_CONNECTION_AUTOCONNECTCHILDREN	32342
マッチした子に接続のための設定	ID_CONNECTION_SETTINGS	32344
マッチした子に接続	ID_CONNECTION_MAPCHILDELEMENTS	32343
ターゲト優先(標準マピング)	ID_POPUP_NORMALCONNECTION	32401
全て⊐ピー(子アイテムを⊐ピー)	ID_POPUP_NORMALWITHCHILDREN_CONN ECTION	32460
ソース優先マピング複合型コンテンソ	ID_POPUP_ORDERBYSOURCECONNECTION	32403
プロ/ የጉイ	ID_POPUP_CONNECTION_SETTINGS	32398

16.6.7 「関数」メニュー

「関数」メニューコお以下のコマイが含まれます

X=1-717L	コマイ名	ID	
ユーザー定義関数の作成	ID_FUNCTION_CREATE_EMPTY	32380	
選択からユーザー定義関数の作成	ID_FUNCTION_CREATE_FROM_SELECTION	32381	
関数設定	ID_FUNCTION_SETTINGS	32387	
関数削除	ID_FUNCTION_REMOVE	32385	
入力の挿入	ID_FUNCTION_INSERT_INPUT	32383	
出力の挿入	ID_FUNCTION_INSERT_OUTPUT	32402	

16.6.8 「出力」メニュー

「出力」メニューコお以下のコマドが含まれます

	コマバ名	ID
XSLT 1.0	ID_SELECT_LANGUAGE_XSLT	32433
XSLT 2.0	ID_SELECT_LANGUAGE_XSLT2	32434
XQuery	ID_SELECT_LANGUAGE_XQUERY	32432
Java	ID_SELECT_LANGUAGE_JAVA	32431
C# (Sharp)	ID_SELECT_LANGUAGE_CSHARP	32430

メニューアイテム	コマバ名	ID
C++	ID_SELECT_LANGUAGE_CPP	32429
ビルイ実行エジン	ID_SELECT_LANGUAGE_BUILTIN	32490
出カファイルの検証	ID_XML_VALIDATE	32458
出カファイルの保存	IDC_FILE_SAVEGENERATEDOUTPUT	32321
全ての出力ファイルの保存	IDC_FILE_SAVEALLGENERATEDOUTPUT	32374
出力を再生成	ID_REGENERATE_PREVIEW_OUTPUT	32480
SQL スクリプトを実行	ID_TRANSFORM_RUN_SQL	32442
ブックマークの挿入削除	ID_TOGGLE_BOOKMARK	32317
次のブッケマーク	ID_GOTONEXTBOOKMARK	32315
前のブックマーク	ID_GOTOPREVBOOKMARK	32314
全てのブッケマークの削除	ID_REMOVEALLBOOKMARKS	32313
XML テキストの整形出力	ID_PRETTY_PRINT_OUTPUT	32363
テキストビューの設定	ID_TEXTVIEWSETTINGSDIALOG	32472

16.6.9 「デバッグ」メニュー

「デバッグ」メニューはお以下のコマイが含まれます

メニューアイテム	コマバ名	ID
デッジプの開始	ID_DEBUG_START	32540
デッジプの停止	ID_DEBUG_STOP	32541
ステップイン	ID_DEBUG_STEP_INTO	32545
	ID_DEBUG_STEP_OVER	32551
ステップアサ	ID_DEBUG_STEP_OUT	32552
最小ステップ	ID_DEBUG_STEP_NEXT_TRACE	32554

16.6.10 「表示」メニュー

「表示」メニューーコお以下のコマンドが含まれます

メニューアイテム	コマバ名	ID
注釈の表示	ID_SHOW_ANNOTATION	32435
型の表示	ID_SHOW_TYPES	32437
関数のライブラル名を表示	ID_VIEW_SHOWLIBRARYINFUNCTIONHEAD ER	32448
ビトの表示	ID_SHOW_TIPS	32436
XBRL 表示オプション	ID_VIEW_XBRL_DISPLAY_OPTIONS	32473
選択エパーネトの接続線の表示	ID_VIEW_AUTOHIGHLIGHTCOMPONENTCON NECTIONS	32443
ソースからターゲトへの接続線の表示	ID_VIEW_RECURSIVEAUTOHIGHLIGHT	32447
<i>х</i> — <i>ь</i>	ID_VIEW_ZOOM	32451
戻る	ID_CMD_BACK	32479
進む	ID_CMD_FORWARD	32478
ステータスバー	ID_VIEW_STATUS_BAR	59393
ライブラリウィドウ	ID_VIEW_LIBRARY_WINDOW	32445
メッセージ	ID_VIEW_VALIDATION_OUTPUT	32450
概要	ID_VIEW_OVERVIEW_WINDOW	32446
プレジェクトウィンドウ	ID_VIEW_PROJECT_WINDOW	32302
値	ID_DEBUG_VIEW_VALUES_WINDOW	32544
エレテキスト	ID_DEBUG_VIEW_CONTEXT_WINDOW	32546
	ID_DEBUG_VIEW_DEBUGPOINTS_WINDOW	32547

16.6.11 「ツール」メニュー

「ツールメニューコお以下のコマドが含まれています

メニューアイテム	コマバ名	ID
グローイ シリンノース	IDC_GLOBALRESOURCES	37401
	IDC_GLOBALRESOURCES_SUBMENUENTRY 1	37408
反転マピングの作成	ID_CREATE_REVERSED_MAPPING	32489
カスタマイズ	IDC_APP_TOOLS_CUSTOMIZE	32959

メニューアイテム	コマバ名	ID
	ID_TOOLS_OPTIONS	32441

16.6.12 「ヘルプ」メニュー

「ウイボウ」メニューコお以下のコマボが含まれます

メニューアイテム	コマバ名	ID
重ねて表示	ID_WINDOW_CASCADE	57650
上下に並べて表示	ID_WINDOW_TILE_HORZ	57651
左右に並べて表示	ID_WINDOW_TILE_VERT	57652

16.6.13 「ヘルプ」メニュー

「ヘリプ」メニューコお以下のコマドが含まれます

*=1-7174	コマバ名	ID
目次	IDC_HELP_CONTENTS	32966
んデックス	IDC_HELP_INDEX	32967
検索	IDC_HELP_SEARCH	32969
ライセスの登録	IDC_ACTIVATION	32970
注文フォーム	IDC_OPEN_ORDER_PAGE	32971
登録	IDC_REGISTRATION	32972
最新情報のチェック	IDC_CHECK_FOR_UPDATES	32973
MapForce 製品比較	IDC_PRODUCT_COMPARISON	32955
ザポーセター	IDC_OPEN_SUPPORT_PAGE	32961
W eb のFAQ	IDC_SHOW_FAQ	32962
ユポーネトや無料ツールのダウロード	IDC_OPEN_COMPONENTS_PAGE	32963
インターネオ上のMapForce	IDC_OPEN_XML_SPY_HOME	32964
MapForce トレーニング	IDC_OPEN_MAPFORCE_TRAINING_PAGE	32965
MapForce ISIT	ID_APP_ABOUT	57664

16.7 オブジェクトレファレンス

オブジェクト:

MapForceCommand MapForceCommands MapForceControl MapForceControlDocument MapForceControlPlaceHolder

標準のMapForce 機能へのアクセスを与えるコよ MapForce 自動化インターフェイスのオブシェクトにもアクセスすることができます。 詳細に関しては次を参照してくたさい: <u>MapForceControl.Application</u>, <u>MapForceControlDocument.Document</u> と <u>MapForceControlPlaceHolder.Project</u>。

16.7.1 MapForceCommand

プロパティ: ID Label Name IsSeparator ToolTip StatusText Accelerator SubCommands

詳細:

コマドオブジェクトは次のオプションであることができます:実行可能なコマド、コマド コンテナー(例えば、メニュー、サブメニュー、おこ は、ツールドー)、おけよ メニューセ レーター。現在のCommand オブジェクト内に保管される情報の種類を決定するけよ、ID、 IsSeparator、とSubCommands プロ ディをクロルてください。

コマンドオブジェクト	使用可能な場合
実行可能なコマイ	 ID はわよれ大き、場合 IsSeparator がfalse の場合 SubCommands か空白の場合
コマイ コンテナー	 ID はせつ場合 IsSeparator がtrue の場合 SubCommands がCommand オブジェクトを含む場合
セレーター	 ID はむの場合 IsSeparator がtrue の場合

16.7.1.1 Accelerator

プロパティ: Accelerator をstring とて

詳細:

コマドのために定義されたアクセラレータキーを返します。コマドにアクセラレーターキーが割り当てられていない場合、このプロパティは、空の文字列を返します。アクセラレータキーの文字列の表示は次のフォーマナを取ります:

[ALT+][CTRL+][SHIFT+]key

Windows プラオフォーム SDK 関数 GetKeyNameText を使用して key か変換された箇所。

16.7.1.2 ID

プロパティ: IDをlong とて

詳細:

このプロ・ティは、コマドの一意の識別子を取得します。コマドのID は <u>Exec</u>を使用して) コマドを実行するさなに、おさよ (<u>QueryStatus</u>を使用して)ステータスをクロリするさめつ必須です。コマドが他のコマド (例えば、トップレベルのメニュー)のさめのコン テナー、まさは、セッレーターの場合、ID は0 です。

16.7.1.3 IsSeparator

プロパティ: IsSeparator をboolean とて

詳細:

コマドオブジェケがメニューセルーターの場合、の場合、プロゲイは、trueを返します。それ以外の場合は、falseを返します。次も参照してくたさい、コマンド

16.7.1.4 Label

プロパディ: Label をstring とて

詳細:

このプロンティは、MapForceのグランイカルなユーザーインターフェイス内に表示されるとおりのコマンドのテキストを取得します。コマンドがセンレーターの場合、「ラベシレ」は、空の文字列です。このプロンティは、関連したGUIテキストを持たない、一部のソールレーコマンドのために空の文字列を返します。

16.7.1.5 Name

プロパディ: Name をstring 出て

詳細:

このプロ・ティオー意のコマド名を取得します。使用できる箇所で、この値はコマドのアイコンファイルを取得するために使用できます。使用 することのできるアイコンファイルは、使用中のMapForce インストールのフォルダー <ApplicationFolder>\Examples\ActiveX\Images 内におます。

16.7.1.6 StatusText

プロパディ: Label をstring とて

詳細:

(C) 2015–2021 Altova GmbH

コマドが選択されると、ステータステキストは、MapForceのステータスバー内で表示されるテキストです。他のコマドのセルーターまたは、コンテナオではないロマドオブジェクトに対してのみ適用することができます。それ以外の場合、プロパティは空の文字列です。

16.7.1.7 SubCommands

プロパディ: SubCommands をCommands とて

詳細:

SubCommands プロ 守れよ 現在のコマドのサブコマドである Command オブシェクト のコレクションを取得します。プロ 守れよ 他のコ マド(メニュー、サブメニュー、 おけよ ツール シー のけっかのコレテナーであるコマド にのみ適用されます。 このようはコレテナーコマド は 0 に 設定された ID を持ち、および、 false に設定された IsSeparator プロ 守を持ちます。

16.7.1.8 ToolTip

プロパディ: ToolTipをstring とて

詳細:

このプロレティは、各コマドのオメのビトを表示するテキストを取得します。コマドがソールのビトのテキストを持たない場合、プロレティは、空の文字列を返します。

16.7.2 MapForceCommands

プロパティ:

<u>Count</u>

Item

詳細:

MapForceControlのコマイレベルとID へのアクセスを取得するすめの Command オブシェクトのコレクションです。これらのコマイドは、 Exec メノバを使用して実行することができます。 おこ、ステータスをQueryStatusを使用してクロノすることができます。

16.7.2.1 Count

プロパディ: Count をlong とて

詳細: このコレグィンのレベル上のCommand オブジェクトの数。

16.7.2.2 Item

プロパティ: Item (nをlong とのをCommandとて

詳細:

このコレグション内のインデックスnを持つコマンドを取得します。インデックスは1 ベースです。

16.7.3 MapForceControl

プロパティ:

IntegrationLevel Appearance Application BorderStyle CommandsList EnableUserPrompts MainMenu Toolbars

メソッド:

<u>Open</u> <u>Exec</u> <u>QueryStatus</u>

イベント:

OnUpdateCmdUI OnOpenedOrFocused OnCloseEditingWindow OnFileChangedAlert OnContextChanged OnDocumentOpened OnValidationWindowUpdated

このオブシェクトは完全な ActiveX コトロールで、MapForce ライブラルアプリケーションベルモードで使用されている場合、表示されます。

CLSID: A38637E9-5759-4456-A167-F01160CC22C1 ProgID: Altova.MapForceControl

16.7.3.1 プロパティ

以下のプロケウカ定義されています

IntegrationLevel EnableUserPrompts Appearance BorderStyle

コマドに関連したプロッティ

CommandsList MainMenu Toolbars

MapForceAPI へのアクセス: Application

16.7.3.1.1 Appearance

プロパティ: Appearance をshort とて

ティスペやチID: -520

詳細:

0 マオレて等しくない値は、コトローノの周りでクライアトエッジを表示します。デフォルトの値は0 です。

16.7.3.1.2 Application

プロパディ: Application をApplication とて

*ディス・*シチ*ID:*1

詳細:

Application プロ、ティは、完全な MapForce 自動サー、 API の Application オブジェクト へのアクセスを与えます。プロ 、ティオ読み取り専用です

16.7.3.1.3 BorderStyle

プロディ: BorderStyleをshort とて

ディス/ やチID: -504

詳細:

1の値は薄いボーダーを持つコントローノを表示します。デフォルトの値は0です。

16.7.3.1.4 CommandsList

プロパティ: CommandList をCommands とて (読み取り専用)

ティス\ พิFID: 1004

詳細:

このプロ・ティは、MapForceControlを持つ定義された全てのコマドのフラオリストを返します。メニュー構造に従い、コマドを整理する には、MainMenuを使用します。ツール・コマドを取得すること、Toolbarsを使用します。

```
public void GetAllMapForceCommands()
{
    // Get all commands from the MapForce ActiveX control assigned to the current form
    MapForceControlLib.MapForceCommands commands = this.axMapForceControll.CommandList;
    // Iterate through all commands
    for (int i = 0; i < commands.Count; i++)</pre>
```

```
{
   // Get each command by index and output it to the console
   MapForceControlLib.MapForceCommand cmd = axMapForceControl1.CommandList[i];
   Console.WriteLine("{0} {1} {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&", ""));
}
```

C# サンプル

}

16.7.3.1.5 EnableUserPrompts

プロパティ: EnableUserPrompts をboolean とて

รีสวง พร ID : 1006

詳細:

このプロレティを false に設定すると、コトロール内のユーザープロンプトを無効化します。デフォルトの値は true です。

16.7.3.1.6 IntegrationLevel

プロディ: IntegrationLevel をICActiveXIntegrationLevel とて

รีสุวง พิร ID : 1000

詳細:

The IntegrationLevel プロ、ティ determines エトロールのオペレーションモード。次も参照してくたさい、アプリケーションベルレの統合 とドキュメトレベルの統合。

メモ MapForceControl オブジェクトの作成後すくに、このプロッティを設定することは重要です。

16.7.3.1.7 MainMenu

プロディ: MainMenu を Command とて読み取り専用)

รีสุวง พิร.ID : 1003

詳細:

このプロシティは、MapForceControl メインメニュー内で使用することのできる構造の情報とコマイを Command オジシェクト とて提供 します。オジシェクトは、MapForceの使用することのできる全てのサブメニューを含んでします(例「ファイル」、「編集」、「表示」など)。サ ブメニューオブシェクトにアクセスするゴは、MainMenu プロ、ティの SubCommands プロ、ティを使用します。各サブメニューは Command オ ブシェクト でもみます。各サブメニューのかがに、対応する子コマイド セ レーターを取得するかがに、SubCommands プロ、ティを使用して更 に反復することができます(このテクニックが使用される可能性があります。例えば、アプリケーションメニューをプログラム的に作成するなど)。メ ニューコマイドの一部は、SubCommands プロ、ティを持つ他のメニューコマイドのすかのコンテナー(「親」)としての没割を果たします。全ての メニューコマイドの構造をプログラム的に取得するゴは、取得されたコマンドの情報。内のC#で説明されている通り、再帰的な関数を作成 する必要があります。

```
public void GetMapForceMenus()
{
    // Get the main menu from the MapForce ActiveX control assigned to the current form
    MapForceControlLib.MapForceCommand mainMenu = this.axMapForceControll.MainMenu;

    // Loop through entries of the main menu (e.g. File, Edit, etc.)
    for (int i = 0; i < mainMenu.SubCommands.Count; i++)
    {
        MapForceControlLib.MapForceCommand menu = mainMenu.SubCommands[i];
        Console.WriteLine("{0} menu has {1} children items (including separators)",
        menu.Label.Replace("&", ""), menu.SubCommands.Count);
    }
}</pre>
```

C# サンプル

16.7.3.1.8 Toolbars

プロデイ: Toolbars を Commands とて (読み取り専用)

ディス/ やチID: 1005

詳細:

このプロシティは、MapForceControl ツールシーの構造の情報をCommandオブジェクトとして与えます。Commandオブジェクトには、 MapForceの使用することのできる全てのソールシーが含まれています。ツールシーニアクセスするには、Toolbars プロシティの SubCommands プロシティを使用します。各ツールシーは、Commandオブジェクトでもあります。各ツールシーのために、コマンドを使用す なっかにSubCommands プロシティを更に反復します(例えばアプリケーションのツールシープログラム的に作成するオンなどにこのテクニックは使用される場合があります)。

```
public void GetMapForceToolbars()
    // Get the application toolbars from the MapForce ActiveX control assigned to the
current form
    MapForceControlLib.MapForceCommands toolbars = this.axMapForceControl1.Toolbars;
    // Iterate through all toolbars
    for (int i = 0; i < toolbars.Count; i++)</pre>
      MapForceControlLib.MapForceCommand toolbar = toolbars[i];
      Console.WriteLine();
      Console.WriteLine("The toolbar \"{0}\" has the following commands:",
toolbar.Label);
      // Iterate through all commands of this toolbar
      for (int j = 0; j < toolbar.SubCommands.Count; j++)</pre>
      {
         MapForceControlLib.MapForceCommand cmd = toolbar.SubCommands[j];
         // Output only command objects that are not separators
         if (!cmd.IsSeparator)
            Console.WriteLine("{0}, {1}, {2}", cmd.ID, cmd.Name, cmd.Label.Replace("&",
 ""));
         }
      }
    }
```

