



Informatica®
10.2.2

Web サービスガイド

本ソフトウェアおよびマニュアルは、使用および開示の制限を定めた個別の使用許諾契約のもとでのみ提供されています。本マニュアルのいかなる部分も、いかなる手段（電子的複製、写真複製、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

米政府の権利プログラム、ソフトウェア、データベース、および関連文書や技術データは、米国政府の顧客に配信され、「商用コンピュータソフトウェア」または「商業技術データ」は、該当する連邦政府の取得規制と代理店固有の補足規定に基づきます。このように、使用、複製、開示、変更、および適応は、適用される政府の契約に規定されている制限およびライセンス条項に従うものとし、政府契約の条項によって適当な範囲において、FAR 52.227-19、商用コンピュータソフトウェアライセンスの追加権利を規定します。

Informatica および Informatica ロゴは、米国およびその他の国における Informatica LLC の商標または登録商標です。Informatica の商標の最新リストは、Web (<https://www.informatica.com/trademarks.html>) にあります。その他の企業名および製品名は、それぞれの企業の商標または登録商標です。

本ソフトウェアまたはドキュメンテーション（あるいはその両方）の一部は、第三者が保有する著作権の対象となります。必要な第三者の通知は、製品に含まれています。

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、infa_documentation@informatica.com までご報告ください。

Informatica 製品は、それらが提供される契約の条件に従って保証されます。Informatica は、商品性、特定目的への適合性、非侵害性の保証等を含めて、明示的または黙示的ないかなる種類の保証をせず、本マニュアルの情報を「現状のまま」提供するものとします。

発行日: 2019-05-05

目次

| | |
|---|----|
| 序文 | 10 |
| Informatica のリソース..... | 10 |
| Informatica Network..... | 10 |
| Informatica ナレッジベース..... | 10 |
| Informatica マニュアル..... | 10 |
| Informatica 製品可用性マトリックス..... | 11 |
| Informatica Velocity..... | 11 |
| Informatica Marketplace..... | 11 |
| Informatica グローバルカスタマサポート..... | 11 |
| 第 1 章 : Web サービス | 12 |
| Web サービスの概要..... | 12 |
| REST および SOAP Web サービスの相違点..... | 13 |
| Web サービスプロセス..... | 13 |
| Web サービスコンシューマトランスフォーメーションの処理..... | 14 |
| 第 2 章 : SOAP Web サービス | 15 |
| SOAP Web サービスのコンポーネント..... | 15 |
| 操作..... | 15 |
| WSDL..... | 16 |
| SOAP..... | 16 |
| SOAP Web サービスの開発..... | 17 |
| SOAP Web サービスの例..... | 17 |
| 第 3 章 : WSDL データオブジェクト | 19 |
| WSDL データオブジェクトの概要..... | 19 |
| WSDL データオブジェクトの概要ビュー..... | 20 |
| WSDL データオブジェクトの詳細ビュー..... | 20 |
| WSDL データオブジェクトのインポート..... | 20 |
| WSDL の同期..... | 21 |
| WSDL データオブジェクトの同期..... | 21 |
| 証明書管理..... | 22 |
| Informatica Developer の証明書プロパティ..... | 22 |
| Informatica Developer への証明書の追加..... | 22 |
| 第 4 章 : スキーマオブジェクト | 23 |
| スキーマオブジェクトの概要..... | 23 |
| スキーマオブジェクトの概要ビュー..... | 23 |
| スキーマファイル..... | 24 |
| スキーマオブジェクトのスキーマビュー..... | 24 |

| | |
|---|-----------|
| 名前空間のプロパティ..... | 25 |
| 要素のプロパティ..... | 25 |
| 単純型のプロパティ..... | 27 |
| 複合型のプロパティ..... | 28 |
| 属性プロパティ..... | 29 |
| スキーマオブジェクトの詳細ビュー..... | 29 |
| スキーマオブジェクトの作成..... | 30 |
| スキーマの更新..... | 31 |
| スキーマの同期..... | 31 |
| スキーマファイルの編集..... | 32 |
| 証明書管理..... | 34 |
| Informatica Developer の証明書プロパティ..... | 35 |
| Informatica Developer への証明書の追加..... | 35 |
| 第 5 章 : SOAP Web サービスの作成方法..... | 36 |
| SOAP Web サービスの作成の概要..... | 36 |
| 型と要素..... | 37 |
| Web サービスの概要ビュー..... | 37 |
| Web サービスの WSDL ビュー..... | 39 |
| WSDL データオブジェクトからの Web サービスの作成..... | 39 |
| 手順 1.WSDL データオブジェクトからの Web サービスの作成..... | 40 |
| 手順 2.Web サービスへの操作の追加..... | 40 |
| WSDL データオブジェクトを Web サービスに関連付ける..... | 41 |
| SOAP Web サービスを手動で作成..... | 41 |
| 手順 1.Web サービスの手動作成..... | 41 |
| 手順 2.操作の作成..... | 42 |
| 手順 3.要素の作成..... | 46 |
| 手順 4.定義済みフォールトの作成..... | 47 |
| 手順 5.ヘッダーの作成..... | 47 |
| 第 6 章 : 操作マッピング..... | 48 |
| 操作マッピングの概要..... | 48 |
| 操作マッピングの [全般] タブ..... | 49 |
| 操作マッピングの [操作] タブ..... | 49 |
| 操作マッピングの [詳細] タブ..... | 49 |
| 入力トランスフォーメーション..... | 50 |
| 入力トランスフォーメーションの [ポート] タブ..... | 50 |
| 操作入力をポートにマップするためのルールとガイドライン..... | 50 |
| 入力トランスフォーメーションの設定..... | 51 |
| 出力トランスフォーメーション..... | 52 |
| 出力トランスフォーメーションの [ポート] タブ..... | 52 |
| 出力トランスフォーメーションの [詳細] タブ..... | 53 |
| ポートを操作出力にマップするためのルールとガイドライン..... | 53 |

| | |
|--|-----------|
| 出力トランスフォーメーションの設定. | 53 |
| フォールトトランスフォーメーション. | 55 |
| フォールトトランスフォーメーションの [ポート] タブ. | 56 |
| フォールトトランスフォーメーションの [詳細] タブ. | 56 |
| ポートを操作フォールトにマップするためのルールとガイドライン. | 56 |
| フォールトのトランスフォーメーションの作成. | 57 |
| フォールトトランスフォーメーションの設定. | 57 |
| フォールト処理. | 59 |
| システム定義フォールト. | 60 |
| 定義済みフォールト. | 61 |
| 汎用フォールト. | 62 |
| 操作マッピングのテスト. | 62 |
| 操作マッピングのテスト. | 62 |
| ビューをカスタマイズするためのオプション. | 63 |
| 第 7 章 : Web サービス SOAP メッセージの解析. | 64 |
| Web サービス SOAP メッセージの解析の概要. | 64 |
| トランスフォーメーションのユーザーインタフェース. | 65 |
| 複数出現出力設定. | 66 |
| 正規化したリレーショナル出力. | 66 |
| 生成キー. | 66 |
| 非正規化したリレーショナル出力. | 67 |
| ピボット化したリレーショナル出力. | 68 |
| anyType 要素の解析. | 68 |
| 派生型の解析. | 69 |
| QName 要素の解析. | 70 |
| 代替グループの解析. | 70 |
| SOAP メッセージ内の XML 構造の解析. | 70 |
| choice 要素. | 71 |
| list 要素. | 71 |
| union 要素. | 71 |
| 第 8 章 : Web サービス SOAP メッセージの生成. | 72 |
| Web サービス SOAP メッセージの生成の概要. | 72 |
| トランスフォーメーションのユーザーインタフェース. | 73 |
| [入力ポート] 領域. | 73 |
| 操作領域. | 74 |
| ポートと階層レベルのリレーション. | 74 |
| キー. | 75 |
| ポートのマップ. | 76 |
| ポートのマップ. | 77 |
| グループのマップ. | 77 |
| 複数のポートのマップ. | 78 |

| | |
|---|-----------|
| 複数出現ポートのピボット化 | 78 |
| 非正規化データのマップ | 79 |
| 派生型および要素の置き換え | 81 |
| 派生型の生成 | 81 |
| anyType 要素および属性の生成 | 81 |
| 置き換えグループの生成 | 82 |
| SOAP メッセージ内の XML 構造の生成 | 82 |
| choice 要素 | 82 |
| list 要素 | 83 |
| union 要素 | 84 |
| 第 9 章 : Web サービスコンシューマトランスフォーメーション | 85 |
| Web サービスコンシューマトランスフォーメーションの概要 | 85 |
| SOAP メッセージ | 86 |
| WSDL ファイル | 86 |
| 操作 | 87 |
| Web サービスのセキュリティ | 87 |
| WSDL の選択 | 88 |
| Web サービスコンシューマトランスフォーメーションのポート | 89 |
| HTTP ヘッダー入力ポート | 90 |
| その他の入力ポート | 90 |
| Web サービスコンシューマトランスフォーメーションの入力マッピング | 91 |
| 入力ポートをノードにマップするためのルールとガイドライン | 92 |
| [ビューのカスタマイズ] のオプション | 92 |
| 操作入力への入力ポートのマッピング | 92 |
| Web サービスコンシューマトランスフォーメーションの出力マッピング | 94 |
| ノードを出力ポートにマップするためのルールとガイドライン | 95 |
| SOAP メッセージを XML としてマップ | 95 |
| [ビューのカスタマイズ] のオプション | 95 |
| 出力ポートへの操作出力のマッピング | 96 |
| Web サービスコンシューマトランスフォーメーションの詳細プロパティ | 97 |
| Web サービスのエラー処理 | 99 |
| メッセージの圧縮 | 100 |
| 並行処理 | 100 |
| フィルタの最適化 | 101 |
| Web サービスコンシューマトランスフォーメーションでの初期選択の最適化の有効化 | 101 |
| Web サービスコンシューマトランスフォーメーションによるプッシュイン最適化 | 102 |
| Web サービスコンシューマトランスフォーメーションの作成 | 103 |
| Web サービスコンシューマトランスフォーメーションの例 | 105 |
| 入力ファイル | 105 |
| 論理データオブジェクトモデル | 105 |
| 論理データオブジェクトマッピング | 106 |
| Web サービスコンシューマトランスフォーメーション | 107 |

| | |
|---|------------|
| 第 10 章 : REST Web サービス | 109 |
| REST Web サービスの概要 | 109 |
| REST Web サービスの処理 | 110 |
| Web サービスコンシューマトランスフォーメーションの処理 | 110 |
| REST Web サービスリソース | 111 |
| REST Web サービスのスキーマビュー | 113 |
| データオブジェクトの同期 | 113 |
| リソースキー | 113 |
| リソースマッピング | 114 |
| デフォルトリソースマッピング | 114 |
| カスタムリソースマッピング | 115 |
| REST Web サービスの出力トランスフォーメーション | 116 |
| REST 出力トランスフォーメーションの複数出現データ | 117 |
| 要求メッセージ | 118 |
| リソースマッピングでのデータのフィルタリング | 118 |
| キーで検索 | 119 |
| 応答メッセージ形式 | 120 |
| 応答のデータレプレビュー | 121 |
| 第 11 章 : REST Web サービスの作成方法 | 122 |
| REST Web サービスの作成 | 122 |
| REST Web サービスを手動で作成する方法 | 123 |
| REST Web サービスの例 | 123 |
| 手順 1. REST Web サービスリソースの作成 | 123 |
| REST Web サービスリソースの作成 | 123 |
| 手順 2. リソースマッピングの定義 | 125 |
| リソースマッピングの定義 | 127 |
| 手順 3. 出力マッピングの設定 | 130 |
| 出力マッピングの設定 | 131 |
| 手順 4. [データビューア] ビューでのマッピングのテスト | 131 |
| リソース ID による出力のフィルタリング | 132 |
| フィルタ条件による出力のフィルタリング | 132 |
| 手順 5. アプリケーションのデプロイ | 133 |
| アプリケーションのデプロイ | 134 |
| 手順 6. ブラウザからの Web サービスのクエリ | 135 |
| Web サービスのクエリ | 136 |
| データオブジェクトから REST Web サービスを作成する方法 | 137 |
| データオブジェクトを REST Web サービスとしてデプロイする方法 | 140 |
| 第 12 章 : REST Web サービスコンシューマトランスフォーメーション | 144 |
| REST Web サービスコンシューマトランスフォーメーションの概要 | 144 |

| | |
|--|-----|
| REST Web サービスコンシューマトランスフォーメーションの処理. | 146 |
| REST Web サービスコンシューマトランスフォーメーションの設定. | 146 |
| メッセージの設定. | 146 |
| リソースの識別. | 147 |
| HTTP メソッド. | 148 |
| HTTP Get メソッド. | 148 |
| HTTP Post メソッド. | 149 |
| HTTP Put メソッド. | 149 |
| HTTP Delete メソッド. | 150 |
| REST Web サービスコンシューマトランスフォーメーションのポート. | 151 |
| 入力ポート. | 151 |
| 出力ポート. | 151 |
| パススルーポート. | 151 |
| 引数ポート. | 152 |
| URL ポート. | 152 |
| HTTP ヘッダポート. | 152 |
| クッキーポート. | 153 |
| 出力 XML ポート. | 153 |
| 応答コードポート. | 153 |
| REST Web サービスコンシューマトランスフォーメーションの入力マッピング. | 153 |
| 入力ポートを要素にマップするためのルールとガイドライン. | 154 |
| メソッド入力への入力ポートのマッピング. | 154 |
| REST Web サービスコンシューマトランスフォーメーションの出力マッピング. | 156 |
| 要素を出力ポートにマップするためのルールとガイドライン. | 156 |
| ビューのカスタマイズのオプション. | 157 |
| 出力ポートへのメソッド出力のマッピング. | 157 |
| REST Web サービスコンシューマトランスフォーメーションの詳細プロパティ. | 158 |
| REST Web サービスコンシューマトランスフォーメーションの作成. | 159 |
| REST Web サービスコンシューマトランスフォーメーションの作成. | 159 |
| 配列を含む JSON 応答メッセージの解析. | 160 |
| JSON 応答メッセージの例. | 160 |
| 応答メッセージ内の名前なし配列. | 160 |