C# サンプル

16.7.3.2 メソッド

以下のメンドが定義されます

<u>Open</u> <u>Exec</u> <u>QueryStatus</u>

16.7.3.2.1 Exec

Method: Exec (nCmdID をlong とて)をboolean とて

ディス やチID: 6

詳細:

このケンドは ID nCmdID を持つMapForce コマドを呼び出し、コマドか実行可能な場合、メンドは true を返します。使用 可能な全てのコマドのノスを取得するコよ CommandsList を使用してくたさい。.コマドの状態を取得するコよ QueryStatus を使用してくたさい。

16.7.3.2.2 Open

Method: Open (strFilePathをstring 出の をboolean 出て

*ディスト*シチID:5

詳細:

メンドの結果は、引数 strFilePath 内で やされる拡張子により異なります。ファイル拡張子が.sps の場合、新規ドキュメントが開かれます。ファイル拡張子が.svp の場合、対応するプロジェントが開かれます。異なるファイル拡張子がシンドにつ やされると、エトロールはファイルを新規のコンポーネントとしてアクティブぶドキュメントにコードします。

ドキュメトレベルの統合モド内でエトロール使用する際、このメンドを使用して、ドキュメトますまプロジェクトをロードしないでけきい。 代わりに<u>MapForceControlDocument.Open</u>と<u>MapForceControlPlaceHolder.OpenProject</u>を使用してください。

16.7.3.2.3 QueryStatus

Method: QueryStatus (nCmdID をlong とのをlong とて

*ティス\ ฃ*チ*ID*:7

詳細:

QueryStatus は、nCmdID には指定されたコマドのチェックされたチェックが解除されたステータスを返します。このステータスはビットマスクとして返されます。

- 0 1 サポートされているコマイがサポートされている場合設定します。
- 1 2 有効化されているコマイが有効化されている場合設定します(実行可能)。
- 2 4 チェックされている コマンドかチェックされている場合設定します。

これは、QueryStatus が0 コマイを返すと、ID は、有効な MapForce コマイとして認識されないことを意味します。 QueryStatus が1 おけよ5 の値を返すとコマイドは無効化されます。

16.7.3.3 Events

MapForceControl ActiveX コトロールは次の接続ポイトイベトを提供します:

OnUpdateCmdUI OnOpenedOrFocused OnCloseEditingWindow OnFileChangedAlert OnContextChanged

OnDocumentOpened OnValidationWindowUpdated

16.7.3.3.1 OnCloseEditingWindow

イベント: OnCloseEditingWindow (i_strFilePathをString 出て)をboolean 出て

รัสวง พิร.ID : 1002

詳細:

MapForce か既に開かれているドキュメントを閉じる場合、このイベントはトリガーされます。このイベントへの応答として、ケライアントは i_strFilePath に関連するエディターウィンドウを閉じます。このイベントがtrue を返すと、ケライアントがドキュメントを閉じたことを指します。 特定の処理が必要とされておらず、MapForceControl がエディターを閉じようと試み、関連したドキュメントコントロールを破棄する場合、 ケライアント は、false を返すことができます。

16.7.3.3.2 OnContextChanged

イベナ: OnContextChanged (i_strContextNameをString とて、i_bActiveをbool とのをbool とて

ティス\ พิFID : 1004

詳細: ごハベトは is not used in MapForce

16.7.3.3.3 OnDocumentOpened

イベント: OnDocumentOpened (objDocument をDocument とつ

ディスノ シチ ID: 1

詳細:

このイベトはドキュメトカ開かれる都度トリガーされます。引数 objDocument は、MapForce 自動化イクターフェイス からの Document オブジェクトであり、ドキュメトの詳細をクエリするためでき、または、追加オペレーションを行います。ドキュメ ントレベルで統合される場合、イベト <u>MapForceControlDocument.OnDocumentOpened</u> を代わりに使用することが奨 励されます。

16.7.3.3.4 OnFileChangedAlert

イベント: OnFileChangedAlert (i strFilePathをString とて)をbool とて

รีสุวง พิร ID : 1001

詳細:

MapForceControlを使用してファイルがロードされると、isこのイベトーはトリガーされます。他のアプリケーションによりレードディスク上で変更されます。 イベトを処理すると クライアントは、trueを返します。 おうは、MapForceが通常の方法で処理すると、falseを返します。 例えば、ユーザーに再ロードを促すなど。

16.7.3.3.5 OnLicenseProblem

イベナ: OnLicenseProblem (i strLicenseProblemTextをString とつ

รีสุวง พร ID : 1005

詳細:

MapForceControl がのエトロールのオタの有効なライセンを使用できないことを検知すると、このイベトーは・リガーされます。制限されているユーザーファイセンスの場合、エトロールが初期化されると発生する可能性があります。インテグレーターはこのイベトをしようして、このエトロールの機能へのアクセンを無効化することができます。このイベトから戻ると、エトロールは機能へのアクセンをブロックします(例、 エトロールの機能へのアクセンを無効化することができます。このイベトから戻ると、エトロールは機能へのアクセンをブロックします(例、

16.7.3.3.6 OnOpenedOrFocused

イベナ: OnOpenedOrFocused (i_strFilePath をString 出て、i_bOpenWithThisControl をbool 出 つ

*ティスペ*୬*チ<i>ID* : 1000

詳細:

アプリケーションレベルで統合する場合、このイベトはクライアントレギキュメントが開かれたこと、おけるMapForce によりアクティブに設定されたことを通知します。

ドキュメナレベルで統合する場合、このイベナトは、クライアナービギュメナウイボウをファイルi_strFilePathで開くように命令します。ファイルが既に開かれている場合、対応するドキュメナウイボウをアクティブなウイボウエーてください。

i_bOpenWithThisControlがtrueの場合、内部アクセスが必要とされるため、ドキュメントは、MapForceControlにと開たのであります。それ以外の場合、ファイルは、異なるエディターで開くことができます。

16.7.3.3.7 OnToolWindowUpdated

イベント: OnToolWindowUpdated(pToolWnd をlong とて)

รีสุวง พิร.ID : 1006

詳細: ツールウィドウか更新されると、このイベトはトリガーされます。

16.7.3.3.8 OnUpdateCmdUI

イベント: OnUpdateCmdUI ()

รีสุวง พิร ID : 1003

詳細: 頻繁に呼び出され、<u>MapForceControl.QueryStatus</u>を使用して MapForce コマドの状況をチェックする良い機会をイン テグレーターに与えます。このコールレジク内でlong オペレーションを実行しないでください。

16.7.3.3.9 OnValidationWindowUpdated

イベント: OnValidationWindowUpdated ()

ディス/ シチID:3

詳細: 新しい情報と共に検証出カウイドウが更新されると、このイベトーはトリガーされます。

16.7.4 MapForceControlDocument

プロパティ:

Appearance BorderStyle Document IsModified Path ReadOnly

メソッド:

Exec New Open QueryStatus Reload Save SaveAs イベント:

OnDocumentOpened OnDocumentClosed OnModifiedFlagChanged OnContextChanged OnFileChangedAlert OnActivate

MapForceControlがキュメトレベルモド内で統合されている場合、各ドキュメントは、型 MapForceControlDocument のオブジェクト内で表示されています。MapForceControlDocument には、1度につつバキュメントのみか含まれていますが、場合によっては異なるファイルを表示するために使用される場合があります。

このオブシェクトは完全な ActiveX コトロールです。

CLSID: DFBB0871-DAFE-4502-BB66-08CEB7DF5255 ProgID: Altova.MapForceControlDocument

16.7.4.1 プロパティ

以下のプロ、ティカ定義されています

ReadOnly IsModified Path Appearance BorderStyle

MapForceAPI へのアウセス: Document

16.7.4.1.1 Appearance

プロパティ: Appearance をshort とて

*ティス\ํ*୬*チID: -*520

詳細:

0 に対して等しくない値はドキュメントコトローノの周リニクライアトエッジを表示します。デフォルトの値は0 です。

16.7.4.1.2 BorderStyle

プロパティ: BorderStyle をshort とて

*ティス\ํ*พ*FID:-*504

詳細:

1の値は薄いボーダー持つエトロールを表示します。デフォルトの値は0 です。

16.7.4.1.3 Document

プロッティ: Document がキャメトとて

*ディス や*チ ID : 1

詳細:

Document プロノティは、MapForce 自動サーイーAPIのDocument オブジェクト へのアクセスを与えます。このインターフェイス は、コントロール内にコードされるドキュメントと使用することのできる追加機能を与えます。プロノティは読み取り専用です

16.7.4.1.4 IsModified

プロパティ: IsModified をboolean とて(読み取り専用)

*ディスト*シチ*ID*: 1006

詳細:

ドキュメトコンテン防最後に開かれてから変更された場合、IsModifiedはtrueを返します。再ロードし、オペレーションを保存します。それ以外の場合は、falseを返します。

16.7.4.1.5 Path

プロパディ: Path をstring とて

Fixi พ.FID: 1005

詳細: コトロール内ロードされナジキュメトのフルマ名に設定、おけよ取得します。

16.7.4.1.6 ReadOnly

プロパディ: ReadOnly をboolean とて

*Fix พิ*F*ID*: 1007

詳細:

このプロ、テクを使用すると、ドキュメトの読み取り専用モードをオンとオスこ切り替えることができます。ReadOnlyがtrueの場合、変更はできないことに注意してくざさい。

16.7.4.2 メソッド

以下のメソドが定義されています

ドキュメナハンドリング: New Open Reload Save SaveAs

コマドハンドリング::

Exec QueryStatus

16.7.4.2.1 Exec

メンッド: Exec (nCmdIDをlong とのをboolean とて

ティスレ ѷチID: 8

詳細:

このメンドは、ID nCmdID を持つMapForce コマイを呼び出し、コマイカ実行可能な場合、メンドは、true を返します。現在アクティブボドキュメトカアプリケーションで使用できる場合のみ、このメンドを呼び出すことができます。

メニュー構造に従いコマドを整理すること、MapForceControlのMainMenu プロティを使用します。ツール・コマドを取得する によ、MapForceControlのToolbars プロティを使用します。

16.7.4.2.2 New

メンッド: New()をboolean とて

รีาวง พิร ID : 1000

詳細:

このメソドはコントロール内で新規ドキュメントを初期化します。

16.7.4.2.3 Open

メソッド: Open (strFileName をstring とて) をboolean とて

รัาวง พร ID : 1001

詳細:

OpenはファイルstrFileNameを新規のドキンメトとて、コトロールコードします。

16.7.4.2.4 QueryStatus

メソッド: QueryStatus (nCmdID をlong とて) をlong とて ディス・ ゆチID: 9

詳細:

QueryStatus は、nCmdID にお指定されたコマドのチェックされた/チェックが解除されたステータスを返します。このステータスはビットマスクとして返されます。

ビット 	值	名前	意味
0	1	サポートされている	コマイドがサポートされている場合設定します。
1	2	有効化されている	コマイドがす効化されている場合設定します(実行可能)。
2	4	チェックされている	コマイドが有効化されている場合設定します(実行可能)。

これは、QueryStatusが0コマドを返すと ID は、有効なMapForceコマドとして認識されないことを意味します。 QueryStatus が1 おけよ5 の値を返すとコマドは無効化されます。アプリケーション内で現在アクティブボキュメトか存在する場合、クライアトはドキュメトナコトロールのQueryStatus メノバを呼び出します。

16.7.4.2.5 Reload

メソッド: Reload ()をboolean とて

รัสวง พิร.ID: 1002

詳細:

Reload ファイルシステムからドキュメノトコンテンツを更新する。

16.7.4.2.6 Save

メソッド: Save ()をboolean とて

รีสุวง พิร ID : 1003

詳細: Save は現在のギュメトをロケーション Path に保存する。

16.7.4.2.7 SaveAs

メンッド: SaveAs (strFileName をstring 出の をboolean 出て

รีสุวง พิร ID : 1004

詳細:

SaveAs はPath をstrFileName に設定し、ドキュメトをこの場所に保存します。

16.7.4.3 Events

MapForceControlDocumentActiveX コトロールは次の接続ポイントイベントを提供します:

OnDocumentOpened OnDocumentClosed OnModifiedFlagChanged OnContextChanged OnFileChangedAlert OnActivate OnSetEditorTitle

16.7.4.3.1 OnActivate

イベント: OnActivate ()

รัาวง พิร ID : 1005

詳細:

ドキュメトコトロールが有効化されると、フォーカスのある、ユーザー入力として使用することのできるこのイベトがリガーされます。

16.7.4.3.2 OnContextChanged

イベナ: OnContextChanged (i strContextName をString 出て、i bActive をbool 出つをbool 出て

รีสวง พิร ID : 1004

詳細: なし

16.7.4.3.3 OnDocumentClosed

イベナ: OnDocumentClosed (objDocument をDocument 出つ

รีสุวง พิร ID : 1001

詳細:

このイベトーはキュメトかこのエトロールコードされると閉じられます。引数 objDocument は、MapForce 自動化インターフェイスからの Document オブンエクトです、で慎重に使用してくたさい。

16.7.4.3.4 OnDocumentOpened

イベナ: OnDocumentOpened (objDocumentをDocument とつ

รีสุวง พิร ID : 1000

詳細:

このイベトは、このエトロール内でドキュメトが開かれるよりガーされます。引数 objDocument は、MapForce 自動化イクター フェイスからのDocument オブジェクトです。ドキュメントの詳細をクエリオるために使用することができ、おまよ、追加オペレーションを行います。

16.7.4.3.5 OnDocumentSaveAs

イベナ: OnContextDocumentSaveAs (i strFileNameをString とつ

รัสวง พิร.ID : 1007

詳細: このドキュメントが新しい名前で内部で保存される場合このイベントがトリガーされます。

16.7.4.3.6 OnFileChangedAlert

イベント: OnFileChangedAlert ()をbool とて

รัาวง พิร ID : 1003

詳細:

このドキュメトトコードされると、このイベトトはトリガーされます。コトロールはハードディスク上で他のアプリケーションで変更されます。イベントを処理すると、クライアントは、trueを返します。おけよ、MapForceが通常の方法で処理すると、falseを返します。例えば、ユーザー・イニ再ロードを促すなど。

16.7.4.3.7 OnModifiedFlagChanged

イベナ: OnModifiedFlagChanged (i bIsModified をboolean とつ

*รัสวง พิ***ร***ID* : 1002

詳細:

このイベトーはドキュメトの状態を、変更された、および、変更されていない状態に切り替えると、トリガーされます。ドキュメトコンテンパ 元のコンテンル異なる場合、パラメーターi_blsModifed is true それ以外の場合は、false を返します。

16.7.4.3.8 OnSetEditorTitle

イベント: OnSetEditorTitle ()

รัาวง พิร ID : 1006

詳細:

このイベトは含まれているドキュメトが内部で名前を与えられると挙げられます。

16.7.5 MapForceControlPlaceHolder

すべてのプロジェクトプレースホルダーウィンドウニ使用することのできるプロパティ: PlaceholderWindowID

プロジェクトプレースホルダーウィンドウのためのプロパティ: Project

プロジェクトプレースホルダーウィンドウのためのメソッド: OpenProject CloseProject

MapForceControlPlaceHolder コトロールは、概要、ライブラリおはプロジェクトウィンドウンどの追加のMapForceウィンドウを表示するためご使用されます。他のActiveX コトロールと同様に使用することができ、クライア・トアプリケーション内に配置することができます。

CLSID: FDEC3B04-05F2-427d-988C-F03A85DE53C2 ProgID: Altova.MapForceControlPlaceHolder

16.7.5.1 プロパティ

以下のプロ、ティカ定義されています

PlaceholderWindowID

MapForceAPI へのアクセス: Project

16.7.5.1.1 Label

プロパティ: Label をString (読み取り専用)とて

รีสุวง พิร ID : 1001

詳細: このプロ・ティオプレース市ルダのタイトルへのアクセスを与えます。 プロ・ティオ読み取り専用です

16.7.5.1.2 PlaceholderWindowID

プロディ: PlaceholderWindowiDを<u>MapForceControlPlaceholderWindow</u> 出て

ディス/ シチ*ID* : 1

詳細:

このプロ、ティを使用すると、コトロールのプライア・トコリア内で表示される MapForce ウイドウが通知されます。 PlaceholderWindowID は、<u>MapForceControlPlaceholderWindow</u>列挙の有効な値に設定することができます。コート ロールはすくに状態を変更し、すくに新規のMapForce ウイドーグを表示します。

16.7.5.1.3 Project

プロパティ: Project Project とて(読み取り専用)

*ティスペ*୬チ*ID*:2

詳細:

Project プロ ティは MapForce 自動サイ ー API の Project オジジェクト へのアクセスを与えます。このインターフェイスはコ ントロールヘロードされるプロジェクト に使用される追加機能を与えます。 プレースオルダウイド ウが MapForceXProject Window (=3)の値を持つPlaceholderWindowID の場合のみ、プロ ティは有効なプロジェクト インターフェイスを返します。プロ ティイは読 み取り専用です

16.7.5.2 メソッド

以下のメソバカ定義されています

OpenProject CloseProject

16.7.5.2.1 OpenProject

メンッド: OpenProject (strFileName をstring との をboolean とて

ディスト やチ ID : 3

詳細:

OpenProject はstrFileName を新規のプロジェクトと、てコトロールコードします。 プレースオルダウイドウが XMLSpyXProjectWindow (=3)と異なる <u>PlaceholderWindowID</u>を持つ場合、メノドは失敗します。

16.7.5.2.2 CloseProject

メソッド: CloseProject ()

ティス พิチID:4

詳細:
CloseProject はプロジェクト レロードされたエントロールを閉じます。プレース市レダウイドウが XMLSpy XProject Window (=3)と異なる Placeholder Window ID を持つ場合、メクバは失敗します。

16.7.5.3 イベント

MapForceControlPlaceholder ActiveX コトロールは次の接続ポイトイベトを提供します:

OnModifiedFlagChanged

16.7.5.3.1 OnModifiedFlagChanged

イベナ: OnModifiedFlagChanged (i bIsModified をboolean とつ

*ディスト*シチID:1

詳細:

このイベトはMapFor ceXPr oj ect W ndow (=3)の <u>PlaceholderWindowID</u>を持つルースヤルダエトロールのため のチョン・リガーされます。プロジェクトのエレテンパ変更された、おけよ変更されていない状態の間で変更があった場合、イベトは実行され ます。プロジェクトコンテンパ元のエレテンと異なる場合、パラメーターi_blsModifed はtrueを返し、それ以外の場合は、falseを 返します。

16.7.5.3.2 OnSetLabel

イベント: OnSetLabel (i strNewLabel をstring とて)

*รัสวง พิร.*ID: 1000

詳細:

プレースカリダウイドウのタイトリカ変更されると、挙したれます。

16.7.6 列挙

以下の列挙が定義されています

ICActiveXIntegrationLevel MapForceControlPlaceholderWindow

16.7.6.1 ICActiveXIntegrationLevel

MapForceControlのIntegrationLevelプロゲイのための可能な値。

ICActiveXIntegrationOnApplicationLevel = 0
ICActiveXIntegrationOnDocumentLevel = 1

16.7.6.2 MapForceControlPlaceholderWindow

この列挙にはサポトされる追加 MapForce ウイドウのノストが含まれています。

MapForceXNoWindow	=	
MapForceXLibraryWindow	=	0
MapForceXOverviewWindow	=	1
MapForceXValidationWindow	=	2
MapForceXProjectWindow	=	3
MapForceXDebuggerValuesWindow	=	4
MapForceXDebuggerContextWindow	=	5
MapForceXDebuggerPointsWindow	=	6

17 付録

付録

以下の対録コはMapForce に関する技術的な情報や、ライセンスに関する重要な情報が収められています。各付録コお以下のようにサブセクションが収められています:

<u>技術的なデータ</u>

- OS ならび こともの 要件
- Altova XML パーサー
- Altova XSLT とXQuery エンジン
- Unicode のりドト
- ライセンス使用状況測定

ライセンス情報

- 電子的なソフトウェアの配布
- 著作権
- 使用許諾契約書

17.1 エンジン情報

このセクションゴは Altova XML バリデーター、 Altova XSLT 1.0 エンジン、 Altova XSLT 2.0 エンジン、 そして Altova XQuery エンジン つ実装に 特化した情報が 含まれます。

17.1.1 XSLT および XQuery エンジンに関する情報

MapForceのXSLTおよびXQueryエンジノは、W3C仕様に従っています、ですから、XMLSpyの以前のデンシン内のAltova エンジンより人厳密です。この結果、以前のエンジンで無視されていたシンゴラーが、MapForceによりエラーとして挙げられます。

例えば

- パス演算子の結果がシードと非ーノードを両方含む場合、型エラー(err:XPTY0018)です。
- パス式 E1/E2 内のE1 がードのシーケンスを評価したい場合、型エラー(err:XPTY0019)です。

この種類のエラーが発生した場合、XSLT/XQueryドキュメイオゴコンクスタンスドキュメイを必要につたじて修正してくたさい。

このセクションは、エンジンの実装固有の機能を仕様別に整理して説明します。

- <u>XSLT 1.0</u>
- <u>XSLT 2.0</u>
- <u>XQuery 1.0</u>

17.1.1.1 XSLT 1.0

MapForce のXSLT 1.0 エンジンは W orld W ide W eb Consortium(ワールド・ウボ・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。

実装になっての注意点

xsl:output Omethod 属性がHTMLに設定された場合、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 ドキュメントに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

17.1.1.2 XSLT 2.0

<u>ב הצלישי</u>

- エジン 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

MapForce のXSLT 2.0 エンジノよ W orld W ide W eb Consortium (ワールド・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"

```
</xsi: 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 エンジンが、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こ扱うかの説明です。

xsl:result-document 追加してサポトされるエンコードは以下の通りです(Altova-固有): x-base16tobinary および x-base64tobinary.

function-available インスコープ関数の使用をテストする関数(XSLT、XPath、および拡張関数)。

unparsed-text

href 属性は 以下を受け入れます(i) ベースuri フォルダー内のファイルの相対パス および(ii) 相対パンを持つませまけたい file://プロトコール 追加してサポートされる エンコードは以下の通りです(Altova-固有): x-binarytobase16 および 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 サンドのエンコード値は使用しな でくたさい base16tobinary, base64tobinary, binarytobase16 and binarytobase64.

17.1.1.3 XQuery 1.0

- エバン 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・モジュール
- 外部関数
- 照合順序
- 数値データの精度
- <u>XQuery 命令ザポト</u>

適合性

MapForce のXQuery 1.0 エンジノよ W orld W ide W eb Consortium (ワールド・ウェブ・コンノーシアム) (W 3C) の2010 <u>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 かどのようについの機能を実装するかについて説明する したが下に挙じたれています。

スキーマ認識

XQuery 1.0 エノ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:	ttp://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 エンジノは 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、ウンドとて認識します。
- Since the 上にフストされた内蔵の関数名前空間は、XQuery内のデフォルの関数です。内蔵の関数が呼び出される際、名前空間、fn: プレフィックスを使用する必要はありません。(例えば、string("Hello")がfn:string 関数を呼び出す場合。)しかし、プレフィックスfn:はクロレプローグ内で名前空間を宣言することがく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 アフォルの 関数 名前空間をすることに しょう
- 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xs: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ブローグ内で、インドすることは、使用することができます。(サンプル xs: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 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。ですから以下といます:xs: year MonthDuration。

関数のけるの名前空間、型エンストラクタ、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 プレファックス、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、 おこ有効な場合、 ポストスキーマ検証 情報が XML データのためこ生成され、 クロド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)、呼び出された男数と変数が定義済みであるか、など構文などのクエーリのアスペクトをチェックします。静的分析フェーズでエラーが検知されると、実行は停止されます。

クロカ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れ場合、エラーが報告されます。例えば、式xs:string("1") + 1は、エラーを返します。型xs:stringのオペライを足し算のオペレーションが実行できないサングです。

ライブラノモジュール

ライブラノモジューールは、再利用のナダ関数と変数を保管します。XQuery 1.0 エンジノは単一の外部 XQuery ファイルに保管されて いるモジュールをサポートします。このようなモジュールファイルはプロローグにターゲ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" };

すべての関数および変数は、モジュールに関連した名前空間に属するモジュール内で宣言されています。モジュールはクエレプロローグ内の import module ステートメートを使用して X Query ファイルにインポートする際に使用されます。 import module ステートメート は、ライブラノモジュー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。 照合順序オブションの指定されていない場合、デオパトのUnicode-コードポイト、照合順序のが使用されます。

数値データの精度

- xs:integer データ型に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 データ型はおか数点の後に20桁の制限があります。
- xs:float とxs:double データ型には15 桁の精度の制限があります。

XQuery 命令サポート

Pragma 命令は、サポートされていません。発生した場合、無視されフォールックの式が検証されます。

17.1.2 XSLT と XPath/XQuery 関数

このセクションでは、XPath およびませま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XQuery エンジンが、XPath/XQuery 関数仕様で指定されているhttp://www.w3.org/2005/xpathfunctionsプレフィックス無しの関数をデフォルト関数の名前空間に属するものとして読み込むためです。
- 関数において、各アイテムか引数となるようなシーケンスか期待されており、2つ以上のアイテムがシーケンスによい呼び出された場合、エラーか返されます。
- 全ての比較はUnicodeコードポイトコレグシ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桁か球められます。xs:decimal型に結果が収められる除算の場合、端数処理を行うことなく精度は小数点以下の19桁になります。

<u> 黙示的なタイムゾーン</u>

2つのdate、time、おけま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かあります。値の中にタイムゾーンが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。

<u>照合順序</u>

デフォルの照合順序は、Unicode コードポイントをベースに文字列を比較するUnicode コードポイント照合順序です。エンジンは Unicode 照合アルゴリズムを使用しています。他のサポートされる照合順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リストのURIを提供してください(アのテーブル)。max とmin 関数を含む、文字列の比較は、指定された照合順 序に沿って行われます。照合順序オプションが指定されていない場合、デフォルトのUnicode コードポイン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: フフレス語	<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//)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: 才之久語	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ι語	pt_AO, pt_BR, pt_GW, pt_MZ, pt_PT, pt_ST
ru: 印ンア 語	ru_MD, ru_RU, ru_UA
sv: 자ナーデン語	sv_FI, sv_SE

名前空間軸

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 メカニズムにより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 ださい。

17.1.2.1 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 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、デジョンが拡張機能への継 続的サポート、ますよ週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。

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)のコ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、xsut 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スト されています。Altova 拡張関数などの、他のライブ ラからの関数はプレフィックスと共にリスト 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

XSLT 関数

XSLT 関数はXSLT 2.0 のcurrent-group() やkey() 関数と同様に XSLT コンテキストにて使用することができます。(例え ば、XQuery コンテキストなどの) 非-XSLT コンテキストでは使用することができません。 XBRL に対する XSLT 関数は、 XBRL をサポ ートするエディンコンの 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
- 位置情報
- <u>イメージ 関連した</u>
- <u>数値</u>

17.1.2.1.1 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、このセグションでは使用されます。製品の今後の、ージョンの拡張機能への 継続的サポト、または個別の関数の振る舞いは変更する可能性があることに注意してくたさい。Altova 拡張機能へのサポトに関し ては、今後のリースのドキュメトを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

標準関数

distinct-nodes [altova:]

altova:distinct-nodes(node()*) をnode()* とする XSLT1 XSLT2 XSLT3 入力とて1つ以上のノードを必要と、同じセナから重複した値を持つノードを除しケノードを返します。XPath/XQuery 関数 fn:deep-equal を使用して比較を行うことができます。 🖃 サンプル

- altova:altova:distinct-nodes (country) は重複し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パレます。評価された式の出力を返します。例えば altova:evaluate('//Name[1]') はドキュメイ内の最初のName 要素のエレテン
を返します。式 //Name[1] は 一重引用符を使用することには、文字列としていなわます。

altova:evaluate 関数は、オブションと、て追加の引数を持つことができます。これらの引数は、p1, p2, p3... pN の名前を持 つスコープ内の変数の値です。使用に関して以下の点に注意してくたさい(i)変数は、 X か整数である箇所のフォーム DX の名前と 共に定義される必要かあります。(ii) altova:evaluate 関数の引数は (上の署名参照) 2 番目の引数からよ 数値順の 変数のシーケンスに対応した引数のシーケンス変数の値を与えます: p1 to pN: 第2 引数は変数 p1 の値で、第3 引数は、変数 p2 の値です。(iii) 変数の値は型 item* である必要がみます。

🖃 サンプル

<xsl:variable name="xpath" select="'\$p3, \$p2, \$p1'" />

```
<xsl:value-of select="altova:evaluate($xpath, 10, 20, 'hi')" />
outputs "hi 20 10"
```

上のストに関して、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、ユーザーが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)" or der="ascending"/>。altova:evaluate()

関数は、エ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 al tova: evaluate() 関数にい返され、select 属性 <xsl: sort select="**Price**" order="ascending"/> の値になります。この sort 命令が、Order とう要素のエ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。これは、以下の式な どでは達成することはできません: <xsl: sort select=".../UserReq/@sortkey" order="ascending"/>。上の場 合、並べ替え条件は sortkey 属性自身であり、Price まけは Date (まけは、現在の sortkey のエテング)ではありません。

- メモ 静的なコンテキストは、変数以外以外で、呼び出し環境の名前空間、型、機能、を含みます。ベースURIとデフォルトの 名前空間は継承されます。
- □ <u>追加サンプル</u>
 - 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 / べかして定義されている変数/はありません。

encode-for-rtf [altova:]

```
altova:encode-for-rtf(input as xs:string, preserveallwhitespace as xs:boolean,
preservenewlines as xs:boolean) をxs:string とする XSLT2 XSLT3
RTF のよめのロードに入力文字列を変換します。空白と新しい行は、それぞれの引数には指定されるboolean の値に基づき保管
されます。
```

[トップ]

XBRL 関数

Altova XBRL 関数はXBRL をサポートする 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 フナ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>]

17.1.2.1.2 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タを処理するための追加機能を提供します。このセクションの関数は、Altova のXPath 3.0 およびXQuery 3.0 エンジン で使用することができます。これらの関数は、XPath/XQuery コンテキストで使用することができます。

関数の名前指定と言語の適用性に関するメモ

Altova 拡張関数はXPath/XQuery 式で使用するこかでき、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て返す
- 日付から週数を整数とて返す
- 各型の構文コンポーネトから日付、時刻、期間の型を構築する
- 文字列入力から日付、日付時刻ませま時刻を構築する
- 年齢に関連した関数

▼ アルファベナ順にグループ化

altova:add-davs-to-date altova:add-days-to-dateTime altova:add-hours-to-dateTime altova:add-hours-to-time altova:add-minutes-to-dateTime altova:add-minutes-to-time altova:add-months-to-date altova:add-months-to-dateTime altova:add-seconds-to-dateTime altova:add-seconds-to-time altova:add-years-to-date altova:add-years-to-dateTime altova:age altova:age-details altova:build-date altova:build-duration altova:build-time altova:current-dateTime-no-TZ altova:current-date-no-TZ altova:current-time-no-TZ altova:date-no-TZ altova:dateTime-no-TZ altova:days-in-month altova:hours-from-dateTimeDuration-accumulated altova:minutes-from-dateTimeDuration-accumulated altova:seconds-from-dateTimeDuration-accumulated altova:format-duration altova:parse-date altova:parse-dateTime altova:parse-duration altova:parse-time altova:time-no-TZ altova:weekday-from-date altova:weekday-from-dateTime altova:weeknumber-from-date altova:weeknumber-from-dateTime

[<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 「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 <mark>を</mark> 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

日付までの期間を牛致で表示します。第20万日銀は第10万日銀色で与えられたxs:date に追加される牛致です。結果はxs:date 型です。

□ <u>サンプル</u>

- 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 (DateTime as xs:dateTime, Months を xs:integer)
asxs:dateTime とする XP3.1 XQ3.1
xs:dateTime に月数での期間を追加します(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 に追加される月数です。結果はxs:dateTime 型です。

= サンプル
```

- altova:add-months-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), 10) 2014-11-15T14:00:00 を返します。
- altova:add-months-to-dateTime(xs:dateTime("2014-01-15T14:00:00"), -2) 2013-11-15T14:00:00 を返します。
- add-days-to-dateTime [altova:]

altova:add-days-to-dateTime(DateTime as xs:dateTime, Days as xs:integer) をxs:dateTime どうる 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(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 に追加される時間数です。結果はxs:dateTime 型です。

= サンプル
```

- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), 10) は2014-01-15T23:00:00 を返します。
- altova:add-hours-to-dateTime(xs:dateTime("2014-01-15T13:00:00"), -8) は2014-01-15T05:00:00 を返します。
- add-minutes-to-dateTime [altova:]

altova:add-minutes-to-dateTime (DateTime as xs:dateTime, Minutes as xs:integer)を xs:dateTime とする XP3.1 XQ3.1 xs:dateTime に分数での期間を追加します(*下のサンプル参照*)。第2の引数は、第1の引数とて与えられた xs:dateTime に追加される分数です。結果はxs:dateTime 型です。 = <u>サンプル</u>

- 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(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 に追加される秒数です。結果はxs:dateTime 型です。 = サンプル

- altova:add-seconds-to-dateTime(xs:dateTime("2014-01-15T14:00:10"), 20) 2014-01-15T14:00:30 を返します。
- 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 (期間を追加する xp3.1 xq3.1

これらの関数はxs:date に期間を追加し、xs:date を返します。xs:date 型はCCYY-MM-DD フォーマナです。

add-years-to-date [altova:]

- 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。

🖃 サンプル

- altova:add-months-to-date(xs:date("2014-01-15"), 10) 2014-11-15 を返します。
- altova:add-months-to-date(xs:date("2014-01-15"), -2) 2013-11-15 を返します。
- add-days-to-date [altova:]

altova:add-days-to-date (Date as xs:date, Days as xs:integer) をxs:date とする XP3.1 XQ3.1 日付まで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

第1の引数とて提出された期間を、第2の引数として提出された文字列によをフォーマナします。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。

🗉 <u>サンプル</u>

- altova:format-duration(xs:duration("P2DT2H53M11.7S"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]")は"Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7" を返します。
- altova:format-duration(xs:duration("P3M2DT2H53M11.7S"), "Months:[M01] Days: [D01] Hours:[H01] Minutes:[m01]") は"Months:03 Days:02 Hours:02 Minutes:53" を返 します。
- parse-duration [altova:]

altova:parse-duration(InputString as xs:string, Picture as xs:string) をxs:duration と する XP3.1 XQ3.1

```
・ ゆ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。
ー ・ ・・ ーー・・・
```

🗉 <u>サンプル</u>

 altova:parse-duration("Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7"), "Days:[D01] Hours:[H01] Minutes:[m01] Seconds:[s01] Fractions:[f0]") は

```
"P2DT2H53M11.7s" を返します。
```

```
• altova:parse-duration("Months:03 Days:02 Hours:02 Minutes:53 Seconds:11 Fractions:7", "Months:[M01] Days:[D01] Hours:[H01] Minutes:[m01]") は "P3M2DT2H53M" を返します。
```

[<u>トップ</u>]

xs:time (期間を追加する xp3.1 xq3.1

これらの関数はxs:timeに期間を追加し、xs:timeを返します。xs:time型はhh:mm ss.sss構文フォームです。文字 Z は協定世界時(UTC)を表します。他のタイムゾーンはUTCとの差異を+hh:mm おけよ-hh:mmのフォーマナで表示しています。タイ ムゾーンの値が無い場合は、UTCではない未知のタイムゾーンとして見なされます。

add-hours-to-time [altova:]

```
altova:add-hours-to-time(Time as xs:time, Hours as xs:integer) をxs:time とする XP3.1
XQ3.1
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 どる
XP3.1 XQ3.1
```

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。

🖃 <u>サンプル</u>

- 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。

□ <u>サンプル</u>

- 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。

[<u>トップ</u>]

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、おけはxs:time 値からそれぞれタイムゾーンを削除します。xs:dateTimeと xs:dateTimeStamp の差異は、後者のタイムゾーンが必要な場合です。(前者の場合は任意です。)xs:dateTimeStamp 値の フォーマトはCCYY-MM-DDThh:mm:ss.sss±hh:mm おけはCCYY-MM-DDThh:mm:ss.sssZ です。、日付と時刻が xs:dateTimeStamp とてシステムシアン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|お|数を必要としません。current-date()(システムカロックによる現在の時刻)のタイムジーンの部分を削除し、
xs:date の値を返します。

国 <u>サンプル</u>
```

現在のdate が2014-01-15+01:00 の場合:

- altova:current-date-no-TZ() は2014-01-15 を返します。
- current-dateTime-no-TZ [altova:]

altova:current-dateTime-no-TZ() をxs:dateTime とする XP3.1 XQ3.1 この関数は引数を必要とません。current-dateTime()(システムクロックによる現在の時刻)のタイムゾーンの部分を削除し、 xs:dateTimeの値を返します。