第 13 章 : REST Web サービスコンシューマトランスフォーメーションの使用例. 162

| | |
|--|-----|
| データを受信できるように REST Web サービスコンシューマトランスフォーメーションを設定する 方法. | 162 |
| Web サービスから未解析のデータを受信する方法. | 163 |
| Web サービスから解析されたデータを受信する方法. | 165 |
| スキーマを使用して REST Web サービスコンシューマトランスフォーメーションを作成する方法. | 169 |
| XML フォーマットで要求メッセージおよび応答メッセージを処理する方法. | 169 |
| JSON フォーマットで解析メッセージおよび応答メッセージを扱う方法. | 176 |
| REST Web サービスコンシューマトランスフォーメーションでカスタム URL を生成する方法. | 180 |

| | |
|--|------------|
| URL ポートを使用して動的 URL を構築する方法. | 180 |
| 引数ポートを使用して URL パラメータを生成する方法. | 181 |
| 第 14 章 : REST および SOAP Web サービスの管理. | 183 |
| Web サービスの管理の概要. | 183 |
| Web サービスのプロパティの構成. | 184 |
| Web サービスのプロパティ. | 184 |
| Web サービスの操作とリソースのプロパティ. | 186 |
| Web サービスの結果セットキャッシュ処理. | 186 |
| Web サービスのセキュリティ管理. | 188 |
| Web サービスの権限. | 188 |
| SOAP リクエスト内のユーザー名トークン. | 190 |
| グリッド上の Web サービス. | 192 |
| サービスプロセスでジョブを実行するグリッドの設定. | 192 |
| Web サービスのログ. | 193 |
| Web サービスのトレースレベル. | 193 |
| Web サービスの監視. | 193 |
| Web サービスの [プロパティ] ビュー. | 194 |
| Web サービスの [レポート] ビュー. | 194 |
| REST または SOAP Web サービスの [操作] ビュー. | 194 |
| Web サービスの [要求] ビュー. | 195 |
| 付録 A : データ型の互換性. | 196 |
| データ型リファレンスの概要. | 196 |
| XML データ型とトランスフォーメーションデータ型. | 197 |
| Decimal. | 199 |
| 索引. | 200 |

序文

『Web サービスガイド』は、データ品質とデータサービスの開発者を対象としています。このガイドは、Web サービスの概念について理解していることを前提としています。

Informatica のリソース

Informatica は、Informatica Network やその他のオンラインポータルを通じてさまざまな製品リソースを提供しています。リソースを使用して Informatica 製品とソリューションを最大限に活用し、その他の Informatica ユーザーや各分野の専門家から知見を得ることができます。

Informatica Network

Informatica Network は、Informatica ナレッジベースや Informatica グローバルカスタマサポートなど、多くのリソースへの入口です。Informatica Network を利用するには、<https://network.informatica.com> にアクセスしてください。

Informatica Network メンバーは、次のオプションを利用できます。

- ナレッジベースで製品リソースを検索できます。
- 製品の提供情報を表示できます。
- サポートケースを作成して確認できます。
- 最寄りの Informatica ユーザーグループネットワークを検索して、他のユーザーと共同作業を行えます。

Informatica ナレッジベース

Informatica ナレッジベースを使用して、ハウツー記事、ベストプラクティス、よくある質問に対する回答など、製品リソースを見つけることができます。

ナレッジベースを検索するには、<https://search.informatica.com> にアクセスしてください。ナレッジベースに関する質問、コメント、ご意見の連絡先は、Informatica ナレッジベースチーム (KB_Feedback@informatica.com) です。

Informatica マニュアル

Informatica マニュアルポータルでは、最新および最近の製品リリースに関するドキュメントの膨大なライブラリを参照できます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

Informatica はマニュアルポータルに加えて、Informatica ナレッジベースでも多くの製品に関するドキュメントを管理しています。マニュアルポータルで製品または製品バージョンに関するマニュアルが見つからない場合は、<https://search.informatica.com> でナレッジベースを検索してください。

製品マニュアルに関する質問、コメント、ご意見については、Informatica マニュアルチーム (infa_documentation@informatica.com) までご連絡ください。

Informatica 製品可用性マトリックス

製品可用性マトリックス (PAM) には、製品リリースでサポートされるオペレーティングシステム、データベースなどのデータソースおよびターゲットが示されています。Informatica PAM は、<https://network.informatica.com/community/informatica-network/product-availability-matrices> で参照できます。

Informatica Velocity

Informatica Velocity は、Informatica プロフェッショナルサービスによって開発された、数百に及ぶデータ管理プロジェクトでの実世界の経験に基づくヒントやベストプラクティスをまとめたものです。世界中の組織と協力して優れたデータ管理ソリューションの計画、開発、展開、および維持を行ってきた弊社コンサルタントの知識の集合体と言えます。

Informatica Velocity のリソースは、<http://velocity.informatica.com> でご覧ください。Informatica Velocity についての質問、コメント、またはアイデアがある場合は、ips@informatica.com から Informatica プロフェッショナルサービスにお問い合わせください。

Informatica Marketplace

Informatica Marketplace は、お使いの Informatica 製品を拡張したり強化したりするソリューションを検索できるフォーラムです。Marketplace で、Informatica デベロッパーやパートナーからの多数のソリューションを活用すれば、生産性を向上したり、プロジェクトでの実装時間を短縮したりできます。Informatica Marketplace は、<https://marketplace.informatica.com> からアクセスしてください。

Informatica グローバルカスタマサポート

電話または Informatica Network からグローバルサポートセンターに連絡できます。

各地域の Informatica グローバルカスタマサポートの電話番号は、Informatica Web サイト (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>) を参照してください。

Informatica Network でオンラインサポートリソースを見つけるには、<https://network.informatica.com> にアクセスし、eSupport オプションを選択します。

第 1 章

Web サービス

この章では、以下の項目について説明します。

- [Web サービスの概要, 12 ページ](#)
- [REST および SOAP Web サービスの相違点, 13 ページ](#)
- [Web サービスプロセス, 13 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの処理, 14 ページ](#)

Web サービスの概要

Web サービスクライアントは、Informatica Web サービスに接続してデータのアクセス、変換、または配信を行うことができます。外部アプリケーションや Web サービスコンシューマトランスフォーメーションは、Web サービスクライアントとして Web サービスに接続できます。Informatica Web サービスは、Developer tool で作成できます。

Web サービスでは、情報の要求、データ更新の要求、タスク実行の要求を実行できます。例えば、Web サービス操作を実行する要求を Web サービスクライアントが送信するとします。Web サービスクライアントは、要求で顧客 ID を渡します。Web サービスは顧客および注文情報を取得し、その情報を応答でクライアントに返します。

Informatica Web サービスは、Simple Object Access Protocol (SOAP) または Representational State Transfer (REST) メッセージングプールを使用して Web サービスクライアントと通信します。

以下のタイプの Web サービスまたは Web サービスクライアントを Developer tool で作成できます。

SOAP Web サービス

SOAP プロトコルを使用する Web サービス。Web サービスクライアントの要求および Web サービスの応答は、SOAP メッセージです。Web サービス記述言語 (WSDL) とは、Web サービスの機能を記述する XML ベースのインタフェース定義言語です。WSDL ファイルには、Web サービスの呼び出し方法、Web サービスで期待されるパラメータ、Web サービスが返すデータ構造の記述が含まれます。WSDL ファイルから Informatica SOAP Web サービスを作成できます。

SOAP Web サービスコンシューマトランスフォーメーション

Web サービスクライアントとして Web サービスに接続し、マッピングのデータ中間ストリームのアクセスまたは変換を行います。WSDL から SOAP Web サービスコンシューマトランスフォーメーションを作成できます。

REST Web サービス

HTTP 要求を受信して Web サービス操作を実行する Web サービス。Informatica REST Web サービスは HTTP 要求を受信して GET 操作を実行します。Informatica REST Web サービスは、JSON ファイルまたは XML ファイルで応答を返すことができます。

REST コンシューマトランスフォーメーション

Web サービスクライアントとして REST Web サービスに接続し、マッピングのデータ中間ストリームのアクセスまたは変換を行います。REST Web サービスコンシューマトランスフォーメーションは、トランスフォーメーション、HTTP 接続、または HTTPS 接続で定義する URL を通じて、Web サービスに接続します。要求および応答メッセージには XML または JSON データが含まれます。

REST および SOAP Web サービスの相違点

Informatica Developer tool で REST または SOAP Web サービスを作成できます。

REST および SOAP Web サービスには次の違いがあります。

要求メッセージ形式

SOAP メッセージは構造化された XML です。SOAP Web サービスは XML を解析して Web サービスが実行する必要のある処理を決定します。REST 要求はクエリを含む単純な URI 文字列です。

応答メッセージ形式

SOAP Web サービスは WSDL で定義された XML 形式の応答を返します。

Informatica REST Web サービスは JavaScript Object Notation (JSON) または XML 応答メッセージを返します。応答メッセージ形式は WSDL やスキーマでは定義されません。Informatica REST Web サービスを定義する場合は、出力形式を定義します。

Web サービスマッピング形式

Informatica SOAP Web サービスには、操作マッピングが含まれています。SOAP 操作マッピングには、要求メッセージの XML を解析する入力トランスフォーメーションが含まれています。クライアント要求によって要求されたデータを処理するトランスフォーメーションを Web サービスマッピングに追加する必要があります。

Informatica REST Web サービスには、リソースマッピングが含まれています。このリソースマッピングは、要求クエリを読み取りません。REST リソースマッピングには、入力トランスフォーメーションではなく、読み取りトランスフォーメーションが含まれています。読み取りトランスフォーメーションでは、モデルリポジトリのデータオブジェクトが読み取られ、データが取得されてクライアントに返されます。デフォルトでは、フィルタトランスフォーメーションやルックアップトランスフォーメーションを追加して、クライアントクエリに基づいてデータを取得する必要はありません。マッピングからデータが返ってきたら、出力データが REST Web サービスによってフィルタリングされます。

Web サービスプロセス

Web サービスは、Web サービスクライアントから要求を受け取ります。

以下のプロセスは、Web サービスクライアントからの Web サービス要求が、データ統合サービスでどのように処理されるかを示しています。

1. データ統合サービスは、Web サービスクライアントから要求を受け取ります。
2. マッピングの実行により、データ統合サービスの Web サービスモジュールまたはデータ統合サービスの REST Web サービスモジュールによって要求が処理されます。
3. Web サービスモジュールまたは REST Web サービスモジュールによって応答が Web サービスクライアントが送信されます。

Web サービスコンシューマトランスフォーメーションの処理

外部アプリケーションや Web サービスコンシューマトランスフォーメーションは、Web サービスクライアントとして Web サービスに接続できます。

以下のプロセスは、Web サービスコンシューマトランスフォーメーションがどのように要求を送信し、Web サービスからの応答を受信するかを示しています。

1. Web サービスコンシューマトランスフォーメーションが要求を生成し、接続オブジェクトがある Web サービスに接続します。
2. Web サービスコンシューマトランスフォーメーションが Web サービスから応答を受信します。
3. Web サービスコンシューマトランスフォーメーションが応答からデータを抽出し、トランスフォーメーション出力ポートでデータを返します。

第 2 章

SOAP Web サービス

この章では、以下の項目について説明します。

- [SOAP Web サービスのコンポーネント, 15 ページ](#)
- [SOAP Web サービスの開発, 17 ページ](#)
- [SOAP Web サービスの例, 17 ページ](#)

SOAP Web サービスのコンポーネント

SOAP Web サービスのコンポーネントは、Web サービスの目的および Web サービスクライアントと Web サービスの通信方法を定義します。

Web サービスには、以下のコンポーネントがあります。

操作

Web サービスには 1 つ以上の操作を組み込むことができます。各操作は、Web サービスでのアクションに対応します。

Web サービス記述言語 (WSDL)

WSDL は、Web サービス操作のプロトコル、形式、およびシグネチャを記述する XML ドキュメントです。

Simple Object Access Protocol (SOAP)

SOAP は、Web サービス用の通信プロトコルです。

操作

Web サービスには、その Web サービスがサポートする各アクションの操作が含まれています。

例えば、顧客名を受信して顧客の詳細を応答で返す `getcustomerid` という操作を Web サービスに含めることができます。操作入力には、顧客名の要素が含まれています。操作出力には、顧客名に基づいた顧客詳細の要素が含まれています。

Developer ツールで操作を定義する際に、操作のコンポーネントを定義します。操作は以下のコンポーネントで構成されます。

操作入力と操作出力

操作入力では、操作の SOAP リクエスト内の要素を定義します。操作出力では、操作の SOAP レスポンス内の要素を定義します。

操作入力と操作出力には、ヘッダーを入れることができます。ヘッダーでは、SOAP メッセージ内のデータが送受信されます。ヘッダーでは、SOAP リクエストまたは SOAP レスポンスのヘッダー内の要素を定義します。

操作フォールト

操作フォールトでは、操作の結果として出力される可能性があるエラーメッセージのメッセージ形式を定義します。1つの操作に複数の操作フォールトを定義できます。

各操作には、操作マッピングを設定する必要があります。操作入力、操作出力、および各操作フォールトは、操作マッピングのトランスフォーメーションに対応します。

WSDL

WSDL は、Web サービス操作のプロトコル、形式、およびシグネチャを記述する XML スキーマです。

WSDL には、サービス要求の送信側と受信側の両方が交換されるデータを認識できるように、Web サービスに渡されるデータが記述されています。WSDL の要素には、メッセージの受信側がデータの処理方法を認識できるように、データ上で実行される操作が記述されます。また、メッセージの送信側が送信方法を認識できるように、プロトコルまたはトランスポートのバインドが記述されます。