*⊒ <u></u>דירוו*ע

現在のdateTime が2014-01-15T14:00:00+01:00 の場合:

- altova:current-dateTime-no-TZ() は2014-01-15T14:00:00 を返します。
- current-time-no-TZ [altova:]

```
altova:current-time-no-TZ()をxs:timeとする XP3.1 XQ3.1
この関数はお|数を必要とません。current-time()(システムケロックによる現在の時刻)のタイムゾーンの部分を削除し、
xs:timeの値を返します。
```

🗉 <u>サンプル</u>

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
注意してくたさい。

コ 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(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、秒数をそれそれ返します。

days-in-month [altova:]

```
altova:days-in-month(Year as xs:integer, Month as xs:integer) asxs:integer XP3.1 XQ3.1
は指定された月内の日数を返します。Year とMonth 引数を使用して月を指定することができます。

サンプル
```

- altova: days-in-month (2018, 10) は 31 を返します。
- altova: days-in-month (2018, 2) は28 を返します。
- altova:days-in-month(2020, 2) は29 を返します。
- hours-from-dayTimeDuration-accumulated

altova:hours-from-dayTimeDuration-accumulated(DayAndTime as xs:duration) asxs:integer XP3.1 XO3.1

DayAndTime 引数(which is of type xs:duration)にお提供される期間内の時間の総数を返します。Day とTime コン ポーネント内の時間は、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。オガデ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

DayAndTime 引数(which is of type xs:duration) にと提出される期間内の分数の総数を返します Day とTime コン ポーネトは、追加され、整数である結果に追加されます。オガティブょマイナスの値で期間の結果をナビゲートします。

- 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 か曜日を整数とつを返します。曜日は(米国式フォーマナを使用して)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 のフォーマナ が必要な場合、この関数の他の署名を使用します(アの次の署名を参照 。

- 🗉 <u>サンプル</u>
 - 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して返します。曜日は月曜-1 から開始して番号を付けます。第2 の引数(整数)が0 の場合、日曜-1 から開始して、曜日は1から7と番号を付けます。第2の引数が0 以外の整数の場合、 月曜-1です。第2の引数がない場合、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 (Date as xs:date) を xs:integer とする XP3.1 XQ3.1 日付を単一の引数とて、この日付の曜日を正数とて返します。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 のフォーマトが必要な場合、この関数の他の署名を使用します(*下の次の署名を参照*)。 = <u>サンプル</u>

• 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して返します。曜日は月曜=1 から開始して番号を付けます。第2(フォーマナ) 引数が0,の場合、日曜=1 から開始し、曜日は1 から7で番号付けられます。第2 引数が整数で0 以外の場合、月曜=1です。第 2 の引数がない場合、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)。

- 🖃 サンプル
 - altova:weekday-from-date(xs:date("2014-02-03"), 1) は月曜を示す1 を返します。
 - altova:weekday-from-date(xs:date("2014-02-03"), 4) は月曜を示す1 を返します。
 - altova:weekday-from-date(xs:date("2014-02-03"), 0) は月曜を示す2 を返します。

[<u>トップ</u>]

xs:dateTime おは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りです

- 0 = US 米国のカレンダー(週の始まりは日曜日)
- 1 = I SO 標準、欧州のカレンダー(週の始まりは月曜日)
- 2 = イスラムのカレンダー(週の始まりは土曜日)

デンオルトは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、地格ー週間先です。

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)は続くカレンダーンステムを指定します。 サポ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)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.

∃ <u>サンプル</u>

- 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は欧州カレンダーのこの日付よりに一週間先です。

[<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 おけまxs:duration の構文コンポーネントを入力引数と、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 以上であってしないません。 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 XO3.1

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。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)。

```
□ <u>サンプル</u>
```

• 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 どる 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型の値を構築するためコは2つの引数か必要です。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(月数かと同じおけより大きくなると、整数は12により割されます。 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型を作成するコよ次の署名を参照してくたさい。

```
□ <u>サンプル</u>
```

- 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コよ4 つの引数が必要です。最初の引数は期間の日数値の部分で、第2、 第3、第4引数は、それそれ期間の時間数、分数、秒数値のです。これらの3 つの時間引数は、次の高い単位の値に加算され、結果は期間の全体の値を計算するよめご使用されます。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めご使用されます。xs:yearMonthDuration型の期間を構築するよめにはたの署名を参照してくたさい。

□ <u></u>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 まけよ時刻を構築する 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。タ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任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 年

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 (DateTime as xs:string, DateTimePattern as xs:string) を xs:dateTime とする XP2 XQ1 XP3.1 XQ3.1 DateTime 入力文字列をxs:dateTime の値とて返します。第二の引数であるDateTimePattern は、入力文字列のペ ターン(コ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ポーネト指定子および狂意の文 字であるコポーネト セシータと共に説明されています。下のサンプルを参照してくたさい。 D 日 M 月 Y 年

н	時間
m	分
s	秒

DateTimePatternの やーンは DateTime の や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 構文フォーマトにないます。

🖃 <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 ターン(エポーネン
トのシーケンス)を指定します。TimePattern は 下にリストされるエポーネント指定子および狂意の文字であるエポーネント セ
パータと共に説明されて、ます。 下のサンプルを参照してください。
H 時間
m 分
s 秒
```

TimePatternのターンはTimeのターンに一致する必要が対はす。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、あるオブジェクトの年齢の年数で
ある正数を返します。入力引数が1 年よりに大き、場合、または一年の場合、返される値は負の数です。
```

If the current date is 2014-01-15:

- altova:age(xs:date("2013-01-15")) は1 を返します。
- altova:age(xs:date("2013-01-16")) は0 を返します。
- altova:age(xs:date("2015-01-15")) は-1 を返します。
- altova:age(xs:date("2015-01-14")) は0 を返します。

altova:age (StartDate as xs:date, EndDate as xs:date) をxs:integer とする XP3.1 XQ3.1 最初の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、あるオブジ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の引数より上後の日付の場合、返される値は負の数です。

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 り、用しておけば遅い値をもつことができますが、入力の日付が早いか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)* どうる XP3.1 XQ3.1

二つの別数間の年数、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。最初の別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う遅いかを示しません。 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>]

17.1.2.1.3 XPath/XQuery 関数: 位置情報

以下の位置情報 XPath/XQuery 拡張関数は、MapForce の現在の デュ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、 おけよ(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: 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

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して出力します。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、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」を参照してくたさい。緯度の値の範囲は +90 から-90(北から南)です。経度の値の範囲は+180 から-180(東から西)です。

メモ <u>image-exif-data</u> 関数とExif メタデータの<u>@Geolocation</u> 属性を位置情報入力文字列を提供する際に使用することができます。

- *□ <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" を返します。
- □ <u>位置情報出力文字列フォーマホ</u>:

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。

1

<mark>度、分、10 進の秒、方角のサフィックス付き(N/S, E/W)</mark> 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

2

3

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

<mark>度、分、10 進の秒、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。</mark> +/-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)のオックのプラスサインは狂意です。 +/-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

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ット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 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)の一ケンズ返します。
- 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。

付録

- 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。このセクションのサンプルでは、入力文字列を区別するオーメコを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 <u>サンプル</u>: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィックスサイン付き(+/-); (N/E) のオュタ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)のオック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

国 <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(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	E	33°51'21.91" <mark>S</mark> 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ット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かがっ使用されていると、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、不一致をもたっします。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ために使用されている引用符 は黄色い(")でいくうくいされており、エスケープした単位インジケーターは青い("")でいくうくいされています。

- 度、分、10進の秒、方角のサフィックス付き(N/S, E/W)
 D°M'S.SS"N/S D°M'S.SS"W/E
 サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10 進の秒、プレフィ・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)のオック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

<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-distance-mi [altova:]

altova:geolocation-distance-mi(GeolocationInputString-1 as xs:string, GeolocationInputString-2 as xs:string) をxs:decimal とする XP3.1 XQ3.1 2 つの位置情報の間の距離をマイルで計算します。位置情報入力文字列を提供することのできるフォーマナは下にリストされていま す。緯度の値の範囲は+90 から-90 (北から南)です。経度の値の範囲は+180 から-180 (東から西)です。

メモ <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 とかできます。

- *□ <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オ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オで、経度が他のフォーマオ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たみ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。

- 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(+/-); (𝑥/𝖻)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"

Altova MapForce 2021 Professional Edition

- 度、分、10 進の分、方角のサフィクス付き(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) のナック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(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:]

altova: geol ocati ons-bounding-rectangle(Geol ocati ons を xs: sequence として Geol ocati onOutput StringFormat 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におい下にコストされています、「位置情報入力文字列書式」を参照してくたさい。 緯度の値は+90 から-90 への(N から S への)の範囲です。 経度の値は+180 から-180 への(E から W への)の範囲です。

関数の2番目の引数は出力シーケンス内の2つの位置情報文字列の書式を指定します。引数はそれぞれの値が異なる位置情報文字列書式を識別する1から4の整数の値を取ります(下記の「位置情報出力文字列書式」を参照してください)。

メモ image-exif-data 関数とExif メタデータの属性は入力文字列を提供するために使用することができます。

<u>サンプル</u>

• altova:geolocations-bounding-rectangle(("48.2143531 16.3707266", "51.50939 -

0.11832"), 1) ばーケンス ("51°30'33.804"N 0°7'5.952"W<mark>"</mark>, "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) はシーケンス("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 ブルコー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(+/-); (𝑥/𝖻) 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
- 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>位置情報出力文字列フォーマナ:</u>

与えられた緯度と経度は下にリストされる出力フォーマトによりフォーマトされます。希望するフォーマトは整数 ID(1 から 4)によし識別されます。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(N/S, E/W)</mark> 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

2

1

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

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" サンプル: 33°55'11.11" -22°44'66.66"

4

10進の度、プレフィックスサイン付き(+/-); (N/E)のオッタのプラスサインは任意です。 +/-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

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) か多角形のエレ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、(アバンストされるフォーマーケの)位置情報入力文字列により提供されます。Geolocation 引数が 多角形エレア内にある場合、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>ー 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ム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、不一致をもちょうします。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 ブルコー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コで使用されている引用符 は黄色い(") でいくうくいされており、エスケープした単位インジケーターは言い("") でいくうくいされています。

- 度、分、10 進の秒、方角のサフィックス付き(N/S, E/W)
 D°M'S.SS"N/S D°M'S.SS"W/E
 サンプル: 33°55'11.11"N 22°44'55.25"W
- 度、分、10進の秒、プレフ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)のオック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

■ Altova Exif 属性: 位置情報

Altova XPath/XQuery エンジンはカスタム属性 Geolocationを標準 Exif メタデータ タグから生成します。 Geolocation は、4 つのExif タグの連結です:単位の追加された(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

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)には説明されている 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は提供されます。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(東からの)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を用されていると、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) のオメタ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>אלילילי</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) のオック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>]

17.1.2.1.4 XPath/XQuery 関数: イメージに関連

以下のイメージ 環連した XPath/XQuery 拡張関数は、MapForce の現在の レジョン 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

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因したことかです。この情報が使用できない場合、空の文字列が返されます。この関数は、Base64 イメージをファイルとして保存し、適切なファイル拡張子を動的取得する際に役に立ちます。

🗉 サンプル

- 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、およびカイリティ。

サイズ引数には3 つのサイズ変更のオプションかあります。

(X,Y	絶対ピクセルの値。アスペクト率は保持されません。高さと幅は自動的にイメージの長いおよい短い辺に逢わされるため、高さと幅の指示は関係ありません。2 つの整数アイテムのシーケンスとして値が入力されます。かって が必要です。
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に回転させます。負の値はイメージを反時計回りに回転させます。 Altova Exif 属性 OrientationDegree を使用して、イメージの現在の回転の度数(0, 90, 180, 270) をイメージのExif Orientation タグから取得することができます。ですが、OrientationDegree 属性はデータのOrientation タグから取得されるため、Exif データ内にOrientation タグか存在する場合のみ使用することができます。(下の OrientationDegree 説明を参照してくたさい)。
- クオレディは、0から100の値で、JPEG 圧縮のIJG クオレディスケールの値を参照していますが、クオレディのペーセンテージのインジケーターではありません。サイズとウオレディのとちらかを優先すると、もう一方の優先度が下がます。フルカラーノースのイメージでは、75が通常最高値と見なされています。もし、75が満足の、く結果をもたらささい、場合は、値を上げてくたさい。

メモ Exif データが元のイメージェ存在する場合、変換時に消除され、変換されたイメージコはExif データは存在しません。

- *□ <u>サンプル</u>*
 - 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 のクオリティレベリを与えられます。
 - 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ク、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。)。





• 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 メタデーをを含むExif とう名の要素を返します。 Exif メタデー タはExif 要素の属性の値ベ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。 ベンダー特有のタグが Exif データ内に存在する場合、 タグとその値も属性値のペアとして返されま す。 標準 Exif メタデータタグに追加して (*下のJスト参照*) Altova 特有の属性値のペアもまた生成されます。 これらの Altova Exif 属性は以下のとおしです。

🗉 <u>サンプル</u>

- 属性に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
- 全ての属性にアク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ご役に立ちます。

■ <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 エンジノはカスタム属性 OrientationDegree をExif メタデータタグ Orientation から生成します。

OrientationDegree は標準 Exif タグ Orientation を正数の値(1、8、3 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それぞれ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>]

17.1.2.1.5 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する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 エノ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、このセグ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、今後のリリースのドキュメートを参照してくたさい。

XPath 関数 (XSLT 内のXPath 式で使用):	XP1 XP2 XP3.1
XSLT 関数 (XSLT 内のXPath 式で使用):	XSLT1 XSLT2 XSLT3
XQuery 関数 (XQuery 内のXQuery 式で使用):	XQ1 XQ3.1

自動付番関数

generate-auto-number [altova:]

altova:generate-auto-number(ID as xs:string, StartsWith as xs:double, Increment as xs:double, ResetOnChange as xs:string) を xs:integer とする XP1 XP2 XQ1 XP3.1 XQ3.1 関数が呼び出される度に番号を生成します。関数が初めて呼び出される際に生成される最初の番号はStartsWith 引数により 指定されます。その後の関数の呼び出しは新し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リメートします。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リメートされます。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 値にしたとれます。自動付番は<u>altova:reset-auto-</u> number 関数を使用して)たよ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ずつインクルメトします。その後の呼び出しの4番目の"SomeString"引数がである限り、インクル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 と呼ばれるカナターは1に比小されます。ChapterNumber の値もaltova:reset-auto-number("ChapterNumber")ようごたナされます。
- reset-auto-number [altova:]

altova:reset-auto-number(ID as xs:string) XP1 XP2 XQ1 XP3.1 XQ3.1 この関数は ID 引数内で名づたれた自動付番カウンターの番号をルセナします。引数内のカウンター名を作成した関数の引数 によ指定された数値にルセナされます。altova:generate-auto-number = サンプル

- altova:reset-auto-number("ChapterNumber") は <u>altova:generate-auto-number</u> 関数にり 作成されたhapterNumber とら名の自動付番カウターをルナします。
- 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。

[[トップ]]

数値関数

hex-string-to-integer [altova:]

```
altova:hex-string-to-integer(HexString as xs:string) をxs:integer とする XP3.1 XQ3.1
10 進のシステム(10進)内の正数の16 進の同値の文字列引数を必要と、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 正数を引数として必要とし、文字列として自身のベース16の同値を返します。 🖃 サンプル

- - altova: integer-to-hex-string (1) は'1' を返します。
 - altova: integer-to-hex-string (9) は '9' を返します。
 - altova: integer-to-hex-string (10) は 'A' を返します。
 - altova:integer-to-hex-string(11) は'B' を返します。
 - altova: integer-to-hex-string (15) は'F' を返します。
 - altova: integer-to-hex-string (16) は '10' を返します。
 - altova: integer-to-hex-string (32) は'20' を返します。
 - altova: integer-to-hex-string (33) は'21' を返します。
 - altova: integer-to-hex-string (90) は'5A' を返します。

[[トップ]]

数値フォーマト関数

[[トップ]]

17.1.2.1.6 XPath/XQuery 関数: スキーマ

下にリストされる Altova 拡張関数はスキーマの情報を返します。以下は関数の詳細(i)サンプル(ii)スキーマエンポーネントのリストと ー スペクティブのプロ・ティです。 Altova のXPath 3.0 とXQuery 3.0 エンジンと共に使用することができ XPath/XQuery コンテキスト 内で見つけることができます。

<u>スキーマドキュメントからのスキーマ情報</u>

関数 altova:schema には以下の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エンポーネントをロケートします。2-引数関数は自身のQName によ「職別 される特定のエ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には、特定のエン ポーネントのプロンティを選択する必要かあります。プロンティか非動的アイテム(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 をさらに選択してナビゲートすることが可能です。選択されたプロンティか動的なアイテムの場合、アイテムの値が返され、ナビゲートを更に行う ことはできません。

XML ノードカらのスキーマ情報

関数 altova: type はXMLドキュメトのノードを送信しノードの型情報をPSVI から返します。

関数の名前指定と言語の適用性に関するメモ

Altova 拡張関数はXPath/XQuery 式で使用することができ、XPath、XQuery、およびXSLT 関数の標準ライブラノで使用可能 な機能に更なる機能性を与えます。Altova 拡張関数はAltova 拡張関数名前空間、http://www.altova.com/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、まけは個別の関数の振る舞いは変更する可能性があることに注意してくたさい。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

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 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けっかに繰り返すことができます。
- コポーネトか動的な値の場合、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 ディ

```
"type definitions"
"attribute declarations"
"element declarations"
"attribute group definitions"
"model group definitions"
"notation declarations"
```

メモ Xquery 式ではスキーマは明示的にんポートされる必要があます。XPath 式では、スキーマは処理環境にんポートされる必要 かあます。 例えば XSLT にはxslt:import 命令を使用してんポートします。

"identity-constraint definitions"

(schema 以外の他のすべてのエレポーネント型のプロ ディが下にリストされています。

メモ Xquery 式ではスキーマは明示的にインポートされる必要があます。XPath 式では、スキーマは処理環境にインポートされる 必要があます。例えば XSLT にはxslt:import 命令を使用してインポートします。

∃ <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ませる複合型の名前を返します

• import schema "" at "C:\Test\ExpReport.xsd"; altova:schema() ("type definitions")[1]("name") はスキーマ内の単純型おけ複合型の最初の名 前を返します

Assertion

プロパティ名	プロ/ 守/型	プロ/ テー/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/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/ ディ名	プロ/ 守ィ型	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/ ティ名	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/ 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″)	

Complex Type

プロ/ ディ名	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・ディを持つ関数:("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 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/ ティ名	プロ/ 守ィ型	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/ 守ィ型	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/ ディ名	プロ/ 守/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 (″key″ ″unique″ ″keyRef″)	
selector	XPath プロ/ ゔィレコード	
fields	XPath プロ デルコード のシーケンス	
referenced key	(keyRef 0750074): Identity Constraint	対応するキー制約

Model Group

プロ/ ዮ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/ 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/ የጉ/名	プロ/ 守/型	プロパティ値
kind	文字列	"Notation Declaration"
name	文字列	表記のローカル名
target namespace	文字列	表記の名前空間 URI
system identifier	anyURI	
public identifier	文字列	

Particle

プロ/ 守/名	プロ/ 守/型	プロ/ 守/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/ ディ名	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	(ユニオン型のナメカの) Simple Type のシーケンス	

Type Alternative

プロパテ	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/ 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つの別数)

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。

- このプロレディがエレポーネトの場合、このエレポーネトのプロレディのレク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。
- コンポーネントか動的な値の場合、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 式ではスキーマオ胡示的にインポートされる必要があります。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")
 (お番目のparticles エポーネトの kind プロ ディを返します。このparticles エポーネトは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。

Assertion

プロパティ名	プロ/ 守/型	プロ/ 守/値
kind	文字列	"Assertion"

test	XPath プロ・ディレコード	

Attribute Declaration

プロ/ ディ名	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/ テー値
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

プロパティ名	プロ/ 守/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/ 守/名	プロ/ 守/型	プロパティ値
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/ ディ名	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/ 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	プロノティを持つ関数 ("class": "Type Table",	

	"alternatives": sequence of Type Alternative, "default type definition": Simple Type or Complex Type)	
scope	プロ 守を持つ関数 (″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/ 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/ ディ名	プロ/ 守/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 ゲルコード のシーケンス	
referenced key	(keyRef 07-007): Identity Constraint	対応するキー制約

Model Group

プロノティ名	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

プロパティ名	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/値
kind	文字列	"Particle"
min occurs	整数	
max occurs	整数却出文字列("unbounded")	
term	Element Declaration、Element Wildcard、 おゴよ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	(リスト型のみのための) 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/ ティレコード	
type definition	Simple Type おはComplex Type	

■ XPath Property Record

プロノティ名	プロ/ 守/型	プロ/ 守ィ値
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

▼ 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ります。

🖃 <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摂数を返します。
- for \$element in //Email
 let \$type := altova:type(\$element)
 return \$type ("kind")
 はノードの型のエンポーネント(単純型まゴは複合型)を取りてパーネントのkind property の値を返します。

Assertion

プロノティ名	プロ/ 守/型	プロ/ 守/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/ テー値
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 とまできないことに注意してくたさい。	
inheritable	ブール値	

Attribute Group Declaration

プロ/ ዮィ名	プロ/ 守/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 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/ 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/ 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")	
abstract		
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 守疹持つ関数 (″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	プレ 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/ ディは文字列のシ ーケンスです)

□ Identity Constraint

プロ/ 守/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/ ゔィレコード	
fields	XPath プロ የテルコード のシーケンス	
referenced key	(keyRef 0750074): 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/値
kind	文字列	"Model Group"
compositor	文字列 (″sequence″ ″choice″ ″all″)	
particles	Particle のシーケンス	

Model Group Definition

プロパティ名	プロ/ 守/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/ 守ィ名	プロ/ 守/型	プロパティ値
---------	---------	--------

kind	文字列	"Notation Declaration"
name	文字列	表記のローカル名
target namespace	文字列	表記の名前空間 URI
system identifier	anyURI	
public identifier	文字列	

Particle

プロパティ名	プロ/ 守ィ型	プロ/ 守/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

プロパティ名	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	(ユニオン型のナメカの) Simple Type のシーケンス	

Type Alternative

プロパティ名	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

17.1.2.1.7 XPath/XQuery 関数: シーケンス

Altova のシーケンス拡張関数はXPathとXQuery 式で使用することができ、XML スキーマの異なる日付および時刻データ型で保存されているデータを処理する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

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。

• 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、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; AttributeName は正規表現検索文字列である必要かあります;

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 f が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ば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つしか特たない関数と同じ効果を持ちます、前の署名)、ですが、空のシーケンスは第2の月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 ワラッグエラーを返します。
- altova: attributes ("MyAttribute",) 見つからは 第2引数を返します。

elements [altova:]

altova:elements(ElementName as xs:string) 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アッグを含みます。使用することのできるファッグは以下の通りです:

ェ = 正規表現検索を切り替えます; ElementName は正規表現検索文字列である必要があります;

f = このオブションが指定されている場合、ElementName は完全一致を提供します。それ以外の場合は、ElementName は 属性名に部分的に一致するとその属性を返します。例えば f が指定されていない場合、MyElem はMyElement を返します。 i = 大文字と小文字を一致させる検索に切り替えます。

p = 検索に名前空間プレフィックスを含みます。ElementName は、名前空間プレフィックスを含みます。例えば altova:MyElement。

フラッグは順序に関われる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 は第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",) 見つからよ第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 のアリティを持つ)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りに、Sequence の各アイテムが提出されます。(注意: この関数は1 つの引数のみを必要 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(5 to 10, function(\$a) {\$a mod 2 = 0}) はxs:integer 6 を返します。 Condition 引数は \$a とう名のヘライ人関数を宣言し、定義します。XPath 3.0 ヘライン関数 function()を 参照します。altova: Sequence 引数内の各アイテムでは find-first が やされ、代わりに \$a を入力値とま す。入力値は関数定義(\$a mod 2 = 0)内の条件に対してテストされます。この条件を満たす最初の入力値が altova:find-first(この場合は6)の結果として返されます。

• altova:find-first((1 to 10), (function(\$a) {\$a+3=7})) はxs:integer 4 を返します。

<u>更なるサンプル</u>

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。

<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 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こかです。doc-available#1の意味は以下のとおりです。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 関数の結果として返されます。doc-available()関数に関しての注意点:相対パスは、デンオリル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
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、各シーケンスからの1つずコニ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たれまた。組み合わせお以下のよう:指定されています。
```

```
If Seq-01 = X1, X2, X3 ... Xn
And Seq-02 = Y1, Y2, Y3 ... Yn
Then (X1 Y1), (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()を評価しない場合、altova:find-first-combination に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はアイテムではい場合。

🗉 <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 かったを返します。

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わまた。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() と評価するよう:指示する最初のペアはaltova:find-first-pair の結果として返されます。以下の点に注意してくたさい(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ない場合。

- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 32}) は xs:integers (11, 21) のシーケンスを返します。
- altova:find-first-pair(11 to 20, 21 to 30, function(\$a, \$b) {\$a+\$b = 33}) は結果 無しを返します。

上の2 つのサンプルに表示される通り、ペアの順序は以下の通りです: (11, 21) (12, 22) (13, 23)...(20, 30)。(33 を返す指示されたペアがは、すめ) 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 ご規数は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 つまけは 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、下のように指定されています。

If Seq-01 = X1, X2, X3 ... Xn And Seq-02 = Y1, Y2, Y3 ... Yn Then (X1 Y1), (X2 Y2), (X3 Y3) ... (Xn Yn)

Condition 関数にtrue()を評価させる、最初に指示されたの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 ポジュノには1 か返されます。第2 のサンプルでは、33 を返すペアがな サムな、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 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 のアリティを持つ1 つの別数を必要と、boolean を返す XPath 関数に対する参照です。 Condition で参照された関数の代われて、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>サンプル</u>
 - altova:find-first-pos(5 to 10, function(\$a) {\$a mod 2 = 0}) はxs:integer 2 を返します。

Condition 引数は \$a とう名のインライン関数を宣言し、定義します。XPath 3.0 インライン関数 function()を 参照します。 Sequence 引数内の各アイテムでは、find-first-pos が ひされ、代わりに、 \$a を入力値とま す。入力値は関数定義($$a \mod 2 = 0$)内の条件に つしてテスト 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。

<u>更なるサンプル</u>

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。

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つのアトライであることを表示するためです。doc-available#1の意味は以下のとおけです。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ラストするdoc-available() につなされます。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、もう一つの属性がを番目の要素から取得された箇所で、最初の2つの引数は、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、ペアは名前別にファルファベト順にセナー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たは場合、ペアのジョイナー は解除されます。これは、一つのメンバーには飲み構成されることを意味します。関数アイテム(3番目の引数 Function)はアイテ ムのシーケンスである出力であるペアのシーケンス内のペア(ジョイント、ジョイントの解除)に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) は次のアクションにより取得されます: (1+1, ()+2, 3+3, 4+()) 。演算子の一つか空 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(FirstSequence as item()*, SecondSequence as item()*, function(\$i,\$j) {\$i || \$j})をitem()* どる XP3.1 XQ3.1 第3の引数とて与えられた、インライン関数を返します。FirstSequence内の各 \$i を SecondSequence内の各 \$j と結 合します。出力はこれらのアイテムのシーケンスです。 = <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。

for-each-combination [altova:]

altova:for-each-combination(FirstSequence as item()*, SecondSequence as item()*, Function(\$i,\$j){\$i || \$j}) asitem()* XP3.1 XQ3.1 最初の2つの別数内の2つのシーケンスのアイテムは最初のシーケンスの各アイテムが番目のシーケンスはなように結合されるように結 合されます。3番目の別数として与えられる関数は結果のシーケンス内の組み合わせに適用され、アイテムのシーケンスである出力内で
出力されます(サンプル参照)。 ■ <u>サンプル</u> • altova:for-each-combination(('a', 'b', 'c'), ('1', '2', '3'), function(\$i, \$j) {\$i || \$j})は('a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3')を返します。

substitute-empty [altova:]

altova:substitute-empty(FirstSequence as item()*, SecondSequence as item())をitem()* とする XP3.1 XQ3.1

FirstSequence 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)) la(4, 5, 6) を返します。

17.1.2.1.8 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

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 とする XP3.1 XQ3.1

SplitChars を使用して、次の大文字をトリガーする文字を決定し、入力文字列を InputString キャメルケース で変換します。 SplitChars は、IsRegex = true()の場合、おけよ IsRegex = false()の場合、プレ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 (文化てのエ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 定されたポジンタンにある文字を含む文字列を返します。引数により提出されたインデックスに文字が存在してい場合、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 v 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 引数にお指定されている vシュアルゴリズムを使用して InputString から vシュ文字列を生成します。次の vy シュアルゴリズムは(大文字、おけよ 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まい場合、SHA-256 / v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 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:]

altova:first-chars(X-Number as xs:integer) をxs:string とする XP3.1 XQ3.1 xs:string し対するエンテキストアイテムの値を変換することにお得られた最初のX-Number 文字を含む文字列を返します。 *サンプル*

コンテキストアイテムが1234ABCD の場合:

- altova:first-chars(2) は12 を返します。
- altova:first-chars(5) は1234A を返します。
- altova:first-chars(9) は1234ABCD を返します。

altova:first-chars(InputString as xs:string, X-Number as xs:integer) をxs:string とする XP3.1 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(最初の引数)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番目の引数として提出 されている) フォーマトシーケンス内の対応するポジンコ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(文字列)が存在しない場合は、配置/ ラメーターが返されます。これは、配置/ ラメーターのインデッ クスが書式シーケンス内のアイテムの数よ児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>

ゴンテキストアイテムが1234ABCD の場合:

- altova:last-chars(2) はCD を返します。
- altova:last-chars(5) は4ABCD を返します。
- altova:last-chars(9) は1234ABCD を返します。

```
altova:last-chars(InputString as xs:string, X-Number as xs:integer) asxs:string 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 XO3.1

PadCharacter 引数は1文字です。文字列の左側ンジンされ、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(正数まけは負数)を持つこができます が、)ディングは、StringLength の値がStringToPad 内の文字数よ比多い場合の場合のみ発生します。もし、 StringToPad がStringLength の値よ比多くの文字数も持つ場合、StringToPad は変更されません。 ■ サンプル

- altova:pad-string-left('AP', 1, 'Z') は'AP' を返します。
- altova:pad-string-left('AP', 2, 'Z') は 'AP' を返します。
- altova:pad-string-left('AP', 3, 'Z') は'ZAP' を返します。
- altova:pad-string-left('AP', 4, 'Z') は'ZZAP' を返します。
- altova:pad-string-left('AP', -3, 'Z') は 'AP' を返します。
- altova:pad-string-left('AP', 3, 'YZ')は[パド文字が長すぎます] エラーを返します。

pad-string-right [altova:]

altova:pad-string-right(StringToPad as xs:string, StringLength as xs:integer, PadCharacter as xs:string) をxs:string どる XP3.1 XQ3.1

PadCharacter 引数は1文字です。文字列の右側ンドボされ、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(正数まけは負数)を持つこかできます が、パディングは、StringLength の値がStringToPad 内の文字数よ比多い場合の場合の分発生します。もし、 StringToPad がStringLength の値よ比多くの文字数も持つ場合、StringToPad は変更されません。 ■ サンプル

- altova:pad-string-right('AP', 1, 'Z')を'AP'を返します。
- altova:pad-string-right('AP', 2, 'Z')を'AP'を返します。
- altova:pad-string-right('AP', 3, 'Z')を'APZ'を返します。
- altova:pad-string-right('AP', 4, 'Z')を'APZZ'を返します。
- altova:pad-string-right('AP', -3, 'Z')を'AP'を返します。

• altova:pad-string-right('AP', 3, 'YZ') は[パインマクション コン かくしょう。

repeat-string [altova:]

altova:repeat-string(InputString as xs:string, Repeats as xs:integer) をxs:string とする XP2 XQ1 XP3.1 XQ3.1 最初のInputString引数に以構成される文字列、Repeats 回繰い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 にあります。StringToFind がStringToCheck 内で発生した 場合、整数 0 か返されます。第 2 まけまその後のStringToCheck,発生を確認するには、この関数の次の署名を確認してくたさい。

- 🗉 <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 おこ は後の発生を検索する際に役に立ちます。 StringToFind がStringToCheck 内で発生しない場合、整数 0 か返されます。

□ <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(" 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-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-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。

17.1.2.1.9 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MapForce 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、このセグ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

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。エノコードが指定されてよい場合

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 エロード済 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-エノ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ロードを使用して送信された文字列をエレロードします。エロードが与えられていない場合 UTF-8 エレロードが使用されます。エレロードさせた文字列はbase64Binary 文字に変換され、エロードされたbase64Binary 値が返されます。最初にUTF-8 エレロードがサポートされ、サポートは以下のエレロ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 エンコード済み文字列 "Altova" の等価である base64Binary を返します。
- altova:encode-string("Altova", "UTF-8")はUTF-8 エンコード済み文字列 "Altova" の等価であるbase64Binaryのを返します。

get-temp-folder [altova:]

altova:get-temp-folder() をxs:string とする XP2 XQ1 XP3.1 XQ3.1 この関数は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 やを返します。 = サンプル

- 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
ユニークな文字列 GUID 文字列を生成します。
```

🖻 <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 イれぞリューションタイマーの値を返します。ハイれぞリューションタイマーがシステム内で存在すると、必要とされる場合、 高度に正確な時間の計算を有効化します(例えば、アニメーションと正確なコード実行の時間の決定などが例として挙げたれます)。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ドキュン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、おさよ 含まむ 場合かります。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であることが奨励されます。

🗉 <u>サンプル</u>

• altova:parse-html("<html><head/><body><h1>Header</h1></body></html>") は提供された 文字列からHTML ツーを作成します。

sleep[altova:]

altova:sleep(Millisecs as xs:integer) asempty-sequence() XP2 XQ1 XP3.1 XQ3.1 Millisecs に以与えられるいかで示される期間のための実行を延期します。 = <u>サンプル</u>

• altova:sleep(1000)は1000 ミトののための実行を延期します。

[[トップ]]

17.1.2.2 その他の拡張関数

Java やC# などのプログラング言語はよ XPath 2.0 / XQuery 関数、おはXSLT 2.0 関数とて利用できな、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、スタイルシートやかエル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。および XQuery ドキュメン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:

- Java 拡張関数
- <u>.NET 拡張</u>関数
- XSLT に対するMSXSL スクレプト

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 の実行を行うコンピューターに 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、ませばアクセ スできる環境が整っている必要があります。

17.1.2.2.1 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、引数を持たなレッシ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、以下のサブセクションにて記述されます。

このセグションは以下のサブセグションには構成されます:

- Java: エレストラクター
- <u>Java:静的メハドと静的フィールド</u>
- Java: インスタンスメンドとインスタンスフィールド
- <u>データ型: XPath/XQuery からJava へ</u>
- <u>データ型: Java からXPath/XQuery へ</u>

<u>以下の点に注意してください</u>

- Altova デスケ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検知しようとします:(i) Windows レジストリ(ii) JAVA_HOME 環境変数。アプリケーションのオディンダイアログ内にカスタム、やを追加することもできます。このエントリは自動的に検知された他のJava VM / や以上の優先順位を有します。
- Altova サート製品が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。成功したは、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 マシン上でサーンシ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 ディレケリ内の jvm.dll ファイルを指している必要がおます。

拡張関数のフォーム

XPath/XQuery 条件式におお拡張関数では prefix:fname()の形式を取る必要かがます。

- prefix: 部にや拡張関数が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 メンドが識別され、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えた場合、 Java クラスの識別はクラスの前にくる fname() 部にて行うことしてなり、ピノオドによりクラスから分離されることしてなります(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コピューターのクラスマトレーある必要がみあす。

XSLT サンプル

以下に静的メンドを呼び出す2つのサンプルを示します。最初のサンプルでは、クラス名(java.lang.Math)が名前空間 URI に加え られており、fname() へ加えることはできません。2番目のサンプルでは、prefix:部にjava:が与えられており、fname()部に てクラスとメンドが識別されます。

拡張関数内にあるメンボ名(上の列ではcos())は、名前付き Java クラス(上の列ではjava.lang.Math)のpublic な静的メ ンボの名前に一致する必要がみます。

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) 「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<u>ーザー定義クラスファイル</u>ならび、ニユーザー定義 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 ことに注意してくた さい。

17.1.2.2.1.1 ユーザー定義のクラスファイル

アクセスかりラスファイルを介したものである場合、4つのケースか考えられます

- ケラスファイルはパッケージである。XSLT おさはXQuery ファイルがJava パッケージと同じ場所に収められている。(下のサンプ ルを参照)
- ケラスファイルはパッケージでは無い。XSLT おさはXQuery ファイルがJava パッケージと同じ場所に収められている。(下のサン フルを参照)
- ケラスファイルは、シケージである。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。(下のサンプルを参照)
- ケラスファイルはッケージである。XSLT おさよXQuery ファイルカランダムな場所に収められている。(下のサンプルを参照)

クラスファイルがパッケージではなく、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 は目的となるメソドのフラスか含まれているフラスの名前です。

クラスか名前空間 URI にて識別され、名前空間がソンド呼び出しにて使用されます。

クラスファイルかジッケージで、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"</pre>
```

```
</xsl:stylesheet>
```