Web サービスの WSDL は、Developer ツールまたは Administrator ツールで表示できます。Web サービスを Data Integration Service にデプロイした後に、WSDL URL を表示したり、WSDL をファイルにダウンロードしたりすることができます。Administrator ツールで表示された WSDL URL にアクセスすると、WSDL のコンテンツを見ることができます。

SOAP

SOAP は、Web サービス用の通信プロトコルです。SOAP では、Web サービスの要求、応答、およびフォールトメッセージの形式を定義します。Data Integration Service では、ドキュメント/リテラルのエンコーディングを使用する SOAP 1.1 メッセージおよび SOAP 1.2 メッセージを処理できます。

SOAP メッセージは次のセクションを含んでいます。

SOAP エンベロープ

エンベロープでは、メッセージのフレームワーク、メッセージのコンテンツ、およびメッセージを処理する対象を定義します。

SOAP ヘッダー

ヘッダーでは、SOAP メッセージを送信したエンティティが識別されます。ヘッダーには認証情報が含まれています。また、SOAP メッセージの処理方法に関する情報も含まれています。

SOAP ボディ

ボディは、クライアントと Web サービスプロバイダが互いに受け渡すデータのコンテナです。

SOAP メッセージは XML です。SOAP メッセージに複数出現要素が含まれている場合は、要素のグループによって XML 階層のレベルが形成されます。あるレベルが別のレベルにネストされている場合、グループは関連しています。

SOAP リクエストメッセージには、階層データを含めることができます。例えば、販売データベースに顧客の注文を追加する要求をクライアントが送信するとします。クライアントは、2つのデータグループを SOAP リクエストメッセージで渡します。一方のグループには顧客の ID と名前が含まれ、もう一方のグループには注文情報が含まれています。注文情報は複数回出現します。

SOAP レスポンスメッセージには、階層データを含めることができます。例えば、Web サービスクライアントが顧客注文に対する SOAP リクエストを生成するとします。Web サービスは、注文ヘッダーおよび複数出現する注文詳細の要素を、SOAP レスポンスで返します。

SOAP Web サービスの開発

Web サービスクライアントが操作を実行するために使用するインタフェースを提供するには、SOAP Web サービスを開発します。Web サービスクライアントとなるのは、外部 Web サービスクライアントまたは Web サービスコンシューマトランスフォーメーションです。例えば、Web サービスクライアントは、Web サービスに接続して、顧客名または顧客 ID を基に顧客の詳細を表示できます。

Web サービスを開発するには、以下の手順を実行します。

1. Web サービスを作成します。
 - WSDL データオブジェクトから Web サービスを作成します。WSDL データオブジェクトを作成するには、WSDL ファイルをインポートします。WSDL ファイルにより、Web サービスの操作出力、操作入力、および操作フォールトが定義されます。
 - 手動で Web サービスを作成します。操作入力、操作出力、および操作フォールトを設定します。スキーマオブジェクトの要素および型を使用して、操作のコンポーネント定義することができます。操作の操作入力および操作出力の要素を定義する際に、再利用可能なマプレット、再利用可能なトランスフォーメーション、および再利用可能な論理データオブジェクトを使用することができます。
2. 操作マッピングを設定します。

データ統合サービスでの、SOAP メッセージと入力トランスフォーメーションポートおよび出力トランスフォーメーションポート間のデータの抽出方法を設定します。さらに、操作マッピングロジックを設定し、各操作マッピングをテストします。
3. Web サービスをデータ統合サービスにデプロイします。

Web サービスをアプリケーションに追加し、そのアプリケーションをデータ統合サービスにデプロイします。データ統合サービスで既に稼働している Web サービスを含んだアプリケーションをデプロイすると、データ統合サービスによって Web サービスのサービス名に番号が付加されます。
4. Web サービスに対して管理タスクを実行します。

Web サービスのプロパティとセキュリティを、Administrator ツールで設定します。

これで、SOAP Web サービスクライアントはデータ統合サービスで稼働中の SOAP Web サービスに接続できます。Web サービスクライアントは、WSDL のコンテンツを使用して Web サービスに接続します。Web サービス接続オブジェクトを使用して Web サービスに接続するように、Web サービスコンシューマトランスフォーメーションを設定することができます。

SOAP Web サービスの例

顧客データにアクセスする Web サービスを作成する場合や、顧客のアドレスデータを検証する Web サービスを作成場合があります。

顧客データへのアクセス

Hypostores の顧客サービス担当者は、ロサンゼルスとボストンにあるオフィスの顧客データに、ネットワークを介してアクセスしたいと考えています。担当者らは、顧客名または顧客 ID を基に顧客の詳細を表示できるようにすることを求めています。会社の方針により、ネットワークを介してアクセスされるデータは、セキュリティで保護する必要があります。

開発者と管理者は、顧客サービスが必要とするデータへのアクセスを実現するために、以下の手順を実行します。

1. Developer tool で、開発者が以下の操作が組み込まれた Web サービスを作成します。
 - `getCustomerDetailsByName`
操作入力には、顧客名の要素が含まれています。操作出力には、顧客名に基づいた顧客詳細の要素が含まれています。
 - `getCustomerDetailsById`
操作入力には、顧客 ID の要素が含まれています。操作出力には、顧客 ID に基づいた顧客詳細の要素が含まれています。
2. 開発者は、以下のコンポーネントを使用して各操作の操作マッピングを設定します。
 - 入力トランスフォーメーションと出力トランスフォーメーション。
 - ロサンゼルスおよびボストンのオフィスから得た顧客データの単一ビューを定義する論理データオブジェクトに対してルックアップを実行する、ルックアップトランスフォーメーション。
3. 開発者は、Web サービスをデータ統合サービスにデプロイします。
4. Administrator ツールで、管理者がトランスポートレイヤセキュリティおよびメッセージレイヤセキュリティを使用するように Web サービスを設定します。これにより、Web サービスは HTTPS URL を使用して許可された要求を受信できるようになります。
5. 管理者は、顧客サービスに WSDL URL を送信して、顧客サービスが Web サービスに接続できるようにします。

顧客のアドレスデータの検証

Hypostores の受注処理部門は、受注を完了する前にアドレスデータを検証したいと考えています。アドレスバリデータトランスフォーメーションでは、入力アドレスデータをアドレス参照データと比較して入力データの精度を確認し、それらのアドレスの誤りを修正します。

開発者と管理者は、受注処理部門がアドレス検証機能を利用できるようにするために、以下の手順を実行します。

1. Developer tool で、開発者が、アドレスデータを入力として受け取って検証済みのアドレスデータを出力として返すアドレスバリデータトランスフォーメーションを含むマップレットを作成します。
2. 開発者は Web サービスを作成し、マップレットを使用して Web サービス操作を作成します。
Web サービスの作成ウィザードを使用して、再利用可能なオブジェクトから操作を作成できます。
3. 開発者は、Web サービスをデータ統合サービスにデプロイします。
4. 管理者は、受注処理部門に WSDL URL を送信して、受注処理部門が Web サービスに接続できるようにします。

Web サービスがアドレスを入力として受け入れ、検証済みのアドレスを出力として返します。

第 3 章

WSDL データオブジェクト

この章では、以下の項目について説明します。

- [WSDL データオブジェクトの概要, 19 ページ](#)
- [WSDL データオブジェクトの概要ビュー, 20 ページ](#)
- [WSDL データオブジェクトの詳細ビュー, 20 ページ](#)
- [WSDL データオブジェクトのインポート, 20 ページ](#)
- [WSDL の同期, 21 ページ](#)
- [証明書管理, 22 ページ](#)

WSDL データオブジェクトの概要

WSDL データオブジェクトは、WSDL ファイルをソースとして使用する物理データオブジェクトです。WSDL データオブジェクトを使用して、Web サービスまたは Web サービスコンシューマトランスフォーメーションを作成できます。WSDL データオブジェクトを作成するには、WSDL ファイルをインポートします。

WSDL データオブジェクトをインポートした後に、**概要ビュー**と**詳細ビュー**で、全般プロパティと詳細プロパティを編集できます。**WSDL** ビューには、WSDL ファイルのコンテンツが表示されます。

WSDL をインポートする際には、以下のガイドラインを考慮してください。

- WSDL ファイルは WSDL 1.1 に準拠している必要があります。
- WSDL ファイルは有効でなければなりません。
- Web サービスまたは Web サービスコンシューマトランスフォーメーションに組み込む操作が、ドキュメント/リテラルのエンコーディングを使用している必要があります。WSDL ファイル内のすべての操作がドキュメント/リテラル以外のエンコーディングタイプを使用している場合、WSDL のインポートは失敗します。
- WSDL ファイルで参照されているすべてのスキーマに Developer ツールからアクセスできる必要があります。
- WSDL ファイルにスキーマが含まれている場合または外部スキーマがある場合は、Developer ツールで WSDL データオブジェクト内に埋め込みスキーマが作成されます。
- WSDL ファイルに別の WSDL ファイルをインポートすると、Developer ツールによって両方の WSDL が結合されて、WSDL データオブジェクトが作成されます。
- WSDL ファイルに複数の操作が定義されている場合は、Developer ツールによってすべての操作が WSDL データオブジェクトに組み込まれます。WSDL データオブジェクトから Web サービスを作成する場合は、1 つ以上の操作を組み込むことができます。

WSDL データオブジェクトの概要ビュー

WSDL データオブジェクトの概要ビューには、WSDL および WSDL 内の操作についての全般的な情報が表示されます。

以下の表に、WSDL データオブジェクトに対して設定する全般プロパティを示します。

| プロパティ | 説明 |
|-------|--------------------|
| 名前 | WSDL データオブジェクトの名前。 |
| 説明 | WSDL データオブジェクトの説明。 |

以下の表に、WSDL データオブジェクトで定義される操作のカラムを示します。

| プロパティ | 説明 |
|-------|-----------------------------------|
| 操作 | WSDL で操作のメッセージ形式とプロトコルが定義されている場所。 |
| 入力 | 操作入力に関連付けられた WSDL メッセージ名。 |
| 出力 | 操作出力に関連付けられた WSDL メッセージ名。 |
| フォールト | 操作フォールトに関連付けられた WSDL メッセージ名。 |

WSDL データオブジェクトの詳細ビュー

WSDL データオブジェクトの詳細ビューには、WSDL データオブジェクトの詳細プロパティが表示されます。

以下の表に、WSDL データオブジェクトの詳細プロパティを示します。

| プロパティ | 説明 |
|------------|--|
| Connection | Web サービスコンシューマトランスフォーメーション用のデフォルトの Web サービス接続。 |
| ファイルの場所 | WSDL ファイルが存在する場所。 |

WSDL データオブジェクトのインポート

WSDL データオブジェクトは、WSDL ファイルから、または WSDL の場所を指す URI からインポートできます。WSDL データオブジェクトは、SOAP 1.1 バインディング操作と SOAP 1.2DL バインディング操作またはそのいずれかを含む WSDL ファイルからインポートできます。

1. **【ファイル】** > **【新規】** > **【データオブジェクト】** をクリックします。

2. **【WSDL データオブジェクト】** を選択し、**【次へ】** をクリックします。
【新しい WSDL データオブジェクト】 ダイアログボックスが表示されます。
3. **【WSDL】** オプションの横にある **【参照】** をクリックして、WSDL の場所を入力します。次に、**【OK】** をクリックします。

WSDL の場所を入力する際に、WSDL ファイルを参照するか、または WSDL の URI を入力できます。
注: URI に英語以外の文字が含まれていると、インポートが失敗する場合があります。任意のブラウザのアドレスバーに URI をコピーします。ブラウザから場所をコピーします。Developer ツールは、エンコードされた URI をブラウザから受け付けます。
4. WSDL の名前を入力します。
5. **【場所】** オプションの横にある **【参照】** をクリックして、WSDL データオブジェクトをインポートするプロジェクトまたはフォルダの場所を選択します。
6. **【次へ】** をクリックして、WSDL 内の操作を表示します。
7. **【完了】** をクリックします。
Object Explorer ビューで、プロジェクトまたはフォルダー内の **【物理データオブジェクト】** の下にデータオブジェクトが表示されます。

WSDL の同期

WSDL ファイルを変更したときに WSDL データオブジェクトを同期できます。WSDL データオブジェクトの同期化では、WSDL ファイルからオブジェクトメタデータが再インポートされます。

WSDL データオブジェクトを使用して、Web サービスまたは Web サービスコンシューマトランスフォーメーションを作成できます。WSDL データオブジェクトを更新すると、その WSDL を参照するオブジェクトが更新され、オブジェクトを開くと更新済みとしてマークされています。Developer ツールが新旧の WSDL を比較するときは、名前属性によって WSDL コンポーネントを識別します。

名前属性が変更されていない場合、WSDL コンポーネントを参照するオブジェクトが更新されます。たとえば、WSDL ファイルを編集して、単純要素 "CustID" の型を xs:string から xs:integer に変更します。

名前属性が変更された場合、その WSDL コンポーネントを参照するオブジェクトを開くと変更済みとしてマークされています。

Developer ツールによって、WSDL データオブジェクトを更新する前に WSDL ファイルが検証されます。WSDL ファイルにエラーがある場合、Developer ツールはファイルをインポートしません。

WSDL データオブジェクトの同期

WSDL ファイルを変更したときに WSDL データオブジェクトを同期します。

1. **【Object Explorer】** ビューの WSDL データオブジェクトを右クリックして、**【同期】** を選択します。
【WSDL データオブジェクトの同期化】 ダイアログボックスが表示されます。
2. **【WSDL】** フィールドの横にある **【参照】** をクリックして、WSDL の場所を入力します。次に、**【OK】** をクリックします。

WSDL の場所を入力する際に、WSDL ファイルを参照するか、または WSDL の URI を入力できます。

注: URI に英語以外の文字が含まれていると、インポートが失敗する場合があります。任意のブラウザのアドレスバーに URI をコピーします。ブラウザから場所をコピーします。Developer ツールは、エンコードされた URI をブラウザから受け付けます。