クラスファイルか参照され、XSLT/XQuery ファイルかりラスファイルと同じフォルダーイコ収められている

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 ルダー内にあります: 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内にあります。

```
<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</a>
```

```
</xsl:stylesheet>
```

クラスファイルかシッケージされ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() メク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してURI内にて指定する必要があります:

java:classname[?path=uri-of-package]

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"
    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>
```

クラスファイルがシケージではよく、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 ケラスのgetCarColor() メン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、ケ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 は目的のメンドが含まれているケラス名です。

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ル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 ノマか外部関数により与えられている場合、ClassLoaderにより、マか追加されます。

17.1.2.2.1.2 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ノバか呼び出されます: 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います。 ? はクラス名といなを分離するために使用されます。 path=jar: により、JAR ファイルへのいなかち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、終了を表すデーミタとなります。 classNS:method() により、メンドの呼び出しが行われます。

その他にも、メノンド名とともにクラス名を与えることができます。構文の例を以下に示します

```
xmlns:ns1="java:docx.layout.pages?
path=jar:file:///c:/projects/docs/docx.jar!/"
    ns1:main()
    xmlns:ns2="java?path=jar:file:///c:/projects/docs/docx.jar!/"
    ns2:docx.layout.pages.main()
```

以下にJAR ファイルを使った Java 拡張関数を呼び出す XSLT サンプルを記します:

```
<xsl:stylesheet version="2.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:fn="http://www.w3.org/2005/xpath-functions"
    xmlns:car="java?path=jar:file:///C:/test/Car1.jar!/" >
    <xsl:output exclude-result-prefixes="fn car xsl xs"/>
    <xsl:template match="/">
        <xsl:template match="/">
        <a><xsl:variable name="myCar" select="car:Car1.new('red')" />
        <a><xsl:vale-of select="car:Car1.getCarColor($myCar)"/></a>
</xsl:template match="/>
</xsl:template match="/>
```

</xsl:stylesheet>

メモ: 拡張関数によりなか与えられている場合、ClassLoader になが追加されます。

17.1.2.2.1.3 Java: コンストラクター

拡張関数を使用することで Java コノストラクターを呼び出すことができます。new()により全てのコレストラクターを呼び出すことができます。

Java コンストラクターの呼び出し結果を、<u>黙示的にXPath/XQuery データ型へ変換</u>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 コンストラクターの呼び出し結果がXPath/XQuery データ型へ変換できない 場合、値を返すクラス名でラップした Java オブジェクト がコンストラクターにより作成されます。例えば、java.util.Date クラスに対する コンストラクターが呼び出された場合(java.util.Date.new())、java.util.Date を持オオブジェクトが返されます。返された オブジェクトのレギシカルフォーマットは、XPath データ型のレギンカルフォーマットにマッチしたよ場合もあり、目的のXPath データ型に対する レギシカルフォーマットへ値の変換を行い、その後目的のXPath データ型へ変換を行う必要があります。

コンストラクターにお作成された Java オブジェクトにお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パニ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" />

17.1.2.2.1.4 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

静的メノバならびにフィールドを呼び出す例を以下に示します

上の拡張関数は prefix:fname() とう形式を使用していること注意してくたさい。3つの例にあるプレファクスは全てjMath: となっており、このプレファクスは、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: で開始しなければぶりません。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 な静的メンドが引数ともに続くかく例: 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>
```

17.1.2.2.1.5 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/変数により作成されます。以下に XSLT サンプルを示します:

<xsl:stylesheet version="1.0" exclude-result-prefixes="date"</pre>

上の例ではノード enrollment/@type の値が以下のようご作成されます:

- 1. java.util.Date クラスマオ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 メノボの引数とてJava オジェクトが ~ スされます。
- 3. getClass メノバには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。

17.1.2.2.1.6 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 斥ティブ型) , java.lang.Float,	
	double(プ 兵ティブ型) , 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) とらメノバも存在する。
- メノバ名か同じで、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。

17.1.2.2.1.7 データ型: Java から XPath/XQuery へ

Java メノナドには植か返され、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。

17.1.2.2.2 .NET 拡張関数

.NET プラオフォームこで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 おけ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 エンストラクター</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こ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、ラメーターが必要にないます。

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、デメーターが無視されます。

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け コロードされ とクラスを特定するものであることか示されます。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)}
</math>
```

</ md L11>

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 式によい呼び出されるメ ンドが特定され、引数が与えられます。

17.1.2.2.2.1 .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()により全てのエンストラクターを呼び出すことができます。ケラス内に2つ以上のエンストラクターがある場合、与えられた引数の数が最もマッチするエ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エンストラクターか見つかびよ、場合、 No constructor found' エラーが返されます。

XPath/XQuery データ型を返すコンストラクター

.NET コンストラクター呼び出しの結果が XPath/XQuery データ型へ黙示的に変換することができる場合、.NET 拡張関数から XPath/XQuery データ型のシーケンスが返されます。

.NET オブジェクトを返すコンストラクター

.NET コンストラ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。

返されたオブシェクトのレキシカルフォーマナトは、目的のXPath データ型と違っている場合があります。その場合、返された値を:(i)目的の XPath データ型のレキシカルフォーマナーへ変換し、(ii)目的のXPath データ型へキャストする必要があります。

コンストラクターにより作成された.NET オブジェクトに対して3つのことを行うことができます:

- 変数内で使用することができます:
 <xsl:variable name="currentdate" select="date:new(2008, 4, 29)"
 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(たさい):
 <xsl:value-of select="date:ToString(date:new(2008, 4, 29))"
 xmlns:date="clitype:System.DateTime" />
- 文字列、数値、おはboolean へ変換することができます:
 <xsl:value-of select="xs:integer(date:get_Month(date:new(2008, 4, 29)))" xmlns:date="clitype:System.DateTime" />

17.1.2.2.2.2 .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つの引数ともにンノボ (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>
```

17.1.2.2.2.3 .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、この変数がを番目のdate 要素エンストラクタートこて 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。

17.1.2.2.2.4 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 メンドのうち、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、以下のレールが適用されます

- クラス内に司名のメンバカをつ以上ある場合、呼び出しに使用された引数の数がマッチするメンバだけが、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)、選択されたメノボ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 メノボが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) メノ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けな 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ります。

17.1.2.2.2.5 データ型:.NET から XPath/XQuery へ

.NET メンボにより値が返される際に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、.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() といす)、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。

17.1.2.2.3 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決ます(以下を参照くた)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 属性によ時定されます(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。

17.2 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

17.2.1 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ジオます: プラオ・フォーム更新済みのWindows 7 SP1、Windows 8、Windows 10プラオ・フォーム更新済みのWindows Server 2008 R2 SP1 おけは以降 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。

17.2.2 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。

17.2.3 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 ンのナックハドキュメント と実装に固有の振る舞い は関しては、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。

17.2.4 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。

17.2.5 インターネットの使用

Altova アプリケーションは、次の状況でインターネット 接続を開始します:

- 登録ダイアログ(「ヘルプ | ソフトウェアのライセンス認証」)内の「評価キーコードをリクエスト」をクリックした場合、登録ダイアロ グボックス内の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、無料の評価キーが顧客に通常の SMTP電子メールを使用して送り返されます。
- Altova 製品の一部では、インターや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、次のプロトコルメノバと接続の「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 サーバーをポト 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、おさま、DTD を参照する XML ドキュメナ、と、URL によ 指定されているドキュメナを開くと、参照されている スキーマドキュメナトは、HTTP 接続(ポート 80) おさま URL によ 指定されている他のプロトコール 上のポイナ 2 参照) によ 地抽出されます。 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、ドキュメナ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アプリケーションでは、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」コマイドを使用する場合、(「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ンス認証と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。

17.3 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
- ソトウェアのアクティベーションとライセノスの計測
- 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 ライセンスの内容を確認するコよ Altova Web サイトのAltova法的な情報のページに移動してくたさい。

17.3.1 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ひに表示され、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ンスファイルから提供される製品のライセンズを購入します。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、製品をアンロッグします。

<u>https://shop.altova.com/</u> 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。

17.3.2 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、ソフトウェアのバージョン、コピュータの名前などのライセンスに関する情報が、お使いのコンピューターとAltova ライセンスサーバー間にてや、取りされます。

お使いのAltova 製品はお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コンピューター(こてプログラムが動作していたい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応えるオーム、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りどめ行われ、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。このようよう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、ケ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ンピューター同士が通信しているのに気づかれるかも知れません。その他の手段により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ンケをブロックすることは勿論可能です。

証明書()関するメモ

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すみに 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。

17.3.3 エンドユーザー使用許諾契約書

- Altova エドューザー使用許諾契約書: <u>http://www.altova.com/ja/legal/eula</u>
- Altova プライバシーポレー: <u>http://www.altova.com/ja/privacy</u>

18 用語

このセクタイコはMapForce およびMapForce に関連する製品に固有のキーワードのノストか含まれています。

コンポーネント

MapForce 内では、「エポーネト」とう用語はデータの構造、スキーマンませまデータかどのようこ変換、関数)されるかを視覚的に表します。 す。エポーネトはすべての<u>マノピング</u>を構築する中心的な役割を果たします。マノピングエリアでは、エンポーネトは正方形の枠で表示され ます。以下は、MapForce エンポーネトの例です:

- 定数
- デーダベース
- フィルター
- 条件
- 関数コンポーネト
- EDI ドキュメント (UN/EDIFACT、ANSI X12、HL7)
- Excel 2007+ ファイル
- 単純型 入力コンポーネント
- 単純型 <u>出力コンポーネント</u>
- スキーマおよびDTD

接続

接続とおっのユネクタ間に描くことができる線です。接続を描くことにより、MapForce にデータを特定の方法で変換するように命令します。 (例 XMLドキュメントからデータを読み込み、他のXMLドキュメントに書き込みます)。

コネクタ

<u>コレポーネト</u>の左右に表示される小さな三角形です。コレポーネトの左に表示されるコネクタはそのコレポーネトのエトリポイトを表します。コレポーネトの右に表示されるコネクタはそのコレポーネトからのデータの終了ポイントを表します。

資格情報

() 認証デーを異なるマピング実行環境で安全なポータブル化する方法を資格情報オブジェクトは提供します。資格情報は基本的な HTTP 認証を必要とするマピング内で役に立ちます。 MapForce と FlowForce Server 内で資格情報を定義することができます。 MapForce 内で資格情報が定義されている場合、マピングデプロイされるように FlowForce Server にデプロイすることができます。

固定長 ファイル(FLF)

データが習慣的に固定値を持つフィールドに分割される共通のテキストフォーマト。(例 行の最初の5 文字がトランザクションID を表し、 次の20 文字がトランザグションの詳細を表すなど)。

FlexText

FlexText とはMapForce によりサポートされる他のフォーマナルで変換するには複雑な非標準まとはレガシーテキストファイルからのデータを 変換するMapForce Enterprise Edition内のモジュールです。

グロー・ジレノノース

Altova グロー・シリノノースはファイル、フォルダー、おさまデータベースへのポータブルなレファレンスです。グロー・シリノノースとして保管されると、 ノマとデータベース接続の詳細は再利用できるようこなりAltova アプリケーション全体で使用可能になります。例えば、複数のAltova デス ケトップアプリケーション内で同じファイルを頻繁に引く必要かある場合、グロー・シリノノースとして定義すると便利な場合かあります。このようこ すると、「ファイルを開く」ダイアログボックスから対応するグローイ ジリノノースを開くことかできるため、ファイルレンを覚えている必要かありません。この方法はファイルレンカ変更されると、ノンの一部を変更するのみという利点かあります。

グロー・シリソースの一般的な使用法は、データベース接続を一度定義し、グロー・シリソースをサポートする全てのAltova アプリケーション で再利用することです。例えば、MapForce マメピングがデザインされるマシンでデータベース接続を作成し、MapForce Server がマメピ ングを実行するマシン上で同じ接続を再利用することができます(これは、一部の場合、両方のマシントコロじデータベースクライア・トソストウェ アがインストールされている場合があります)。

任意で、(「構成」として既知の同じグロー・シリンノースの複数の・シエーションを作成することができます。これによりファイル、おけよフォルダ ー・マ、(おけはデータベース)を必要に応じて変更することができます。例えば、「開発」と「生産」の構成を持つ「データベース」リンースを作成 することができます。MapForceを例に挙げると、マピングをプレビューする前に、希望する構成をリストから選択して、開発、おけは生産デ ータベースからデータを簡単に抽出することができます。

次のAltova デスケップアプリケーションからグロー シレレンースを作成することができます: Altova Authentic、DatabaseSpy、 MobileTogether Designer、MapForce、StyleVision、およびXMLSpy。サード 製品では、次のAltova サードーアプレケー ションでグロー シレレンースを作成することができます: FlowForce Server、MapForce Server、RaptorXML Server、 RaptorXML+XBRL Server。

入力コンポーネント

入力コンポーネントは、単純型の値をマッピングコンマすることができるMapForce <u>コンポーネント</u>です。入力コンポーネントは通常ファイル 名ます」まその他の文字列の値をランタイムニマッピングコンマするオータリニ使用されます。入力コンポーネントと<u>ノースコンポーネント</u>を混同しない ように注意してください。

ジョインコンポーネント

ジョインエンポーネトは、カスタムに定義付けられた条件をベースは、て2つの構造をジョインすることを有効化するMapForce <u>エルポーネン</u> 上です。条件を満たすアイテムのアノシエーション(ジョインされたセナ)を返します。ジョインは、(ID などの)共通のフィールドを共有する2つ の構造からのデーダを結合する際にとても役に立ちます。

MapForce

MapForce は、プログラムコードを作成することよく視覚的なドラッグアンドドロップを使用するグラフィカルなコー・ザーインターフェイスを用いデ 一弦異なるデータ間、スキーマ間で変換するWindows ベースの多目的のIDE (統合された開発環境 統合開発環境)です。また、 事実上、MapForce は時際のデータ変換(まっデータマンピング)を行うプログラムコードを生成します。プログラムコードを生成しない場合、 (MapForce Professional まさ」はEnterprise Editionsで使用することのできる MapForce 内蔵の変換言語を実行します。

マピング

MapForce マピングデザイン(おまま 略して「マピング」)は、データがどのようこうのフォーマトから他のフォーマトに変換さえるかの 視覚的な表現です。マピングは、データ変換を行うすめのMapForce マピングエルア内で「つのスキーマから他のスキーマ」に追加するユン <u>ポーネント</u>から構成されています。(例: XMLドキュメントを1つのスキーマから他のスキーマ」で変換)。有効なマピングは、1つまけよ複数のタ <u>ーゲナコンポーネント</u>に接続されている、1つまけよ複数の複数のノースコンポーネントから構成されています。マピングを実行して、結果を 直接 MapForce 内で確認することができます。コードを生成し、外部で実行することも可能です。MapForce 実行可能ファイルにマピ ングをコンパイルし、MapForce Server ますはま FlowForce Server を使用してマノピングの実行を自動化することもできます。 MapForce は、マピングを、mfdの拡張子を持つファイルとして保存します。

MFF

MapForce 関数ファイルのファイル名拡張子

MFD

MapForce デザインドキュメント(マナング)のファイル名拡張子

MFP

MapForce プロジェクトファイルのファイル名拡張子

出力コンポーネント

出力コンポーネト(おけば「単純型出力」)とは、マンピングから文字列の値を返すことを有効化するMapForce <u>コンポーネト</u>です。出力コンポーネトは、1つの<u>ターゲトコンポーネト</u>型のみをあられしますが、後者と混同しないように注意してくたさい。

parent-context

min、max、avg、count などのMapForce 一部のコア集計関数 parent-context 任意の引数です。複数の階層的シーケン スを持つノースコンポーネトでは、親コンテキストは、ノードのセナかどの関数上で操作されるかを決定します。

ソースコンポーネント

ソースコンポーネントとは、MapForce がデータを読み込む元のユンポーネントです。マンピングを実行すると、ソースコンポーネントのコネクタ により与えられたデータを読み込み、必要とされる型に変換し、ターゲットコンポーネントのコネクタに送信します。

ターゲナコンポーネント

ターゲトコンポーネントとMapForce はガデーダを書き込むコンポーネントです。マンピングを実行すると、ターゲトコンポーネントは、外部プログラム内で更に処理するためこMapForce にファイルますは複数のファイルの生成、ませば結果を文字列の値として出力するように命令します。ターゲトコンポーネントは、ソースコンポーネントの逆です。

インデックス

•

.NET 拡張関数, XSLT と XQuery, 1529 インスタンスメソッド、インスタンスフィールド, 1533 コンストラクター, 1531 データ型変換、.NET から XPath/XQuery へ, 1534 データ型変換、XPath/XQuery から .NET へ, 1534 概要, 1529 静的メソッド、静的フィールド, 1532 .NET 内のXSLT と XQuery のための拡張機能, .NET 拡張関数を参照する, 1529

Α

A から Z. 並べ替えコンポーネント, 194 abs. MapForce 関数として (lang | math 関数), 859 MapForce 関数として(xpath2 | numeric 関数), 912 acos, MapForce 関数として (lang | math 関数), 860 ActiveX, アプリケーションレベルの統合, 1382 ドキュメントレベルの統合, 1385 統合の必要条件, 1378 ActiveX コントロール. Visual Studio ツールボックスへの追加, 1380 add, MapForce 関数として (core | math 関数), 784 ADO, データ接続インターフェイスとして,343 接続のセットアップ,349 ADO.NET. 接続のセットアップ,355 age. MapForce 関数として (lang | datetime 関数), 836 Altova XML パーサー, について、1538 Altova エンジン,

Altova 製品内で, 1538 Altova 拡張関数, チャート関数 (チャート関数を参照), 1450 Ant, Build Java project with, 1073 building Java code with, 1076 setting the environment variables for,, 1183 Any, xs:any, 332 API, documentation, 1203 Application object, 1207 asin, MapForce 関数として (lang | math 関数), 860 atan. MapForce 関数として (lang | math 関数), 860 ATTLIST, DTD 名前空間 URI, 323 auto-number, MapForce 関数として (コア | generator 関数), 776 avg, MapForce 関数として (core | aggregate 関数), 752