3. WSDL の名前と場所を確認します。
4. **【次へ】** をクリックして、WSDL 内の操作を表示します。
5. **【完了】** をクリックします。

WSDL を参照するオブジェクトが更新され、オブジェクトを開くと更新済みとマークされています。

証明書管理

Developer ツールでは、クライアント認証を必要とする URL から WSDL データオブジェクトとスキーマオブジェクトをインポートするために、証明書を使用する必要があります。

クライアント認証を必要とする URL をホストするサーバーが信頼されている証明書を使用している場合は、デフォルトで Developer ツールがその URL からオブジェクトをインポートします。この URL をホストするサーバーが信頼されていない証明書を使用している場合は、その証明書を Developer ツールに追加する必要があります。信頼されていない証明書を Developer ツールに追加しないと、Developer ツールではオブジェクトをインポートできません。オブジェクトのインポート元となる URL のサーバー管理者に、証明書ファイルとパスワードを要求してください。

Developer ツールに追加した証明書は、Developer ツールマシン上で実行するインポートに適用されます。Developer ツールでは、証明書がモデルリポジトリに保管されません。

Informatica Developer の証明書プロパティ

クライアント認証を必要とする URL から、信頼されていない証明書を使用してオブジェクトをインポートする場合は、Developer ツールに証明書を追加します。

以下の表に、証明書のプロパティを示します。

| プロパティ | 説明 |
|------------|----------------------|
| ホスト名 | URL をホストするサーバーの名前。 |
| ポート番号 | URL のポート番号。 |
| 証明書ファイルのパス | クライアント証明書ファイルの場所。 |
| パスワード | クライアント証明書ファイルのパスワード。 |

Informatica Developer への証明書の追加

証明書を追加する場合は、証明書プロパティを設定します。これは、クライアント認証を必要とする URL から信頼されていない証明書を使用してオブジェクトをインポートする場合に、Developer ツールで 사용되는ものです。

1. **【ウィンドウ】** > **【設定】** をクリックします。
2. **【Informatica】** > **【Web サービス】** > **【証明書】** を選択します。
3. **【追加】** をクリックします。
4. 証明書プロパティを設定します。
5. **【OK】** をクリックします。

第 4 章

スキーマオブジェクト

この章では、以下の項目について説明します。

- [スキーマオブジェクトの概要, 23 ページ](#)
- [スキーマオブジェクトの概要ビュー, 23 ページ](#)
- [スキーマオブジェクトのスキーマビュー, 24 ページ](#)
- [スキーマオブジェクトの詳細ビュー, 29 ページ](#)
- [スキーマオブジェクトの作成, 30 ページ](#)
- [スキーマの更新, 31 ページ](#)
- [証明書管理, 34 ページ](#)

スキーマオブジェクトの概要

スキーマオブジェクトは、モデルリポジトリにインポートする階層スキーマです。スキーマをインポートした後に、Developer tool でスキーマコンポーネントを確認することができます。Avro、Parquet、XML、または JSON スキーマをインポートできます。Developer tool によって、スキーマがモデルリポジトリの.xsd ファイルに変換されます。

SOAP Web サービスを作成する際に、階層スキーマに基づいて Web サービスの構造を定義することができます。WSDL を使用せずに Web サービスを作成する場合は、スキーマで定義されている型と要素に基づいて、操作、入力、出力、およびフォールトのシグネチャを定義できます。

スキーマをインポートする場合は、**概要ビュー**でスキーマの全般的なプロパティを編集できます。**詳細ビュー**では詳細プロパティを編集できます。**スキーマビュー**では、スキーマファイルのコンテンツを確認できます。

スキーマオブジェクトの概要ビュー

スキーマ名またはスキーマの説明の更新、名前空間の表示、またはスキーマファイル名前空間を表示するには、**概要ビュー**を選択します。

概要ビューには、スキーマの名前、説明、ターゲット名前空間が表示されます。スキーマ前および説明を編集できます。ターゲット名前空間は、スキーマコンポーネントが属する名前空間を表示します。ターゲット名前空間が表示されない場合は、スキーマコンポーネントが名前空間に属していません。

【スキーマの場所】領域にはスキーマファイルと名前空間のリストを表示します。複数のルート.xsd ファイルを追加できます。スキーマファイルに他のスキーマファイルが含まれる場合、または他のスキーマファイルをインポートする場合、Developer ツールによって子.xsd ファイルがスキーマに含まれます。

スキーマファイル

複数のルートレベルの.xsd ファイルをスキーマオブジェクトに追加できます。また、スキーマオブジェクトからルートレベルの.xsd ファイルを削除することもできます。

スキーマファイルを追加すると、追加したファイルによってインポートされたすべての.xsd ファイルがファイルがインポートされるか、追加したファイルに含まれているすべての.xsd ファイルがインポートされます。Developer ツールは追加するファイルを、スキーマオブジェクトの一部であるファイルに対して検証します。Developer ツールでは、スキーマオブジェクトの一部であるファイルと競合するファイルは追加できません。

たとえば、ルートスキーマファイル"BostonCust.xsd"を含むスキーマオブジェクトがあり、そのスキーマオブジェクトにルートスキーマ"LACust.xsd"を追加するとします。どちらのスキーマファイルにも同じ名前空間があり、"Customer"という要素を定義します。スキーマファイル LACust.xsd をスキーマオブジェクトに追加すると、BostonCust.xsd ファイルを保持するか、LACust.xsd ファイルで上書きするかを確認するように求められます。

どのルートレベルのスキーマファイルでも削除できます。スキーマファイルを削除すると、そのスキーマファイルで定義されている要素の要素型が xs:string に変更されます。

スキーマファイルを追加するには、**概要ビュー**を選択し、**【スキーマの場所】** リストの横の**【追加】** ボタンをクリックしています。スキーマファイルを選択します。スキーマファイルを削除するには、ファイルを選択して**【削除】** ボタンをクリックします。

スキーマオブジェクトのスキーマビュー

スキーマビューには、スキーマ内のグループ、要素、型、属性グループ、および属性がアルファベット順で一覧表示されます。**スキーマビュー**でグループ、要素、型、属性グループ、または属性を選択すると、右側のパネルにプロパティが表示されます。**スキーマビュー**で各.xsd ファイルを表示することもできます。

スキーマビューには、スキーマオブジェクト内の名前空間および.xsd ファイルが一覧表示されます。

スキーマビューでは、以下の操作を実行できます。

- スキーマ制約のリストを表示するには、**【ディレクティブ】** フォルダーを展開します。名前空間、プレフィックス、および場所を表示するには、リストからスキーマ制約を選択します。
- 名前空間のプレフィックス、生成されたプレフィックス、および場所を表示するには、名前空間を選択します。生成されたプレフィックスは変更できます。
- スキーマオブジェクトを.xsd ファイルとして表示するには、**【ソース】** を選択します。スキーマオブジェクトが他のスキーマを含む場合は、表示する.xsd ファイルを選択できます。
- スキーマの各名前空間のアルファベット順のグループ、要素、型、属性グループ、および属性を表示するには、**【デザイン】** を選択します。**【名前】** フィールドに 1 文字以上を入力することで、グループ、要素、型、属性グループ、および属性を名前でフィルタリングできます。
- 要素プロパティを表示するには、グループ、要素、型、属性グループ、または属性を選択します。選択したオブジェクトに基づいて、Developer ツールの右側のパネルに各フィールドが表示されます。

型を表示すると、型が別の型から派生しているかがわかります。インタフェースには親型が表示されます。また、子要素が制限 (restriction) と拡張 (extension) のどちらで値を継承したかも示されます。

名前空間のプロパティ

【名前空間】 ビューには、選択した名前空間のプレフィックスおよび場所が表示されます。

各スキーマファイルに関連付けられている名前空間は、異なるソースの同じ名前の要素を区別します。Uniform Resource Identifier (URI) 参照によって、要素と属性名を含んでいるファイルの場所が定義されます。

複数の名前空間を含んだスキーマをインポートすると、Developer tool によりそれらの名前空間がスキーマオブジェクトに追加されます。スキーマファイルに他のスキーマが含まれている場合は、それらのスキーマの名前空間も追加されます。

Developer ツールにより、それぞれの名前空間に対して生成されたプレフィックスが作成されます。スキーマにプレフィックスが含まれない場合は、Developer tool により名前空間プレフィックス tns0 が生成され、追加される各名前空間プレフィックスのプレフィックス番号が増えます。名前空間プレフィックス xs は、Developer ツールにより予約されています。名前空間プレフィックス xs を含むスキーマをインポートする場合、Developer tool により生成されたプレフィックス xs1 が作成されます。スキーマに生成されたプレフィックス値が含まれる場合は、Developer ツールによりプレフィックス番号が増えます。

例えば、Customer_Orders.xsd に名前空間が 1 つ指定されているとします。このスキーマには、Customers.xsd という別のスキーマが含まれています。Customers スキーマには別の名前空間が指定されています。Developer ツールにより、Customer_Orders 名前空間にはプレフィックス tns0 が、Customers 名前空間にはプレフィックス tns1 がそれぞれ割り当てられます。

名前空間の場所とプレフィックスを確認するには、**スキーマビュー**で名前空間を選択します。

複数のスキーマオブジェクトから Web サービスを作成する場合は、各名前空間に一意のプレフィックスが付加されている必要があります。各名前空間の生成されたプレフィックスは変更可能です。

要素のプロパティ

要素は、単純型または複合型です。複合型には他の型が含まれています。**スキーマビュー**で要素を選択すると、Developer ツールにより画面の右側に子要素とプロパティが一覧表示されます。

以下の表に、要素を選択したときに表示される要素のプロパティを示します。

| プロパティ | 説明 |
|-------|--------|
| 名前 | 要素の名前。 |
| 説明 | 型の説明。 |
| 型 | 要素の型。 |

以下の表に、要素を選択したときに表示される子要素のプロパティを示します。

| プロパティ | 説明 |
|--------|------------------------------|
| 名前 | 要素の名前。 |
| 型 | 要素の型。 |
| 最小出現回数 | インスタンスのある 1 箇所に要素が出現できる最小回数。 |

| プロパティ | 説明 |
|--------|------------------------------|
| 最大出現回数 | インスタンスのある 1 箇所に要素が出現できる最大回数。 |
| 説明 | 要素の説明。 |

子要素の追加のプロパティを表示するには、[説明] カラムの両矢印をクリックしてウィンドウを展開します。

以下の表に、[説明] カラムを展開したときに表示される子要素の追加のプロパティを示します。

| プロパティ | 説明 |
|-----------|--|
| 固定値 | 変化しない要素の特定の値。 |
| 空にすることが可能 | 要素に nil 値を指定できます。nil 要素には、要素タグのみが含まれ、値とコンテンツはありません。 |
| 抽象型 | 要素は抽象型です。インスタンスに、この型から派生した型が含まれている必要があります。要素型が派生していない抽象型は、無効な型となります。 |
| 最小値 | インスタンスでの要素の最小値。 |
| 最大値 | インスタンスでの要素の最大値。 |
| 最小長 | 要素の最小の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 最大長 | 要素の最大の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 列挙型 | 要素のすべての有効値のリスト。 |
| パターン | 要素の有効な値を定義する式パターン。 |

要素の詳細プロパティ

要素の詳細プロパティを表示するには、**スキーマビュー**で要素を選択します。**[詳細]** をクリックします。

以下の表に、要素の詳細プロパティを示します。

| プロパティ | 説明 |
|-------|--|
| 抽象型 | 要素は抽象型です。SOAP メッセージに、この型から派生した型が含まれている必要があります。要素型が派生していない抽象型は、無効な型となります。 |
| ブロック | 階層内でこの要素が派生素素で置き換えられないようにします。ブロック値には、「#all」を指定するか、extension、restriction、または substitution を含んだリストを指定できます。 |
| 最終 | スキーマで単純型が派生型として拡張または制限されないようにします。 |

| プロパティ | 説明 |
|----------|---|
| 置き換えグループ | 要素で置き換える要素の名前。 |
| NULL 可能 | 要素に nil 値を指定できます。nil 要素には、要素タグのみが含まれ、値とコンテンツはありません。 |

単純型のプロパティ

単純型要素は、構造化されていないテキストを含む要素です。**スキーマビュー**で単純型要素を選択すると、その単純型要素についての情報が右側のパネルに表示されます。

以下の表に、単純型について表示されるプロパティを示します。

| プロパティ | 説明 |
|--------|---|
| タイプ | 要素の名前。 |
| 説明 | 要素の説明。 |
| 種類 | 単純型が、共用体 (union)、リスト (list)、anyType、またはアトミック (atomic) かどうかを定義します。アトミック要素には、他の要素や型が含まれていません。 |
| メンバタイプ | Union 構造体に含まれている型の一覧。 |
| 項目タイプ | 要素の型。 |
| 基本型 | アトミック要素の基本型 (整数や文字列など)。 |
| 最小長 | 要素の最小の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 最大長 | 要素の最大の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 空白を省略 | 先頭または末尾の空白を削除します。複数のスペースは 1 つのスペースに縮小します。 |
| 列挙 | 型を有効値のリストに限定します。 |
| パターン | 型をパターン式で定義された値に限定します。 |

単純型の詳細プロパティ

単純型の詳細プロパティを表示するには、**スキーマビュー**で単純型を選択します。**[詳細]** をクリックします。

単純型のプロパティの下に、詳細プロパティが表示されます。

以下の表に、単純型の詳細プロパティを示します。

| プロパティ | 説明 |
|-------|-----------------------------------|
| 最終 | スキーマで単純型が派生型として拡張または制限されないようにします。 |

複合型のプロパティ

複合型とは、その他の要素や属性を含む要素のことです。複合型に含まれる要素は、単純型または複合型です。**スキーマビュー**で複合型を選択すると、Developer ツールにより画面の右側に子要素とそのプロパティが一覧表示されます。

以下の表に、複合型のプロパティを示します。

| プロパティ | 説明 |
|-------|--|
| 名前 | 型の名前。 |
| 説明 | 型の説明。 |
| 継承元 | 親型の名前。 |
| 継承方式 | 制限または拡張。複合型は親型から派生したものです。複合型では、親よりも要素または属性が少ない場合があります。逆に、要素と属性が追加される場合もあります。 |

複合型に含まれる各要素のプロパティを表示するには、[説明] カラムの両矢印をクリックしてウィンドウを展開します。

複合型の詳細プロパティ

複合型の詳細プロパティを表示するには、**スキーマビュー**で要素を選択します。**[詳細]** をクリックします。

以下の表に、複合要素または複合型の詳細プロパティを示します。

| プロパティ | 説明 |
|----------|--|
| 抽象型 | 要素は抽象型です。SOAP メッセージに、この型から派生した型が含まれている必要があります。要素型が派生していない抽象型は、無効な型となります。 |
| ブロック | スキーマ内でこの要素が派生要素で置き換えられないようにします。ブロック値には、「#all」を指定するか、extension、restriction、または substitution を含んだリストを指定できます。 |
| 最終 | スキーマで単純型が派生型として拡張または制限されないようにします。 |
| 置き換えグループ | 要素で置き換える要素の名前。 |
| NULL 可能 | 要素に nil 値を指定できます。nil 要素には、要素タグのみが含まれ、値とコンテンツはありません。 |

属性プロパティ

属性は単純型です。要素と複合型は属性を含みます。グローバル属性は、スキーマの一部として表示されます。**スキーマビュー**でグローバル属性を選択すると、Developer ツールにより画面の右側に属性プロパティと関連する型プロパティが一覧表示されます。

以下の表に、属性プロパティを示します。

| プロパティ | 説明 |
|-------|--|
| 名前 | 属性の名前。 |
| 説明 | 属性の説明。 |
| タイプ | 属性の型。 |
| 値 | 属性の型の値。属性の型の値が固定されているか、デフォルト値があるかを示します。値が定義されていない場合、プロパティにはデフォルトで 0 が表示されます。 |

以下の表に、型プロパティを示します。

| プロパティ | 説明 |
|-------|---|
| 最小長 | 型の最小の長さ。長さは、型によってバイト数、文字数、項目数のいずれかになります。 |
| 最大長 | 型の最大の長さ。長さは、型によってバイト数、文字数、項目数のいずれかになります。 |
| 空白を省略 | 先頭または末尾の空白を削除します。複数のスペースは 1 つのスペースに縮小します。 |
| 列挙 | 型を有効値のリストに限定します。 |
| パターン | 型をパターン式で定義された値に限定します。 |

スキーマオブジェクトの詳細ビュー

スキーマオブジェクトの詳細プロパティを確認します。

以下の表に、スキーマオブジェクトの詳細プロパティを示します。

| 名前 | 値 | 説明 |
|--------------------|---------------------------|---|
| elementFormDefault | Qualified または Unqualified | 要素に名前空間が必要かどうかを決定します。スキーマでは、プレフィックスまたはターゲット名前空間宣言で要素が修飾されます。非修飾値は、要素に名前空間が不要であることを示します。 |

| 名前 | 値 | 説明 |
|----------------------|---------------------------|---|
| attributeFormDefault | Qualified または Unqualified | ローカルに宣言された属性に名前空間が必要かどうかを決定します。スキーマでは、プレフィックスまたはターゲット名前空間宣言で属性が修飾されます。非修飾値は、属性に名前空間が不要であることを示します。 |
| ファイルの場所 | .xsd ファイルのフルパス | .xsd ファイル（インポートした場合）の場所。 |

スキーマオブジェクトの作成

階層スキーマファイルまたはサンプルファイルをインポートして、リポジトリ内にスキーマオブジェクトを作成できます。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダーを選択します。
2. **【ファイル】 > 【新規】 > 【スキーマ】** をクリックします。
【新しいスキーマ】 ダイアログボックスが表示されます。
3. スキーマファイルをインポートするには、**【スキーマから作成】** を選択してから、階層スキーマファイルを参照して選択します。
URI または参照するファイルシステムでの場所を入力します。選択したスキーマが Developer tool で検証されます。検証メッセージを確認します。Avro、Parquet、JSON、または.xsd スキーマファイルを選択できます。
注: URI に英語以外の文字が含まれていると、インポートが失敗する場合があります。任意のブラウザのアドレスバーに URI をコピーします。ブラウザから場所をコピーします。Developer ツールは、エンコードされた URI をブラウザから受け付けます。
4. サンプルファイルからスキーマを作成するには、**【サンプルファイルから作成する】** を選択してから、階層ファイルを参照して選択します。
Avro、Parquet、JSON、または XML ファイルを選択できます。
注: Avro、Parquet、JSON、または XML コンテンツが含まれていて拡張子が異なるファイルを選択した場合、ウィザードはファイルのコンテンツを認識します。
5. 必要に応じて、スキーマ名を変更します。
6. **【次へ】** をクリックして、スキーマ内の要素と型を一覧表示します。
7. **【完了】** をクリックしてスキーマをインポートします。
【オブジェクトエクスプローラ】 ビューで、**【スキーマオブジェクト】** の下にスキーマが表示されます。Developer tool では、スキーマが.xsd ファイルとして保存されます。
8. スキーマ名前空間の生成されたプレフィックスを変更するには、**【オブジェクトエクスプローラ】** ビューで名前空間を選択し、**名前空間ビュー**で、**【生成されたプレフィックス】** プロパティを変更します。

スキーマの更新

要素、属性、型、またはその他のコンポーネントの変更時に、スキーマオブジェクトを更新できます。スキーマオブジェクトを更新すると、そのスキーマを使用するオブジェクトが更新されます。

スキーマオブジェクトは、以下の方法で更新できます。

スキーマを同期する。

Developer ツールの外部でスキーマファイルを更新する場合は、スキーマオブジェクトを同期します。スキーマオブジェクトを同期すると、変更を含むすべてのスキーマ.xsd ファイルが再インポートされます。

スキーマファイルを編集する。

Developer ツール内からファイルを更新する場合は、スキーマファイルを編集します。スキーマファイルを編集すると、.xsd ファイルに使用するファイルがエディタで開きます。ファイルを別のエディタで開いたり、Developer ツールで.xsd ファイルに使用するデフォルトのエディタを設定することができます。

スキーマを使用して、Web サービスの要素タイプを定義できます。Web サービスの WSDL に含まれているスキーマを更新すると、その Web サービスが更新され、開いたときに変更済みとマークされます。Developer ツールが新旧のスキーマを比較するときは、名前属性によってスキーマコンポーネントを識別します。

名前属性が変更されていない場合、Web サービスはスキーマ変更で更新されます。たとえば、Developer ツール内からスキーマファイルを編集して要素「Item」の maxOccurs 属性を「10」から「unbounded」に変更するとします。ファイルを保存すると、その Item 要素を参照するすべての Web サービスの maxOccurs 属性が更新されます。

名前属性が変更された場合、その Web サービスを開くと変更済みとしてマークされます。たとえば、Developer ツール外でスキーマファイルを編集して複合要素型の名前を「Order」から「CustOrder」に変更するとします。それから、スキーマの同期を実行します。その要素を参照する Web サービスを開くと、その要素名はエディタでアスタリスク付きになり、Web サービスに変更が含まれることがわかります。Developer ツールは CustOrder 要素を Web サービスに追加しますが、Order 要素型を削除しません。Developer ツールで Order 要素の型が特定できなくなるため、要素型が xs:string に変更されます。

スキーマの同期

スキーマコンポーネントの変更時に、スキーマオブジェクトを同期できます。スキーマオブジェクトの同期化では、スキーマファイルからオブジェクトメタデータが再インポートされます。

Developer ツールの外部でスキーマオブジェクトに複雑な変更を加える場合、スキーマの同期を使用します。たとえば、次の操作の実行後にスキーマファイルを同期します。

- 複数のスキーマファイルを変更する。
- スキーマファイルにスキーマを追加、または削除する。
- インポートする要素または含める要素を変更する。

Developer ツールによって、スキーマオブジェクトの更新前にスキーマファイルが検証されます。スキーマファイルにエラーがある場合、Developer ツールはファイルをインポートしません。

スキーマオブジェクトを同期するには、**[Object Explorer]** ビューでスキーマオブジェクトを右クリックして、**[同期]** を選択します。

スキーマファイルの編集

スキーマファイルを Developer ツールで編集して、スキーマコンポーネントを更新できます。

少数のファイルにマイナーな更新を加えるには、Developer ツールでスキーマファイルを編集します。たとえば、スキーマファイルに対して、以下のマイナーな更新のいずれかを加えることができます。

- 要素の minOccurs または maxOccurs 属性を変更する。
- 属性を複合型に追加する。
- 単純オブジェクト型を変更する。

スキーマファイルを編集すると、Developer ツールによってスキーマファイルの一時コピーがエディタで開きます。.xsd ファイルを使用するシステムエディタで、または別のエディタを選択して、スキーマファイルを編集できます。xsd ファイル用のデフォルトエディタを Developer ツールで設定できます。編集後に一時スキーマファイルを保存します。

Developer ツールによって、スキーマオブジェクトの更新前に一時ファイルが検証されます。スキーマファイルには、スキーマオブジェクトの他のスキーマファイルと競合するコンポーネントやエラーが含まれている場合、ファイルはインポートされません。

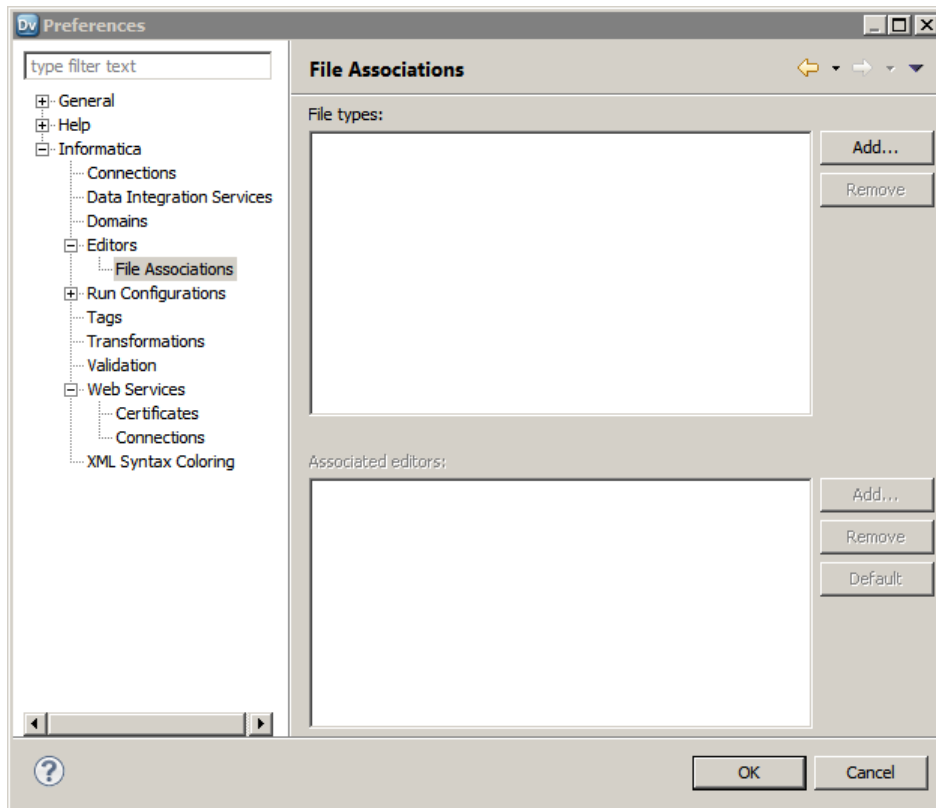
注: 一時スキーマファイルを編集および保存する場合、Developer ツールは【スキーマの場所】リストのスキーマファイルを更新しません。Developer ツールでスキーマファイルを編集した後にスキーマオブジェクトを同期した場合、同期操作によって編集内容が上書きされます。

デフォルトのスキーマファイルエディタの設定

スキーマファイルを編集するときに Developer ツールが開くデフォルトのエディタを設定できます。

1. 【ウィンドウ】 > 【設定】をクリックします。
【設定】ダイアログボックスが表示されます。
2. 【エディタ】 > 【ファイルの関連付け】をクリックします。

【設定】 ダイアログボックスが【ファイルの関連付け】 ページ表示されます。



3. 【ファイルタイプ】 領域の横にある【追加】 をクリックします。
【ファイルタイプの追加】 ダイアログボックスが表示されます。
4. ファイルタイプとして.xsd と入力し、【OK】 をクリックします。
5. 【関連付けられたエディタ】 領域の横にある【追加】 をクリックします。
【エディタの選択】 ダイアログボックスが表示されます。
6. エディタのリストからエディタを選択するか、【参照】 をクリックして別のエディタを選択して、【OK】 をクリックします。
選択したエディタが【関連付けられたエディタ】 リストに表示されます。
7. 必要に応じて、【関連付けられたエディタ】 リストに他のエディタを追加します。
8. 複数のエディタを追加した場合、デフォルトエディタを変更できます。エディタを選択して、【デフォルト】 をクリックします。
9. 【OK】 をクリックします。

スキーマファイルの編集

スキーマオブジェクトのスキーマファイルは編集できます。

1. スキーマオブジェクトを開きます。
2. 【概要】 ビューを選択します。

スキーマオブジェクトの【概要】ビューが表示されます。

3. 【スキーマの場所】リストのスキーマファイルを選択します。
4. 【アプリケーションから開く】をクリックして、以下のいずれかのオプションを選択します。

| オプション | 説明 |
|-----------|--|
| システムエディタ | スキーマファイルが、オペレーティングシステムで.xsd ファイル用に設定されているエディタで開きます。 |
| デフォルトエディタ | スキーマファイルが、Developer ツールのデフォルトエディタで開きます。このオプションは、デフォルトエディタを設定している場合にのみ表示されます。 |
| その他 | スキーマファイルを開くエディタを選択します。 |