B

```
base-uri,
MapForce 関数として(xpath2 | accessors ライブラリ), 884
boolean,
MapForce 関数として (core | conversion 関数), 758
build.xml,
enabling the zip64mode in,, 1183
Built-in エンジン,
使用, 85
定義, 85
```

С

C#,

code, 1055 code generation settings, 1184 error handling, 1208 generating program code, 1064 integrate generated code, 1088 MapForce の統合, 1388

C#,

reference to generated classes, 1153 C++. code, 1055 code generation settings, 1184 error handling, 1208 generating program code, 1059 integrate generated code, 1090 reference to generated classes, 1138 capitalize, MapForce 関数として (lang | string 関数), 869 **CDATA. 330** ceiling. MapForce 関数として (core | math 関数), 785 char-from-code, MapForce 関数として (core | string 関数), 820 charset-decode, MapForce 関数として (lang | string 関数), 869 charset-encode, MapForce 関数として (lang | string 関数), 871 Class ID, MapForce 統合, 1395 Code. built in types, 1198 integrating MapForce code, 1085 SPL. 1186 Code generation, sample, 1221 Code Generator, 1055 code-from-char. MapForce 関数として (core | string 関数), 820 COM API. documentation, 1203 concat (関数として) サンプルの使用方法, 621 MapForce 関数として (core | string 関数), 821 convert-to-utc, MapForce 関数として (lang | datetime 関数), 837 cos. MapForce 関数として (lang | math 関数), 861 count. MapForce 関数として (core | aggregate 関数), 753 count-substring. MapForce 関数として (lang | string 関数), 873 create-guid, MapForce 関数 (lang | generator 関数), 857 CSV. マッピングソースとして,610

```
階層の作成 - キー, 614
  空のフィールドの置換,664
  作成 複数の 行,612
CSV ファイル,
  ソース コンポーネントとして, 617
  ターゲットコンポーネントとして,617
  のエンコードを設定,617
  プレビュー データ,617
  追加または削除フィールド,617
current-date,
  MapForce 関数として(xpath2 | context 関数), 888
current-dateTime,
  MapForce 関数として(xpath2 | context 関数), 888
current-time.
  MapForce 関数として(xpath2 | context 関数), 888
currrent.
  MapForce 関数として(xslt | xslt 関数ライブラリ), 940
```

D

Database connection, setup examples, 372 date-from-datetime. MapForce 関数として (lang | datetime 関数), 837 datetime-add. MapForce 関数として (lang | datetime 関数), 838 datetime-diff, MapForce 関数として (lang | datetime 関数), 839 datetime-from-date-and-time, MapForce 関数として (lang | datetime 関数), 840 datetime-from-parts, MapForce 関数として (lang | datetime 関数), 841 day-from-datetime, MapForce 関数として (lang | datetime 関数), 842 day-from-duration, MapForce 関数として (lang | datetime 関数), 842 DB. ORDER BY, 524 default-collation, MapForce 関数として(xpath2 | context 関数), 889 degrees, MapForce 関数として (lang | math 関数), 861 distinct-values. MapForce 関数 (core | sequence 関数)として, 796 divide, MapForce 関数として (core | math 関数), 785

divide-integer, MapForce 関数として (lang | math 関数), 862 document, MapForce 関数として(xslt | xslt 関数ライブラリ), 940 DoTransform.bat, RaptorXML Server を使用して実行, 945 DTD, ソースとターゲット, 323 duration-add, MapForce 関数として (lang | datetime 関数), 843 duration-from-parts, MapForce 関数として (lang | datetime 関数), 843 duration-subtract, MapForce 関数として (lang | datetime 関数), 845

E

Eclipse, compiling Java code with, 1076 generating mapping code for, 1071 Eclipse のための MapForce プラグイン, creating a MapForce/Eclipse プロジェクト, 1020 Eclipse プロジェクトにマッピングをインポートする, 1024 MapForce パースペクティブに切り替える,1014 インストール、1010 について, 1009, 1014, 1020 マッピングとプロジェクトとの作業,1020 拡張機能, 1030 拡張点, 1030 関数共通のメニューと関数の評価, 1017 自動コードの生成の設定,1027 新規マッピングの作成,1022 element-available, MapForce 関数として(xslt | xslt 関数ライブラリ), 941 empty, MapForce 関数として (lang | string 関数), 874 Environment variables, ANT_OPS, 1183 equal, MapForce 関数として (core | logical 関数), 779 equal-or-greater, MapForce 関数として (core | logical 関数), 779 equal-or-less, MapForce 関数として (core | logical 関数), 780 Error handling, general description, 1208

```
exists,
```

MapForce 関数 (core | sequence 関数)として, 798 exp,

MapForce 関数として (lang | math 関数), 862

F

```
false.
  MapForce 関数として(xpath2 | boolean 関数), 886
find-substring,
  MapForce 関数として (lang | string 関数), 874
Firebird.
  ODBC を使用しての接続, 374
first-items.
  MapForce 関数 (core | sequence 関数)として, 799
floor.
  MapForce 関数として (core | math 関数), 786
FlowForce Server.
  グローバルリソース, 973
  グローバルリソースのデプロイ, 955, 974
  マッピングのデプロイ,955
format-date.
  MapForce 関数として (core | conversion 関数), 758
format-dateTime.
  MapForce 関数として (core | conversion 関数), 759
format-guid-string,
  MapForce 関数として (lang | string 関数), 875
format-number.
  MapForce 関数として (core | conversion 関数), 762
format-time.
  MapForce 関数として (core | conversion 関数), 765
function-available.
  MapForce 関数として(xslt | xslt 関数ライブラリ), 941
Functions.
  finding in the Libraries window, 642
  finding occurences in active mapping, 642
```

G

```
Generate,
code from schema, 1055
generate-id,
MapForce 関数として(xslt | xslt 関数ライブラリ), 941
generate-sequence,
```

generate-sequence, MapForce 関数 (core | sequence 関数)として, 800 get-fileext. MapForce 関数として (core | file path 関数), 772 get-folder. MapForce 関数として (core | file path 関数), 773 Global objects. in SPL. 1190 greater, MapForce 関数として (core | logical 関数), 780 group-adjacent, MapForce 関数 (core sequence 関数)として, 801 group-by, MapForce 関数 (core | sequence 関数)として, 803 group-ending-with, MapForce 関数 (core | sequence 関数)として, 805 group-into-blocks, MapForce 関数 (core | sequence 関数)として, 806 group-starting-with, MapForce 関数 (core | sequence 関数)として, 808

Η

hour-from-datetime, MapForce 関数として (lang | datetime 関数), 846 hour-from-duration, MapForce 関数として (lang | datetime 関数), 846 HRESULT, and error handling, 1208 HTML, MapForce の統合, 1397 マッピングドキュメンテーションの生成, 267 マッピングの出力のプレビュー, 264 HTML サンプル.

MapForceControl 統合, 1395, 1396, 1397 of MapForceControl 統合, 1395

I

IBM DB2, JDBC を介した接続, 376 ODBC を使用しての接続, 378 XML 型フィールドから読み込む, 538 XML 型フィールドへ書き込む, 538

```
IBM DB2 for i,
  ODBC を使用しての接続, 386
IBM Informix,
  connecting through JDBC, 385
  JDBC を使用しての接続, 389
If-Else 条件.
  マッピングに追加,205
implicit-timezone.
  MapForce 関数として(xpath2 | context 関数), 889
INNER JOIN.
  ジョインコンポーネント,509
Integrate,
  into C#, 1088
  into C++. 1090
  into Java, 1086
Integrate MapForce code, 1085
is-not-null,
  MapForce 関数として (db 関数), 834
is-null.
  MapForce 関数として (db 関数), 834
is-xsi-nil,
  MapForce 関数として (core | node 関数), 790
item-at.
  MapForce 関数 (core | sequence 関数)として, 809
items-from-till.
  MapForce 関数 (core | sequence 関数)として, 811
```

J

```
Java, 1401
  avoiding exceptions in generated code, 1183
  code. 1055
  generating program code, 1071, 1076
  integrate generated code, 1086
  reference to generated classes, 1168
Java 拡張関数.
  XSLT と XQuery, 1521
  インスタンスメソッド、インスタンスフィールド, 1527
  コンストラクター, 1526
  データ型変換、Java から XPath/XQuery へ, 1529
  データ型変換、XPath/XQuery から Java へ, 1528
  ユーザー定義の JAR ファイル, 1525
  ユーザー定義のクラスファイル、1523
  概要, 1521
  静的メソッド、静的フィールド, 1527
JavaScript,
```
JavaScript, error handling, 1208 JDBC, handling references in generated code, 1075 Teradata への接続, 421 データ接続インターフェイスとして, 343 接続の設定 (Windows), 365 JScript,

code-generation sample, 1221

last MapForce 関数として(xpath2 | context 関数), 889 last-items, MapForce 関数 (core | sequence 関数)として, 812 leapyear, MapForce 関数として (lang | datetime 関数), 847 left, MapForce 関数として (lang | string 関数), 876 LEFT OUTER JOIN, ジョインコンポーネント, 509 left-trim. MapForce 関数として (lang | string 関数), 876 less, MapForce 関数として (core | logical 関数), 781 Libraries window, finding functions in, 642 Library, 1199 local-name-from-QName. MapForce 関数として (lang | QName 関数), 794 log, MapForce 関数として (lang | math 関数), 863 log10. MapForce 関数として (lang | math 関数), 863 logical-and. MapForce 関数として (core | logical 関数), 781 logical-not. MapForce 関数として (core | logical 関数), 782 logical-or, MapForce 関数として (core | logical 関数), 782 logical-xor, MapForce 関数として (lang | logical 関数), 857 Look-up テーブル, マッピング上での使用,223 lowercase,

MapForce 関数として (lang | string 関数), 877

Μ

main-mfd-filepath, MapForce 関数として (core | file path 関数), 773 MapForce, API. 1203 概要,28 基本概念, 33 統合, 1377 MapForce API, 1203 MapForce Server, から例外をスロー,232 グローバルリソース,972 のためにマッピングをコンパイルする,952 マッピングの自動化,946 MapForce サンプル, ディスク上の場所,44 MapForce 統合, のサンプル, 1395, 1396, 1397 MapForceControl, 1425 C# を使用した統合, 1388 HTML を使用下統合, 1397 Visual Basic を使用した統合, 1410 アプリケーションレベルでのサンプル 統合, 1395, 1396, 1397 オブジェクトレファレンス, 1422 ドキュメント, 1377 ドキュメントレベルでの統合のサンプルサンプル, 1388 MapForceControl ドキュメント, 1432 MapForceControl プレースホルダ, 1439 MapForceコマンド, MapForceControl, 1422 MapForceControl 内, 1424 MariaDB, ODBC を使用した接続, 391 match-pattern, MapForce 関数として (lang | string 関数), 877 max. MapForce 関数として (core | aggregate 関数), 754 MapForce 関数として (lang | math 関数), 863 maxLength, ノード関数内での使用,661 max-string, MapForce 関数として (core | aggregate 関数), 754 MERGE.

MERGE. MapForce-により生成された SQL ステートメントとして, 477 MapForce-により生成された SQL 内のステートメントとして,470 Method names in generating code, reserving, 1183 mfd-filepath, MapForce 関数として (core | file path 関数), 774 mff. mff.xsd ファイル, 712 ライブラリファイル,712 mff ファイル. 構成 712 Microsoft Access, ADO を介しての接続, 349, 393 Microsoft SharePoint Server, ファイルをコンポーネントとして追加,81 Microsoft SQL Server, ADO を介しての接続, 396 ODBC を使用しての接続, 398 millisecond-from-datetime. MapForce 関数として (lang | datetime 関数), 847 millisecond-from-duration, MapForce 関数として (lang | datetime 関数), 848 min. MapForce 関数として (core | aggregate 関数), 755 MapForce 関数として (lang | math 関数), 864 minLength. ノード関数内での使用,661 min-string. MapForce 関数として (core | aggregate 関数), 755 minute-from-datetime. MapForce 関数として (lang | datetime 関数), 848 minute-from-duration, MapForce 関数として (lang | datetime 関数), 849 modulus, MapForce 関数として (core | math 関数), 786 month-from-datetime, MapForce 関数として (lang | datetime 関数), 849 month-from-duration, MapForce 関数として (lang | datetime 関数), 850 MSXML. generating code for, 1184 MSXSL スクリプト内の拡張関数, 1535 msxsl:script, 1535 multiply. MapForce 関数として (core | math 関数), 787 MySQL,

ODBC を使用しての接続, 404

Ν

namespace-uri-form-QName, MapForce 関数として (lang | QName 関数), 795 negative. MapForce 関数として (lang | logical 関数), 858 nillable, XML スキーマ内の属性として, 327 node-name, MapForce 関数として (core | node 関数), 792 MapForce 関数として(xpath2 | accessors ライブラリ), 884 node-name 関数, 243 代替の使用方法,243 normalize-space, MapForce 関数として (core | string 関数), 823 not-equal. MapForce 関数として (core | logical 関数), 783 not-exists, MapForce 関数 (core | sequence 関数)として, 813 now. MapForce 関数として (lang | datetime 関数), 850 NULL. データベースコンポーネントの処理,453 データベースマッピング内での扱い, 498 NULL 值, 複数の発生を置き換える,645 number. MapForce 関数として (core | conversion 関数), 766 numeric, MapForce 関数として (lang | logical 関数), 858

0

Object model, overview, 1207 ODBC, MariaDB への接続, 391 Teradata への接続, 423 データ接続インターフェイスとして, 343 接続のセットアップ, 362 ODBC ドライバー, 使用可能か確認する, 362 OLE DB, データ接続インターフェイスとして, 343 Oracle, XML 型フィールドから読み込む, 538 XML 型フィールドへ書き込む, 538 Oracle データベース, JDBC を使用しての接続, 406 ODBC を使用しての接続, 407 ORDER BY, SQL where コンポーネント, 524 OS, Altova 製品のための, 1538 Out of memory exceptions, resolving, 1183

Ρ

pad-string-left, MapForce 関数として (lang | string 関数), 878 pad-string-right, MapForce 関数として (lang | string 関数), 879 parse-date. MapForce 関数として (core | conversion 関数), 766 parse-dateTime, MapForce 関数として (core | conversion 関数), 768 parse-number, MapForce 関数として (core | conversion 関数), 770 parse-time. MapForce 関数として (core | conversion 関数), 771 PDF. マッピングドキュメンテーションの生成,267 マッピングの出力のプレビュー、264 pi, MapForce 関数として (lang | math 関数), 865 position, MapForce 関数 (core | sequence 関数)として, 813 positive, MapForce 関数として (lang | logical 関数), 859 PostgreSQL, ODBC を使用しての接続, 413 直接 (ネイティブ)接続, 367 直接接続する(ネイティブ),369 pow, MapForce 関数として (lang | math 関数), 865 Progress OpenEdge データベース, JDBC を使用しての接続, 416

ODBC を使用しての接続, 417

Q

QName, MapForce 関数として (lang | QName 関数), 794 QName サポート, 326 QName-as-string, MapForce 関数として (lang | QName 関数), 868

R

radians. MapForce 関数として (lang | math 関数), 865 random, MapForce 関数として (lang | math 関数), 866 RaptorXML Server, 変換の実行,945 read-binarv-file. MapForce 関数として (lang | file 関数), 855 remove-fileext. MapForce 関数として (core | file path 関数), 774 remove-folder. MapForce 関数として (core | file path 関数), 774 remove-timezone. MapForce 関数として (lang | datetime 関数), 851 repeat-string. MapForce 関数として (lang | string 関数), 879 replace, MapForce 関数として (lang | string 関数), 880 replace-fileext, MapForce 関数として (core | file path 関数), 775 replicate-item, MapForce 関数 (core | sequence 関数)として, 815 replicate-sequence, MapForce 関数 (core | sequence 関数)として, 817 resolve-filepath. MapForce 関数として (core | file path 関数), 776 resolve-uri. MapForce 関数として(in xpath2 | anyURI 関数), 885 reversefind-substring, MapForce 関数として (lang | string 関数), 880 right, MapForce 関数として (lang | string 関数), 881

right-trim, MapForce 関数として (lang | string 関数), 882 round, MapForce 関数として (core | math 関数), 787 round-half-to-even, MapForce 関数として(xpath2 | numeric 関数), 912 round-precision, MapForce 関数として (core | math 関数), 788 RTF, マッピングドキュメンテーションの生成, 267 マッピングの出力のプレビュー, 264

S

Schema, code generator, 1055 schemanativetype, 1187 Search, functions in the Libraries window, 642 second-from-datetime, MapForce 関数として (lang | datetime 関数), 851 second-from-duration. MapForce 関数として (lang | datetime 関数), 852 set-empty, MapForce 関数 (core | sequence 関数)として, 818 set-null, MapForce 関数として (db 関数), 834 set-xsi-nil. MapForce 関数として (core | node 関数), 792 sin. MapForce 関数として (lang | math 関数), 866 skip-first-items. MapForce 関数 (core | sequence 関数)として, 818 SPL, 1186 code blocks, 1186 conditions, 1193 foreach, 1194 global objects, 1190 subroutines, 1195 using files, 1191 variables, 1189 SQL. 552 SQL スクリプトのインポート, 553 SOL スクリプトをとして文字列ステートメントをエクスポートする、 553 スクリプトからロード, 551

ステートメントの作成.552 ステートメントの実行,551,553 ステートメントの生成,551,552 データのジョイン,509 SQL Server, ADO.NET を使用して接続する, 355 XML 型フィールドから読み込む, 538, 539 XML 型フィールドへ書き込む, 538, 539 SQL WHERE. ORDER BY, 524 コンポーネント - 挿入, 524 SQL WHERE/ORDER. MapForce コンポーネントとして, 527 SQL エディター. コメントの挿入,553 テキストのコメントアウト, 553 ブックマークの挿入,554 ブックマーク余白,554 作成 領域, 555 削除 コメント, 553 削除 ブックマーク, 554 削除 領域, 555 使用 ブックマーク, 554 使用 領域, 555 設定 - 一般, 563 領域の挿入,555 SQL サーバー. ADO を介しての接続, 349 SQLite. XML ファイルを書き込む, 542 データのマッピング,504 へのデータのマッピング,238 生成されたコード内でデータベースパスを絶対パスにする、 141 相対と絶対パスの使用,138 sqrt, MapForce 関数として (lang | math 関数), 867 starts-with, MapForce 関数として (core | string 関数), 823 static-node-annotation. MapForce 関数として (core | node 関数), 792 static-node-name, MapForce 関数として (core | node 関数), 793 string. MapForce 関数として (core | conversion 関数), 772 MapForce 関数として(xpath2 | accessors ライブラリ), 884 string-as-QName, MapForce 関数として (lang | QName 関数), 868

string-compare-ignore-case, MapForce 関数として (lang | string 関数), 882, 883 string-join. MapForce 関数として (core | aggregate 関数), 756 string-length. MapForce 関数として (core | string 関数), 824 Stylevision. スタイルシートの作成,273 マッピングドキュメンテーションの生成, 268 substitute-missing, MapForce 関数 (core | sequence 関数)として, 819 substitute-missing-with-xsi-nil, MapForce 関数として (core | node 関数), 793 substitute-null. MapForce 関数として (db 関数), 835 substring. MapForce 関数として (core | string 関数), 824 substring-after, MapForce 関数として (core | string 関数), 825 substring-before, MapForce 関数として (core | string 関数), 825 subtract, MapForce 関数として (core | math 関数), 788 sum. MapForce 関数として (core | aggregate 関数), 757 Sybase, JDBC を使用しての接続, 420 system-property, MapForce 関数として(xslt | xslt 関数ライブラリ), 942