Developer ツールは、スキーマファイルの一時コピーを開きます。

5. 一時スキーマファイルを更新し、変更を保存して、エディタを閉じます。
Developer によって、スキーマオブジェクトを更新するように求められます。
6. スキーマオブジェクトを更新するには、【スキーマオブジェクトの更新】をクリックします。
加えた変更によってスキーマファイルが更新されます。

証明書管理

Developer ツールでは、クライアント認証を必要とする URL から WSDL データオブジェクトとスキーマオブジェクトをインポートするために、証明書を使用する必要があります。

クライアント認証を必要とする URL をホストするサーバーが信頼されている証明書を使用している場合は、デフォルトで Developer ツールがその URL からオブジェクトをインポートします。この URL をホストするサーバーが信頼されていない証明書を使用している場合は、その証明書を Developer ツールに追加する必要があります。信頼されていない証明書を Developer ツールに追加しないと、Developer ツールではオブジェクトをインポートできません。オブジェクトのインポート元となる URL のサーバー管理者に、証明書ファイルとパスワードを要求してください。

Developer ツールに追加した証明書は、Developer ツールマシン上で実行するインポートに適用されます。
Developer ツールでは、証明書がモデルリポジトリに保管されません。

Informatica Developer の証明書プロパティ

クライアント認証を必要とする URL から、信頼されていない証明書を使用してオブジェクトをインポートする場合は、Developer ツールに証明書を追加します。

以下の表に、証明書のプロパティを示します。

| プロパティ | 説明 |
|------------|----------------------|
| ホスト名 | URL をホストするサーバーの名前。 |
| ポート番号 | URL のポート番号。 |
| 証明書ファイルのパス | クライアント証明書ファイルの場所。 |
| パスワード | クライアント証明書ファイルのパスワード。 |

Informatica Developer への証明書の追加

証明書を追加する場合は、証明書プロパティを設定します。これは、クライアント認証を必要とする URL から信頼されていない証明書を使用してオブジェクトをインポートする場合に、Developer ツールで使用するものです。

1. **【ウィンドウ】** > **【設定】** をクリックします。
2. **【Informatica】** > **【Web サービス】** > **【証明書】** を選択します。
3. **【追加】** をクリックします。
4. 証明書プロパティを設定します。
5. **【OK】** をクリックします。

第 5 章

SOAP Web サービスの作成方法

この章では、以下の項目について説明します。

- [SOAP Web サービスの作成の概要, 36 ページ](#)
- [型と要素, 37 ページ](#)
- [Web サービスの概要ビュー, 37 ページ](#)
- [Web サービスの WSDL ビュー, 39 ページ](#)
- [WSDL データオブジェクトからの Web サービスの作成, 39 ページ](#)
- [SOAP Web サービスを手動で作成, 41 ページ](#)

SOAP Web サービスの作成の概要

SOAP Web サービスを作成する場合は、リポジトリに Web サービスオブジェクトを作成します。SOAP Web サービスは、WSDL データオブジェクトから作成するか、手動で作成することができます。各 SOAP Web サービスには、1 つ以上の操作を組み込むことができます。

SOAP Web サービスを WSDL データオブジェクトから作成する場合は、Web サービスに組み込む操作を WSDL データオブジェクトから選択します。各操作には、SOAP 1.1 バインディングまたは SOAP 1.2 バインディングを使用できますが、両方は使用できません。

Web サービスを手動で作成する場合は、操作を作成し、各操作の操作入力、操作出力、および操作フォールトの要素を定義します。再利用可能なオブジェクトを使用して、各操作の操作入力および操作出力の要素を定義できます。

Web サービスオブジェクトには、**概要ビュー**と **WSDL ビュー**があります。**概要ビュー**で操作を作成および設定できます。WSDL ファイルの内容は **WSDL ビュー**で表示できます。

Web サービスを作成したら、各操作の操作マッピングを設定します。必要に応じて操作を Web サービスに追加できます。

型と要素

Web サービスを WSDL データオブジェクトから作成する場合は、WSDL データオブジェクトによって各操作の要素および要素型が定義されます。Web サービスを手動で作成する場合は、自分で要素を定義するか、スキーマオブジェクトの型を使用して要素を定義できます。

操作コンポーネントに、anyType 要素、any 要素、anyAttribute 属性、派生型要素、または置き換えグループが含まれている場合は、操作マッピングを設定する際に 1 つ以上の型、要素、または属性を選択する必要があります。例えば、操作入力に anyType 要素が含まれている場合は、入力トランスフォーメーションを設定する際に 1 つ以上の型を選択します。

スキーマオブジェクトを使用して要素型を定義する場合は、Developer ツールによってスキーマオブジェクトが Web サービスの WSDL に追加されます。スキーマオブジェクトを削除するか、Web サービスの WSDL でスキーマオブジェクトへのリンクを削除すると、Developer ツールによってそのスキーマオブジェクトで定義されていた要素の要素型が xs:string に変更されます。この要素型は、別の型に更新できます。

Web サービスを手動で作成する場合は、Web サービスが操作コンポーネントを定義するために使用する名前空間ごとに一意の生成されたプレフィックス値が必要となります。例えば、Web サービスがスキーマオブジェクト schemaA および schemaB を使用して型を定義している場合、schemaA と schemaB はどの名前空間に対しても同じ生成されたプレフィックス値を使用することはできません。

Web サービスの概要ビュー

Web サービスの概要ビューは、Web サービスに関する一般的な情報と、Web サービスの操作やコンポーネントの詳細情報を表示します。

全般プロパティ

以下の表に、Web サービスについて設定する全般プロパティを示します。

| プロパティ | 説明 |
|----------------|--|
| 名前 | Web サービスオブジェクト名。 |
| 説明 | Web サービスの説明。 |
| 名前空間 | Web サービスの targetNamespace。Web サービスが WSDL データオブジェクトに関連付けられている場合、このフィールドは読み取り専用です。 |
| プレフィックス | targetNamespace のプレフィックス。Web サービスが WSDL データオブジェクトに関連付けられている場合、このフィールドは読み取り専用です。 |
| WSDL データオブジェクト | Web サービスに関連付けられている WSDL データオブジェクト。このプロパティは、Web サービスが WSDL データオブジェクトから作成された場合にのみ表示されます。 |
| サービス名 | サービス名。デフォルト値は、Web サービスの名前、または関連する WSDL データオブジェクトで定義されているサービス名です。Web サービスが WSDL データオブジェクトに関連付けられている場合、このフィールドは読み取り専用です。 |

〔操作〕 領域

〔操作〕 領域の左側で操作を選択すると、右側に詳細が表示されます。〔操作〕 領域の左側には、操作のツリービューに関連付けられた入力、出力、およびフォールトが表示されます。

以下の表に、[操作] 領域の右側に表示される操作、入力プロパティおよび出力プロパティを示します。

| プロパティ | 説明 |
|------------|----------------------------------|
| 操作名 | 操作の名前。 |
| 説明 | 操作の説明。 |
| バインディングタイプ | 操作のバインディングタイプ。 |
| 入力 | 操作入力に関連付けられた WSDL メッセージ名。 |
| 出力 | 操作出力に関連付けられた WSDL メッセージ名。 |
| 要素名 | 操作入力または出力によって参照される XSD 要素の名前。 |
| 名前 | 操作入力または出力によって参照される要素の子要素。 |
| タイプ | 要素の型。 |
| 最小発生数 | XML インスタンスのある 1 箇所に要素が出現できる最小回数。 |
| 最大発生数 | XML インスタンスのある 1 箇所に要素が出現できる最大回数。 |
| 説明 | 要素の説明。 |

子要素の追加のプロパティを表示するには、[説明] カラムの両矢印をクリックしてウィンドウを展開します。

以下の表に、[説明] カラムを展開したときに表示される子要素の追加のプロパティを示します。

| プロパティ | 説明 |
|-----------|--|
| 固定値 | 変化しない要素の特定の値。 |
| 空にすることが可能 | 要素に nil 値を指定できます。nil 要素には、要素タグのみが含まれ、値とコンテンツはありません。 |
| 抽象型 | 要素は抽象型です。XML インスタンスに、この型から派生した型が含まれている必要があります。要素型が派生していない抽象型は、無効な型となります。 |
| 最小値 | XML インスタンスでの要素の最小値。 |
| 最大値 | XML インスタンスでの要素の最大値。 |
| 最小長 | 要素の最小の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 最大長 | 要素の最大の長さ。長さは、要素型によってバイト数、文字数、項目数のいずれかになります。 |
| 列挙型 | 要素のすべての有効値のリスト。 |
| パターン | 要素の有効な値を定義する式パターン。 |

操作マッピングの領域

以下の表に、操作マッピングのカラムを示します。

| プロパティ | 説明 |
|---------|------------------------------|
| 操作マッピング | 操作マッピング名。 |
| 操作 | 操作名。 |
| 入力 | 操作入力に関連付けられた WSDL メッセージ名。 |
| 出力 | 操作出力に関連付けられた WSDL メッセージ名。 |
| フォールト | 操作フォールトに関連付けられた WSDL メッセージ名。 |

Web サービスの WSDL ビュー

WSDL ファイルのコンテンツは、Web サービスの **WSDL** ビューでプレビューできます。

WSDL データオブジェクトから Web サービスを作成する場合、WSDL ファイルのコンテンツを表示できます。

Web サービスを手動で作成する場合は、Developer ツールにより、操作入力、操作出力、操作フォールト、およびヘッダーの設定に基づいて WSDL ファイルのコンテンツが生成されます。WSDL のスキーマに要素を追加するか、WSDL のスキーマの要素を編集します。インポートしたスキーマオブジェクトを WSDL から削除することもできます。

操作入力、操作出力、操作フォールト、およびヘッダーはそれぞれ、WSDL 内のメッセージに対応します。操作フォールトとヘッダーは、WSDL 内でメッセージを共有する可能性があります。WSDL のスキーマの要素を使用してフォールトまたはヘッダーを作成した場合は、メッセージ名と要素名が同じになります。

WSDL デザインまたは WSDL ソースを表示できます。WSDL デザインでは、WSDL のコンテンツが階層状に表示されます。WSDL ソースでは、XML 形式の WSDL のコンテンツが表示されます。

Developer ツールで WSDL をプレビューするには、Web サービスの **WSDL** ビューで、**[表示]** フィールドの横にある **[ソース]** を選択します。

WSDL データオブジェクトからの Web サービスの作成

SOAP Web サービスを WSDL データオブジェクトから作成する場合は、WSDL によって 1 つ以上の操作が定義され、さらに各操作入力、操作出力、および操作フォールトの要素が定義されます。

WSDL に選択可能な操作が複数ある場合は、操作を追加できます。WSDL で定義されていない操作は作成できません。

WSDL データオブジェクトから Web サービスを作成する前に、WSDL データオブジェクトを作成する必要があります。WSDL は SOAP 1.1 または SOAP 1.2 に基づきます。**WSDL データオブジェクトからの Web サービス**

の作成ウィザードを使用すると、WSDL データオブジェクトから Web サービスを作成する前に WSDL データオブジェクトを作成できます。

WSDL データオブジェクトから Web サービスを作成する場合、Web サービスは WSDL データオブジェクトに依存します。WSDL データオブジェクトを削除すると、操作マッピングが無効になり、Web サービスに WSDL データオブジェクトを関連付ける必要が生じます。WSDL データオブジェクトがリポジトリから削除されると、Developer ツールによって Web サービスと WSDL データオブジェクトの関連付けが削除されます。

手順 1.WSDL データオブジェクトからの Web サービスの作成

Web サービスを WSDL データオブジェクトから作成する場合は、Developer ツールによって、WSDL データオブジェクトに定義されている情報を基に Web サービス内の各操作に操作マッピングが作成されます。

1. **Object Explorer** ビューで、プロジェクトまたはフォルダを選択します。
2. **[ファイル] > [新規] > [データサービス]** をクリックします。
[新規データサービス] ダイアログボックスが表示されます。
3. **[WSDL データオブジェクトから Web サービス]** をクリックし、**[次へ]** をクリックします。
4. WSDL データオブジェクトを作成する場合は、**[新しい WSDL データオブジェクト]** をクリックします。
 - a. **[場所]** オプションの横にある **[参照]** をクリックして、WSDL の場所を入力します。 **[OK]** をクリックします。
 - b. WSDL の名前を入力します。
 - c. **[完了]** をクリックします。

5. Web サービスの名前を入力します。

6. **[WSDL データオブジェクト]** オプションの横にある **[参照]** をクリックして、WSDL データオブジェクトを選択します。

7. **[操作]** オプションの横にある **[参照]** をクリックして、Web サービスに組み込む操作を 1 つ以上選択します。

WSDL データオブジェクトには、SOAP 1.1 形式および SOAP 1.2 形式の複数のバインディングを含めることができます。これらのバインディングの 1 つからのみ、特定の操作を選択できます。

8. **[次へ]** をクリックします。

9. デフォルトでは、選択する操作の SOAP バージョンが **[操作タイプ]** として表示されます。

10. 必要に応じて、各操作入力の **[マッピング入力]** タブを選択して、操作入力から出力ポートにデータをマップします。

データのマッピングは、入力トランスフォーメーションを設定する際に行うこともできます。

11. 必要に応じて、各操作出力の **[マッピング出力]** タブを選択して、入力ポートから操作出力にデータをマップします。

データのマッピングは、出力トランスフォーメーションを設定する際に行うこともできます。

12. 必要に応じて、各操作フォールトの **[マッピングフォールト]** タブを選択して、入力ポートから操作フォールトにデータをマップします。

データのマッピングは、フォールトトランスフォーメーションを設定する際に行うこともできます。

13. **[完了]** をクリックします。

手順 2.Web サービスへの操作の追加

必要に応じて操作を Web サービスに追加できます。

1. **[Object Explorer]** ビューで、WSDL データオブジェクトに依存している Web サービスを開きます。

2. **概要ビュー**を選択します。
3. **【操作】** セクションで、**【操作マッピング】** 領域を選択します。
4. **【操作マッピング】** 領域で、**【選択】** をクリックします。
【1 つ以上の操作を選択】 ダイアログボックスが表示されます。
5. 追加する操作を選択して、**【OK】** をクリックします。

WSDL データオブジェクトを Web サービスに関連付ける

WSDL データオブジェクトを Web サービスに関連付けるには、WSDL データオブジェクトを選択し、各操作マッピングを WSDL データオブジェクトで定義されている操作に関連付けます。

WSDL データオブジェクトから Web サービスを作成する場合、Web サービスは WSDL データオブジェクトに依存します。WSDL データオブジェクトを削除すると、操作マッピングが無効になり、Web サービスに WSDL データオブジェクトに関連付ける必要が生じます。

1. **Object Explorer** ビューで、WSDL データオブジェクトに依存している Web サービスを開きます。
2. **概要ビュー**を選択します。
3. **【WSDL データオブジェクト】** オプションの横にある **【参照】** をクリックします。
【WSDL データオブジェクトの選択】 ダイアログボックスが表示されます。
4. WSDL データオブジェクトを選択し、**【OK】** をクリックします。
5. **【操作マッピング】** 領域で、WSDL 操作を Web サービスの各操作マッピングに関連付けます。
 - a. 操作マッピング行の **【操作】** カラムを右クリックし、**【操作の選択】** をクリックします。
【操作の選択】 ダイアログボックスが表示されます。
 - b. 操作を選択し、**【OK】** をクリックします。

SOAP Web サービスを手動で作成

WSDL データオブジェクトを使用せずに SOAP Web サービスを作成する場合は、Web サービスプロパティ、操作、および操作コンポーネントを定義します。操作コンポーネントには、操作出力、操作入力、および操作フォールトがあります。

Web サービスを手動で作成する際に、1 つ以上の操作を定義することができます。操作を手動で作成する場合は、SOAP バインディングタイプを選択し、スキーマオブジェクトの要素および型を使用して操作の要素を定義できます。また、マップレット、再利用可能なトランスフォーメーション、論理データオブジェクト、フラットファイルデータオブジェクト、またはリレーショナルデータオブジェクトを使用して操作を作成することもできます。再利用可能なオブジェクトから操作を作成するときは、操作に含めるフィールドをオブジェクトから選択します。

Developer tool によって、定義した Web サービスのプロパティおよび操作を基に、WSDL ファイルのコンテンツが生成されます。WSDL ファイルのコンテンツは、Web サービスを作成した後にプレビューできます。

手順 1.Web サービスの手動作成

Web サービスを作成するには、**Web サービスの作成ウィザード**を使用します。

1. **Object Explorer** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】** > **【新規】** > **【データサービス】** をクリックします。

【新規データサービス】ダイアログボックスが表示されます。

3. 【Web サービス】をクリックします。

【新しい Web サービス】ダイアログボックスが表示されます。

4. Web サービスの名前を入力します。
5. 必要に応じて、名前空間と名前空間のプレフィックスを入力します。
6. 【次へ】をクリックします。

【新しい Web サービス】ダイアログボックスが表示されます。

操作の作成は、Web サービスを作成してから行う必要があります。

手順 2.操作の作成

Web サービスを手動で作成する場合、Web サービスの作成ウィザードまたは新しい操作ウィザードから操作を作成できます。

以下の表に、操作を作成するオプションを示します。

| 操作を作成するオプション | パス |
|-------------------------|---|
| Web サービスの作成ウィザードを使用します。 | 【新規】 > 【データサービス】 > 【Web サービス】をクリックします。 再利用可能なオブジェクトから、または空のオブジェクトから。 注: 再利用可能なオブジェクトには、物理データオブジェクト、マプレット、トランスフォーメーションを指定できます。 |
| 新しい操作ウィザードを使用します。 | Object Explorer または概要ビューを右クリックして、【新規】 > 【操作】をクリックします。 |
| 新しい操作ウィザードを使用します。 | Web サービスを作成し、Web サービスの概要ビューを選択します。【操作】領域で、【新規】 ボタンの横の矢印をクリックして【新しい操作】を選択します。 |
| 新しい操作ウィザードを使用します。 | 再利用可能なオブジェクトを、概要ビューの【操作】領域にドラッグします。 |

再利用可能なオブジェクトからの操作の作成

Web サービスの作成ウィザードを使用して、Web サービスを作成し、再利用可能なオブジェクトから操作を作成します。

論理データオブジェクト、フラットファイルデータオブジェクト、およびリレーショナルデータオブジェクトのデータをルックアップする操作を作成できます。マプレットまたは再利用可能なトランスフォーメーションから操作を作成することもできます。マプレットまたは再利用可能なトランスフォーメーションにより、操作入力および操作出力の要素が定義されます。Web サービスコンシューマトランスフォーメーションから操作を作成することはできません。

Web サービスの作成手順を完了すると、**Web サービスの作成**ウィザードを使用して、データオブジェクト、マブレット、または再利用可能なトランスフォーメーションから操作を作成できます。

1. **【新しい Web サービス】** ダイアログ ボックスで、**【新規】** ボタンの横の矢印をクリックします。次に、**【操作】** > **【再利用可能なオブジェクトから作成】** を選択します。

【再利用可能なオブジェクトの選択】 ダイアログボックスが表示されます。

注: データオブジェクトから操作を作成することもできます。その場合、Web サービスを **Object Explorer** で開き、**【概要】** > **【操作】** 領域を選択します。**【新規】** ボタンの横にある矢印をクリックします。

2. オブジェクトを選択し、**【OK】** をクリックします。

3. 必要に応じて、操作の名前と説明を入力します。

Developer ツールにより、操作に指定した名前を基に、操作入力および操作出力の名前が定義されます。

4. 操作のバインディングタイプを選択します。

デフォルトは SOAP 1.1 です。操作を作成して保存した後は、操作のバインディングタイプを変更できません。

5. 操作入力をクリックしてそのプロパティを表示し、設定します。

- a. **【操作入力】** タブで、操作入力フィールドを選択します。

注: オブジェクトに複数の入力グループがある場合は、**【操作入力】** タブで入力グループおよび操作入力フィールドを選択します。

- b. 必要に応じて、各要素の最小出現回数と最大出現回数を設定します。

- c. 必要に応じて、**【マッピング入力】** タブをクリックして、操作入力から出力ポートへのデータのマッピング方法を設定します。

データのマッピングは、入力トランスフォーメーションを設定する際に行うこともできます。

6. 操作出力をクリックしてそのプロパティを表示し、設定します。

- a. **【操作出力】** タブで、操作出力フィールドを選択します。

注: オブジェクトに複数の出力グループがある場合は、**【操作出力】** タブで出力グループおよび操作出力フィールドを選択します。

- b. 必要に応じて、各要素の最小出現回数を設定します。

- c. 必要に応じて、**【マッピング出力】** タブをクリックして、入力ポートから操作出力へのデータのマッピング方法を設定します。

データのマッピングは、出力トランスフォーメーションを設定する際に行うこともできます。

7. 操作でユーザー定義フォールトを送信する場合は、**【新規】** ボタンの横にある矢印をクリックし、**【フォールト】** をクリックします。

フォールトの要素を作成するか、または再利用可能な要素を選択できます。

8. 各操作フォールトをクリックしてそのプロパティを表示し、設定します。

【マッピングフォールト】 タブをクリックして、入力ポートから操作フォールトへのデータのマッピング方法を設定できます。データのマッピングは、フォールトトランスフォーメーションを設定する際に行うこともできます。

9. 手順 [1](#) から [8](#) を繰り返して、その他の操作の作成と設定を行います。

10. **【完了】** をクリックします。

各操作に対して行った設定を基に、Developer ツールによって操作の操作マッピングが作成されます。

オプションで要素または定義済みフォールトを作成できます。

手動での操作の作成

Web サービスの作成ウィザードを使用して、操作を定義し、各操作の操作入力、操作出力、および操作フォールトを定義します。

Web サービスの作成手順を完了すると、**Web サービスの作成**ウィザードを使用して操作を作成および設定できます。

1. **【新しい Web サービス】** ダイアログ ボックスで、**【新規】** ボタンの横の矢印をクリックします。次に、**【操作】** > **【空の状態から作成】** を選択します。
 - a. 操作の名前を入力します。
Developer ツールでは、操作名を使用して操作入力および操作出力の名前を定義します。
 - b. SOAP バージョンの操作を **【バインディングタイプ】** として選択します。
 - c. フォールトメッセージを定義するには、**【新規】** ボタンの横にある矢印をクリックし、**【フォールト】** をクリックします。
フォールトメッセージの各要素を作成するか、最良可能な要素を選択するかを選ぶことができます。
ウィザードにより、フォールトメッセージの詳細要素に要素が追加されます。
2. 操作入力をクリックしてそのプロパティを表示し、設定します。
 - a. **【操作入力】** タブをクリックします。
 - b. 要素を追加するには、**【新規】** ボタンの横にある矢印をクリックし、**【要素】** をクリックします。子要素を追加するには、要素を選択し、**【新規】** ボタンの横にある矢印をクリックし、**【子要素】** をクリックします。
 - c. 各要素の型を指定するには、**【タイプ】** フィールドで選択ボタンをクリックします。XSD 型またはスキーマオブジェクト型を選択し、**【OK】** をクリックします。
ヒント: **【タイプ】** フィールドをクリックし、選択する型の最初の数文字を入力します。入力した文字で始まる名前を持つ XSD 型の一覧が表示されます。
 - d. 各要素の最小出現回数と最大出現回数を設定します。
 - e. 必要に応じて、各要素の説明を入力します。
 - f. 必要に応じて、**【マッピング入力】** タブをクリックして、操作入力から出力ポートにデータをマップします。
データのマッピングは、入力トランスフォーメーションを設定する際に行うこともできます。
3. 各操作フォールトをクリックしてそのプロパティを表示し、設定します。
【マッピングフォールト】 タブをクリックして、入力ポートから操作フォールトへのデータのマッピング方法を設定できます。データのマッピングは、フォールトトランスフォーメーションを設定する際に行うこともできます。
4. 操作出力をクリックしてそのプロパティを表示し、設定します。
 - a. **【操作出力】** タブをクリックします。
 - b. 要素を追加するには、**【新規】** ボタンの横にある矢印をクリックし、**【要素】** をクリックします。子要素を追加するには、要素を選択し、**【新規】** ボタンの横にある矢印をクリックし、**【子要素】** をクリックします。
 - c. 各要素の型を指定するには、**【タイプ】** フィールドで選択ボタンをクリックします。XSD 型またはスキーマオブジェクト型を選択し、**【OK】** をクリックします。
 - d. 各要素の最小出現回数と最大出現回数を設定します。
 - e. 必要に応じて、各要素の説明を入力します。

- f. 必要に応じて、**【マッピング出力】** タブをクリックして、入力ポートから操作出力にデータをマップします。

データのマッピングは、出力トランスフォーメーションを設定する際に行うこともできます。

5. 手順 [1](#) から [4](#) を繰り返して、その他の操作の追加と設定を行います。

6. **【完了】** をクリックします。

各操作に対して行った設定を基に、Developer ツールによって操作の操作マッピングが作成されます。

オプションで要素または定義済みフォールトを作成できます。

新しい操作ウィザードからの操作の作成

Web サービスを手動で作成した後に、新しい操作ウィザードを使用して操作を作成します。

1. 操作を作成するメソッドを選択します。

- Object Explorer で Web サービスを右クリックし、**【新規】** > **【操作】** を選択します。
- [アウトライン] ビューで Web サービスを右クリックし、**【新規】** > **【操作】** を選択します。
- **Object Explorer** で Web サービスを開き、**【概要】** > **【操作】** 領域を選択します。 **【新規】** ボタンの横にある矢印をクリックします。

【新しい操作】 ダイアログボックスが表示されます。

2. 操作の名前と、必要に応じて説明を入力します。

Developer ツールでは、操作名を使用して操作入力および操作出力の名前を定義します。

3. **【次へ】** をクリックします。

4. 操作のバインディングタイプを選択します。

デフォルトは SOAP 1.1 です。操作を作成して保存した後は、操作のバインディングタイプを変更できません。

5. 必要に応じて、各操作入力の **【マッピング入力】** タブを選択して、操作入力から出力ポートにデータをマップします。

データのマッピングは、入力トランスフォーメーションを設定する際に行うこともできます。

6. 必要に応じて、各操作出力の **【マッピング出力】** タブを選択して、入力ポートから操作出力にデータをマップします。

データのマッピングは、出力トランスフォーメーションを設定する際に行うこともできます。

7. 必要に応じて、**【新規】** ボタンの横にある矢印をクリックして、操作のフォールトを作成します。

フォールトメッセージの各要素を作成するか、再利用可能な要素を選択するかを選ぶことができます。ウィザードにより、フォールトメッセージの詳細要素に要素が追加されます。データのマッピングは、フォールトトランスフォーメーションを設定する際に行うこともできます。

8. **【完了】** をクリックします。

オプションで要素または定義済みフォールトを作成できます。

再利用可能なオブジェクトのドラッグ

再利用可能なオブジェクトから操作を作成するには、**概要ビュー**の **【操作】** 領域に再利用可能なオブジェクトをドラッグします。

1. **【Object Explorer】** ビューから Web サービスを開きます。
2. **概要ビュー** から、**【操作】** 領域を選択します。

3. 再利用可能なオブジェクトを、[Object Explorer] ビューから、概要ビューの **【操作】** 領域にドラッグします。
【新しい操作】 ダイアログボックスが表示されます。
4. 操作のバインディングタイプを選択します。
デフォルトは SOAP 1.1 です。操作を作成して保存した後は、操作のバインディングタイプを変更できません。
5. 必要に応じて、各操作入力の **【マッピング入力】** タブを選択して、操作入力から出力ポートにデータをマップします。
データのマッピングは、入力トランスフォーメーションを設定する際に行うこともできます。
6. 必要に応じて、各操作出力の **【マッピング出力】** タブを選択して、入力ポートから操作出力にデータをマップします。
データのマッピングは、出力トランスフォーメーションを設定する際に行うこともできます。
7. 必要に応じて、**【新規】** ボタンの横にある矢印をクリックして、操作のフォールトを作成します。
データのマッピングは、フォールトトランスフォーメーションを設定する際に行うこともできます。
8. **【完了】** をクリックします。
オプションで要素または定義済みフォールトを作成できます。