Τ

tan, MapForce 関数として (lang | math 関数), 867 Teradata, JDBC を使用した接続, 421, 423 time-from-datetime, MapForce 関数として (lang | datetime 関数), 852 timezone, MapForce 関数として (lang | datetime 関数), 853 tokenize, MapForce 関数として (core | string 関数), 826 tokenize-by-length, MapForce 関数として (core | string 関数), 828 tokenize-regexp, MapForce 関数として (core | string 関数), 828 translate (core | string 関数), MapForce 関数として, 832 true, MapForce 関数として(xpath2 | boolean 関数), 886 Types, built in, 1198 派生した型 - xsi:type, 323

U

```
UDF.
  マッピングコンテキスト, 279
unary-minus,
  MapForce 関数として (lang | math 関数), 867
Unicode.
  コードポイント 照合, 194
  特別文字の置き換え.452
Unicode のサポート,
  Altova 製品内で, 1538
unparsed-entity-uri,
  MapForce 関数として(xslt | xslt 関数ライブラリ), 942
uppercase,
  MapForce 関数として (lang | string 関数), 883
URI.
  DTD 内, 323
  と QNames, 326
URL,
  ファイルをコンポーネントとして追加,81
```

V

```
Value-Map,
サンプル, 226, 228
マッピングコンポーネントとして, 223
Variables,
in SPL, 1189
Visual Basic,
error handling, 1208
MapForce の統合, 1410
Visual Studio,
generating code for, 1184
generating mapping code for, 1059, 1064
MapForce ActiveX コントロールをツールボックスへ追加,
1380
Visual Studio のための MapForce プラグイン,
```

Visual Studio のための MapForce プラグイン, について, 1003 マッピングとプロジェクトとの作業, 1006 共通のメニューと関数にアクセスする, 1007 有効化, 1004 Visual Studio プラグイン, として MapForce を実行する, 1003

W

WebDAV Server, ファイルをコンポーネントとして追加,81 weekday, MapForce 関数として (lang | datetime 関数), 853 weeknumber. MapForce 関数として (lang | datetime 関数), 853 WHERE, SQL WHERE コンポーネント, 524 Windows Altova 製品のためのサポート, 1538 Word マッピングドキュメンテーションの生成,267 Word 2007+. マッピングの出力のプレビュー、264 Wrapper classes, in generated code, 1184 write-binary-file, MapForce 関数として (lang | file 関数), 856

Χ

Xerces, generating code for, 1184 XML, CSV からのデータマッピング, 610 データベース フィールドへ書き込む, 542 マッピングターゲットとして, 610 XML カタログ, しくみ, 999 構成, 999 XML から XML へ, 318 XML データ, データベース フィールドから読み込む, 538 データベース フィールドへ書き込む, 538 XML パーサー. について, 1538 XML ファイル. データベースレコードから生成する,168 単一の XML ソースから生成する, 166 XML 出力. インスタンス ファイル名の変更, 319 エンコード設定の変更.319 スキーマの変更,319 作成 デジタル署名,319 XML 宣言. 出力からの抑制,319 XMLSpy コマンドテーブル, 1413 XQuerv. カスタム関数の追加,700,701 モジュールのインポート,701 拡張関数, 1521 生成されたコードのプレビュー,98 XQuery プロセッサー, Altova 製品内で, 1538 xs: any (xs:anyAttribute), 332 xsi:nil. XML インスタンス内の属性として, 327 xsi:type, マッピングから派生した型,323 XSLT. カスタム関数の削除,694 カスタム関数の追加,694 テンプレート名前空間, 694 拡張関数, 1521 生成されたコードのプレビュー,97 XSLT と XQuerv のためのJava 拡張関数. Java 拡張関数を参照する, 1521 XSLT と XQuery のための拡張関数, 1521 XSLT プロセッサー, Altova 製品内で, 1538 XSLT/XQuery 内のスクリプト, 拡張関数で参照する,1521

Y

year-from-datetime, MapForce 関数として (lang | datetime 関数), 854 year-from-duration, MapForce 関数として (lang | datetime 関数), 854

Ζ

Z to A. 並べ替えコンポーネント, 194 zip64mode. enabling in the build.xml file, 1183 アイコン. データベース クエリのメッセージウィンドウ, 560 データベース クエリの結果ウィンドウ,560 アイテム, 不足している.124 インスタンス, ファイル参照をに変更する.136 インターネットの使用. Altova 製品内で, 1539 エンコード, CSV ファイル内の変更, 617 フラットテキストファイルの変更, 627 エンコード設定, XML 出力, 319 エンドユーザー使用許諾契約書, 1540, 1541 オプション. テキスト フォント - データベース クエリ, 565 結果表示 - データベース クエリ,564 カスタムライブラリ, 追加, 712 キー,614 テキストファイルからのマッピング,614 テキストファイル内のフィールドテキスト, 614 並べ替えキー,194 キーと値のペア. マッピング上での使用,223 グローバルリソース. FlowForce Server にデプロイする, 955, 974 FlowForce Server 内, 972 FlowForce Server 内の, 973 はじめに,964 作成,965 使用例, 976, 978, 980, 984 コードポイント, 照合,194 コード生成設定. プロジェクト全体のためにグローバルに定義する、131 プロジェクト内のフォルダーのために定義する、132 コネクタ,

処理された値の履歴の確認、305 全てコピー,151 コメント. ターゲット ファイルに追加, 329 コンテキストウィンドウ. について, 299, 307 コンポーネント. アプリケーションメニューとして, 1039 データの並べ替え,194 マッピングに追加,80 概要,103 検索, 104 削除されたアイテム, 124 整列.105 設定の変更,106 シーケンス、276 スキーマ. XML ファイルのために生成する, 318 と XML マッピング, 318 ファイル参照をに変更する,136 スケール. ノード関数内での使用,661 ストアドプロシージャ サポートメモ,566 マッピングに追加,568 マッピングの呼び出し、566、571、574、578、582、587 セクション. **CDATA**, 330 ソース優先, - 混在コンテンツマッピング,143 ソース優先接続, 通常 (ターゲット優先) 接続に対して, 150 ソフトウェア製品ライセンス, 1541 ターゲットコンポーネント. 処理順序の変更,291 ターゲット優先 マッピング, 143 ターゲット優先接続, ソース優先接続に対して,150 ツール. アプリケーションメニューとして, 1048 データオーバーレイ. について、299 データストリーミング. 定義,84 データのオーダー. 並べ替えコンポーネント, 194 データの統合.

```
XML ファイルのマージ, 337
```

データベース. XML スキーマをフィールドに割り当てる,539 XML ファイルを書き込む、542 クエリ, 549 グローバルリソースとして,966 シーケンスになった一意の値を生成,453 データを複数にリンクされているテーブルに挿入する,463 テーブルデータの更新.467 テーブルデータの削除,484 テーブルにデータを挿入する,459 テーブルの更新と挿入,470 にデータをマージする,470 に対してマッピングを実行する,450 バルク挿入の使用.495 マッピングコンテキスト,278 マッピングに追加,428 複数の XML ファイルから生成する, 168 データベース クエリ SQL の生成, 555, 560 コメントアウトテキスト,553 テキスト フォント オプション, 565 フィルタリング テーブル, 557 ブックマーク.554 結果表示 オプション, 564 領域,555 データベース リレーションシップ, マッピング内で定義,443 マッピング内に保持,436 データベースオブジェクト. フィルタリング,430 マッピングから削除,430 マッピングへ追加,430 データベースドライバー. 概要.346 データベーストランザクション. ストアドプロシージャのために有効化.578 マッピング内で使用,491,591 ロールバック, 594 データベースへのアクセス. IF 条件をベースにした更新, 621 データベース接続. ウィザードの開始,344 グローバルリソースからの再使用、371 設定,343 データベース例外. ログ,605 処理, 594 テーブルデータ,

並べ替え 194 テーブル名からスキーマ名を削除する, コード生成オプションとして、433 テキスト. CSV ならびに固定長テキストファイルから階層構造を作成,614 マッピング テキストファイル, 610 テキストファイル. エンコードの設定,627 からのデータのプレビュー, 627 ソース コンポーネントとして, 627 ターゲットコンポーネントとして, 627 データのマッピング.621 フィールドの追加または削除, 627 固定されたフィールドサイズの設定. 621 充填文字の設定,621 デジタル署名. XML 出力の作成, 319 デバッガーポジション. の現在の値野確認,305 デバッグ. ステップバイステップ,294 について, 299 のための準備,297 ブレークポイントと,294 開始, 298 制限, 294 設定.316 停止, 298 デフォルトのスキーマに対して相対的にオブジェクト名を使 用する. オプションとして、433 デフォルトの値, 複数のアイテムに適用,645 ドキュメントレベル, XMLSpy の統合のサンプル, 1388 ナビゲート. ブックマーク.554 ノード関数. 削除. 649 編集. 649 ノード名. データをマッピング,243 パーサー. Altova 製品に内蔵の, 1538 バイナリファイル. へのデータの書き込み,636 マッピングに追加する,633 以下からのデータの読み込み,634

バイナリファイル, 概要,633 バッチ. 自動化された 処理,946 パラメーター. マッピングに提供する,170,175 ユーザー定義関数内,674 バリデーター. Altova 製品内で, 1538 ファイル. アプリケーションメニューとして, 1032 コンポーネントとしてのボタン,106 コンポーネントのボタンとして, 161 ファイル - キーフィールドの定義. フィールドテキストファイル内のキー, 614 ファイル DSN. セットアップ、362 ファイル/文字列. コンポーネントとしてのボタン, 106 コンポーネントのボタンとして, 161 ファイル: (デフォルト), ルートノードの名前として,161 ファイル: <dynamic>, ルートノードの名前として,161 ファイルパス. ファイルベースのデータベースの, 138 生成されたコード,141 相対と絶対, 136, 141 破損した参照の修正,140 ファイル名. マッピング入力パラメーターとして提供する,165 フィルター. XML ファイルのマージ, 337 オンラインブラウザー.557 データベース オブジェクト, 557 データベース内.527 マッピングに追加,205 全てコピー コネクタ, 151 フィルタリング、 コンポーネントからのデータ,205 データベーステーブル, 205 マッピング上のデータベースオブジェクト,430 フォルダー. レイアウト - データベース クエリ, 555 ブックマーク. ナビゲート,554 ブックマーク余白、554 削除,554

插入.554 ブラウザー. データベース クエリ, 555 フィルターの適用,557 フィルタリング,557 ブラウザービュー. SQL ステートメントの生成, 558 コンテキストメニューオプション,558 フラット'ファイル. マッピング,610 プラットフォーム, Altova 製品のための, 1538 ブレークポイント. について、299 削除, 302 追加, 302 ブレークポイントウィンドウ, について, 299, 309 プロジェクト, アプリケーションメニューとして, 1038 開く,129 検索, 129 作成.129 閉じる,129 ブロックコメント,553 ヘルプ. アプリケーションメニューとして、1050 マークされたアイテム, 不足しているアイテム, 124 マージ.337 XML ファイル, 337 マッピング. ソース優先 - 混在コンテンツ, 143 デバッグ.294 フラットファイルフォーマット, 610 型優先,151 検証.86 作成,80 自動化された 処理,946 マッピング 出力. として複数のファイルを生成する,161,165 マッピングコンテキスト,277 マッピングドキュメンテーション. 生成.267 マッピングメソッド. ターゲット優先,143 標準,143 標準 / 混在 / 全てコピー, 143

マッピング入力. としてカスタムファイル名を提供する,165 として複数のファイルを提供する, 161, 163, 165 メッセージ. ウィンドウ - データベース クエリ, 560 データベース クエリ内のアイコン, 560 メモリ不足エラー. トラブルシューティング,84 メモリ要件, 1538 ユーザー DSN, セットアップ, 362 ユーザー定義関数. インラインと正規, 679 コピーと張り付け,684 サンプル, 669, 685, 689 ナビゲート,681 パラメーターの追加,674 マッピングにインポート,683 概要,669 呼び出し,683 再帰的な呼び出し, 689 作成,671 削除, 682 編集,681 ライセンス, 1541 情報.1540 ライセンス計測. Altova 製品にて, 1540 ライブラリ. カスタムの追加,712 自動ロード,712 ライブラリファイル. mff, 712 ラインコメント、553 レイアウト. ブラウザー.555 レファレンス, 1031 ローカルロケーション. とストアドプロシージャ, 581, 582, 587 ローカル照合,194 ログ. データベース例外,605 ワイルドカード. xs:any - xs:anyAtrribute, 332 引用文字. CSV ファイル内, 617 階層構造. テキストファイルから, 614

外部キー. データベースマッピング内,436 データベースマッピング内の、443 関数. アプリケーションメニューとして, 1042 パラメーターの削除,644 パラメーターの追加,644 マッピングコンポーネントとして追加,639 引数データ型の確認,643 関数の説明の確認,643 複数のアイテムに条件付きで適用する,656 複数のアイテムに適用,645 含む. MapForce 関数として (core | string 関数), 822 技術データ, 1538 疑問符. 不足しているアイテム, 124 区切り文字. CSV ファイル内の変更, 617 フラットテキストファイルの変更, 627 空のフィールド. CSV ファイル内, 617 フラットテキストファイル内, 627 型優先. 接続.151 結果. ウィンドウ - データベース クエリ 560 データベース クエリ内のアイコン, 560 検索. iマッピング コンポーネント内のアイテム, 104 プロジェクトウィンドウ内のファイル, 129 検証する. デザインのマッピング,86 マッピング 出力.87 固定された. 固定長ファイル - マッピング, 610 構成. mff ファイル, 712 SQL エディター 設定, 563 混在.143 コンテンツマッピング、143 コンテンツマッピング サンプル, 148 コンテンツマッピングのメソッド,143 ソース優先 マッピング,143 作成. 領域,555 削除,151 コメント, 553

削除,151 ブックマーク,554 ブロックコメント, 553 ラインコメント, 553 削除 - 不足しているアイテム, 124 全てコピー 接続, 151 領域,555 試用期間, Altova ソフトウェア製品の試用, 1540 自動, ライブラリのロード,712 自動化された, 処理,946 主キー. データベース コンポーネントのために生成, 453 データベースマッピング内,436 データベースマッピング内の,443 データベース挿入アクション中に生成する,459 出力, アプリケーションメニューとして, 1043 プレビュー,88 検証,87 保存.88 出カパラメーター, ユーザー定義関数内,674 処理. マッピングの自動化,946 処理命令, ターゲット ファイルに追加, 329 処理命令とコメント、 マッピング,144 照合. Unicode コードポイント, 194 ロケール照合,194 並べ替えコンポーネント, 194 親コンテキスト, サンプル, 282 正規表現. ノード関数,656 マッピング内での使用,731 生成されたコード. から例外をスロー, 232 生成されたコード内のパス. 絶対パスに設定する,99 精度. ノード関数内での使用,661 接続. アプリケーションメニューとして, 1041

ルート要素の変更を保存する、119 異なるコンポーネントへ移動する,119 型優先,151 折りたたみ, 領域,555 設定. テキスト フォント - データベース クエリ, 565 結果表示 - データベース クエリ,564 選択. テーブル データ - データベース クエリ, 560 全てコピー. コネクタ, 151 とフィルター, 151 マッピングメソッド, 143 解決 / 削除 コネクタ, 151 挿入. SQL WHERE, 524 アプリケーションメニューとして、1036 コメント, 553 ブックマーク,554 ブロックコメント, 553 ラインコメント, 553 領域,555 単一のターゲット, 複数のソース,337 単純型. 並べ替え,194 値ウィンドウ, コンテキストタブ,305 について、299、305 関連したタブ,305 履歴タブ,305 著作権に関する情報, 1540 追加. カスタムライブラリ,712 定数. マッピングに追加,641 展開. 領域,555 統合. アプリケーション内の MapForce, 1377 入力の複製,56 追加. 1039 入力パラメーター, 671 ユーザー定義関数内,671,674 派生した型, マッピング, 323

背景情報, 1538

```
配布,
 Altova ソフトウェア製品, 1540
 Altova ソフトウェア製品の配布, 1540
標準,
 マッピングメソッド, 143
表示,
 アプリケーションメニューとして, 1046
評価機関,
 Altova ソフトウェア製品, 1540
不足しているアイテム, 124
複合コンテンツ,
 マッピング,150
複合型,
 並べ替え,194
複数のソース.
 単一のターゲットへ,337
文字の重点,
 フラットテキストファイル内,627
文字列,
 からデータを解析する,236
 データのシリアル化,236
 データの解析,236
 へのデータのシリアル化, 236, 238
並べ替え,
 データベース内、527
 テーブル データベース クエリ, 555
 結果ウィンドウ内のデータ,560
 結果ウィンドウ内の列アイコン, 560
 並べ替えコンポーネント, 194
並べ替えの順序,
 変更, 194
並べ替キー.
 並べ替えコンポーネント, 194
変換.
 RaptorXML Server, 945
変換言語,
 選択,85
変数,
 のスコープの変更,188
 マッピングに追加,184
 使用例, 190, 191, 192
 始めに,183
編集.
 アプリケーションメニューとして, 1035
法的な情報,1540
名前空間,
 カスタムの宣言,338
 とワイルドカード (xs:any), 332
```

1560

```
名前空間 URI,
DTD, 323
とQNames, 326
優先コンテキスト, 286
サンプル, 288
領域,
作成, 555
前りたたみ, 555
挿入, 555
展開, 555
例外,
追加, 232
列挙,
MapForceControl, 1441
```