手順 3.要素の作成

必要に応じて、Web サービスの **WSDL** ビューで要素を作成できます。Web サービスの **WSDL** ビューでは、作成する要素と共に、操作フォールトまたはヘッダーを作成することができます。要素には、複数の要素および子要素を含めることができます。

要素を作成する前に、Web サービスを作成し、**Create a Web Service** ウィザードから 1 つ以上の操作を定義します。

1. Web サービスを開きます。
2. **WSDL** ビューを選択します。
3. WSDL の **【デザイン】** を表示します。次に **【スキーマ】** セクションで、要素を選択するか、要素の上の名前空間エントリを選択します。
4. **【新しい要素】** ボタンをクリックします。
5. 要素に要素を追加するには、**【新規】** ボタンの横にある矢印をクリックし、**【要素】** をクリックします。
6. 要素に子要素を追加するには、要素を選択し、**【新規】** ボタンの横にある矢印をクリックし、**【子要素】** をクリックします。
7. 必要に応じて、各要素の型を設定します。
 - a. **【タイプ】** フィールドで **【選択】** ボタンをクリックします。
 - b. XSD 型またはスキーマオブジェクト型を選択し、**【OK】** をクリックします。
8. 必要に応じて、各要素の最小出現回数と最大出現回数を設定します。
9. 必要に応じて、各要素の説明を入力します。
オプションで定義済みフォールトを作成できます。

要素の階層レベルの変更

要素を作成した後で、階層レベルを変更することができます。操作を手動で作成し、要素または子要素の削除、移動、コピーなどの操作を実行する場合には、**【操作入力】** タブおよび **【操作出力】** タブを右クリックします。

要素または子要素をドラッグして、要素の階層を変更することもできます。

手順 4.定義済みフォールトの作成

必要に応じて、SOAP レスポンス内でユーザー定義エラーを送信するためのフォールトを作成できます。定義済みフォールトを作成すると、Developer ツールによりフォールトトランスフォーメーションが操作マッピングに追加されます。

Web サービスを作成して、1 つ以上の操作を定義します。操作作成時にフォールトを作成するか、操作にフォールトを追加することができます。操作には複数のフォールトを指定できます。

1. Web サービスを開きます。
2. **【アウトライン】** ビューで、操作マッピングを選択します。
操作マッピングがエディタに表示されます。
3. **【プロパティ】** ビューで **【操作】** タブをクリックします。
4. **【新規】** ボタンの横にある矢印をクリックします。
5. **【フォールト】** をクリックします。
6. 要素を作成するか、要素を再利用するかを選択します。
 - フォールトの要素を作成する場合は、**【新しい要素の作成】** を選択します。子要素と複数出現要素を作成できます。
 - フォールトの要素を再利用する場合は、**【既存の要素の再利用】** を選択します。

Developer ツールにより、フォールトメッセージ階層の詳細要素に要素が追加されます。

手順 5.ヘッダーの作成

SOAP メッセージのヘッダー内のデータを送受信するためのヘッダーを作成します。操作入力および操作出力には、ヘッダーを作成できます。HTTP POST 方式を使用できます。

1. Web サービスを開きます。
2. **アウトライン** ビューで操作マッピングを選択します。
操作マッピングがエディタに表示されます。
3. **プロパティ** ビューで **【操作】** タブをクリックします。
4. 操作入力または操作出力にヘッダーを追加することを選択します。
 - 操作入力にヘッダーを追加するための入力を選択します。
 - 操作出力にヘッダーを追加するための出力を選択します。
5. **【新規】** ボタンの横にある矢印をクリックします。
6. **【ヘッダー】** をクリックし、要素を作成するか再利用するかを選択します。
 - ヘッダーの要素を作成する場合は、**【新しい要素の作成】** を選択します。
 - ヘッダーの要素を再利用する場合は、**【既存の要素の再利用】** を選択します。

第 6 章

操作マッピング

この章では、以下の項目について説明します。

- [操作マッピングの概要, 48 ページ](#)
- [操作マッピングの \[全般\] タブ, 49 ページ](#)
- [操作マッピングの \[操作\] タブ, 49 ページ](#)
- [操作マッピングの \[詳細\] タブ, 49 ページ](#)
- [入力トランスフォーメーション, 50 ページ](#)
- [出力トランスフォーメーション, 52 ページ](#)
- [フォールトトランスフォーメーション, 55 ページ](#)
- [フォールト処理, 59 ページ](#)
- [操作マッピングのテスト, 62 ページ](#)
- [ビューをカスタマイズするためのオプション, 63 ページ](#)

操作マッピングの概要

Informatica SOAP Web サービスには操作マッピングがあります。操作マッピングでは、Web サービスクライアントに対して Web サービス操作が実行されます。

操作マッピングには、入力トランスフォーメーション、出力トランスフォーメーション、および複数のフォールトトランスフォーメーションを含めることができます。入力、出力、フォールトの各トランスフォーメーションによって、SOAP メッセージが処理されます。マッピングには、クライアントから要求された Web サービス操作に基づいてデータの取得、変換、または更新を行うその他のトランスフォーメーションを含めることもできます。

Developer tool で Web サービスを作成した後に、Web サービス内の各操作に対して操作マッピングを設定します。操作マッピングは、操作のロジックを表します。

操作マッピングを設定するときに、データ統合サービスが SOAP 要求で受信したデータをどのように処理するかを定義します。SOAP 要求は、操作マッピングに関連付けられているバインディング操作によって使用されるバインディングタイプに基づいて、SOAP 1.1 または SOAP 1.2 形式にすることができます。

入力トランスフォーメーションは、Web サービスクライアントから SOAP 要求を受信し、マッピングのダウンストリームのトランスフォーメーションにデータを返します。データを返されたトランスフォーメーションは、クライアントが要求した操作を実行します。

出力トランスフォーメーションは、クライアントに返すデータを受信します。また出力トランスフォーメーションは、クライアントに送信する SOAP 応答メッセージを生成します。

エラーが発生した場合、データ統合サービスはフォールトを生成します。データ統合サービスサービスはフォールトトランスフォーメーションからユーザー定義フォールトを返します。ユーザー定義フォールトには2つの種類のフォールトがあります。定義済みフォールトと汎用フォールトです。

操作マッピングを設定するには、以下の手順を実行します。

1. 入力、出力、フォールトの各トランスフォーメーションを設定します。
2. その他のトランスフォーメーションの作成と設定を行って、操作ロジックを実装します。
3. ポートをリンクします。
4. マッピングを検証し、保存します。

操作マッピングの【プロパティ】ビューの【全般】タブ、【操作】タブ、および【詳細】タブで、操作マッピングのプロパティを表示して設定できます。

操作マッピングの【全般】タブ

【プロパティ】ビューの【全般】タブでは、操作マッピングの名前と説明を設定します。

【全般】タブには、操作マッピングに関連する操作の名前も表示されます。

操作マッピングの【操作】タブ

【プロパティ】ビューの【操作】タブでは、操作のプロパティを表示または設定します。

Web サービスを手動で作成する場合、【操作】タブを使用してフォールトを定義するか、操作入力または出力を更新できます。操作入力または操作出力にヘッダーを追加することもできます。

操作マッピングの【詳細】タブ

【プロパティ】ビューの【詳細】タブでは、操作マッピングの詳細プロパティを設定します。詳細プロパティを設定するときに、Data Integration Service に SOAP リクエストの XML を検証させるかどうかを指定できます。

以下の表に、操作マッピングの詳細プロパティを示します。

| プロパティ | 説明 |
|------------|--|
| XML スキーマ検証 | SOAP リクエストメッセージを実行時に検証します。【無効な XML でのエラー】または【検証なし】を選択します。XML が無効の場合、Data Integration Service は SOAP レスポンス内でフォールトを返し、Web サービスランタイムログにエラーを記録します。 |

入力トランスフォーメーション

入力トランスフォーメーションは、Web サービス WSDL での入力要素およびヘッダー要素を表します。入力トランスフォーメーションは、クライアントから SOAP リクエストを受信します。次に、XML メッセージをリレーショナルデータのグループへと解析し、そのデータを操作マッピングの他のトランスフォーメーションに渡します。

Web サービスに対して操作入力を定義すると、Developer ツールによって入力トランスフォーメーションが作成されます。

入力トランスフォーメーションの【ポート】タブを使用して、操作入力階層の表示、出力ポートの定義、および操作入力から出力ポートへのデータのマッピングを行います。操作入力階層によって、SOAP リクエストメッセージ階層が定義されます。

リレーショナルデータのグループを別々の出力ポートで返す代わりに、SOAP リクエスト全体を XML としてマップすることができます。SOAP リクエストを XML としてマップすると、Data Integration Service から SOAP メッセージ全体が 1 つの出力ポートで返されます。

入力トランスフォーメーションの【ポート】タブ

【ポート】タブでは、出力グループの定義、出力ポートの定義、および操作入力から出力ポートへのノードのマッピングを行います。

操作入力階層を表示する必要がない場合は、ポートの表示を選択します。ポートを表示しているときは、グループの定義、ポートの定義、および操作入力から出力ポートへのノードのマッピングを行うことができます。操作入力から出力ポートにノードをマップするには、【場所】カラムのフィールドをクリックし、【場所の選択】ダイアログボックスで階層を展開します。次に、階層からノードを選択します。

操作入力階層を表示する場合は、入力マッピングの表示を選択します。タブの左側は【操作入力】領域、タブの右側は【ポート】領域です。【操作入力】領域には、SOAP リクエストメッセージ階層が表示されます。【ポート】領域では、出力ポートを定義できます。操作入力から出力ポートにノードをマップすると、【操作入力】領域の【場所】カラムにノードの場所が表示されます。

入力マッピングを表示しているときは、階層内の出力ポートを表示することができます。出力ポートを操作入力のノードに接続する線を表示することもできます。

階層の第 1 レベルをマップすることを選択すると、Developer ツールによって操作入力の第 1 レベルのノードが出力ポートにマップされます。また、データをマップするために必要な出力ポートが作成されます。階層の第 1 レベルに、複数出現子ノードを 1 つ以上持つ複数出現親ノードが含まれている場合は、Developer ツールでポートの作成または階層の第 1 レベルのマッピングが行われません。

操作入力をポートにマップするためのルールとガイドライン

入力トランスフォーメーションを設定する場合は、操作入力階層から出力ポートにノードをマップします。

操作入力階層から出力ポートにノードをマップする際には、以下のルールとガイドラインを考慮します。

- ノードおよび出力ポートのデータ型に互換性があることが必要です。
- グループ内の複数の出力ポートにノードをマップすることはできません。

入カトランスフォーメーションの設定

Web サービス操作が SOAP リクエストメッセージを受信する場合は、その要求を処理するように入カトランスフォーメーションを設定します。このトランスフォーメーションで出力ポートを定義します。これらの出力ポートに操作入力からノードをマップします。

1. エディタで入カトランスフォーメーションを選択します。
2. **【プロパティ】** ビューの **【ポート】** タブをクリックします。
3. **【入カマッピング】** をクリックします。
【操作入力】 領域に、要求メッセージ階層が表示されます。**【ポート】** 領域で出力ポートを定義します。
4. 必要に応じて、**【行の表示】** をクリックして、出力ポートを操作入力のノードに接続する線を表示します。
すべての線を表示することも、選択したポートの線を表示することもできます。
5. 必要に応じて、**【階層で表示】** をクリックして、出力ポートを階層形式で表示します。
各子グループは親グループの下に表示されます。
6. 操作入力に anyType 要素、any 要素、anyAttribute 属性、派生型要素、または置き換えグループが含まれている場合は、**【操作入力】** 領域でオブジェクトを選択します。ノードの **【タイプ】** カラムで **【選択】** をクリックし、リストから 1 つ以上のタイプ、要素、または属性を選択します。
7. 出力グループを追加するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|------------------------------|---|
| ノードをドラッグする | 【操作入力】 領域内のグループノードまたは子ノードから、 【ポート】 領域の空のカラムまでポインタをドラッグします。グループノードの場合、Developer ツールによってポートのないグループが追加されます。ほかに出力グループが存在する場合は、 【新しいグループにマップ】 ダイアログボックスでグループを別のグループに関連付けるように求められます。Developer ツールによって関連するグループにキーが作成されます。 |
| 手動でグループを追加する | 【新規】 > 【グループ】 をクリックしてグループを追加します。 |
| 【階層の第 1 レベルの抽出】 を選択する | 【階層の第 1 レベルの抽出】 を選択します。Developer ツールにより、操作入力の第 1 レベルのノードが出力ポートおよび出力グループにマップされます。また、データをマップするために必要な出力ポートと出力グループが作成されます。階層の第 1 レベルに、複数出現子ノードを 1 つ以上持つ複数出現親ノードが含まれている場合は、Developer ツールでポートの作成または階層の第 1 レベルのマッピングが行われません。 |

8. 出力ポートを追加し、そこにノードをマップするには、次のいずれかの方法を使用します。

| オプション | 説明 |
|-------------------------|--|
| ノードをドラッグする | 操作入力のノードから、 【ポート】 領域内のグループ名またはポートにポインタをドラッグします。 |
| 【マップ】 ボタンをクリックする | 【操作入力】 領域でノードを 1 つ以上選択します。 【ポート】 領域で抽出先を選択します。 【マップ】 ボタンをクリックします。 |
| ポートをコピーする | 別のトランスフォーメーションからポートを選択し、 【操作入力】 領域にコピーします。ポートをコピーするには、キーボードショートカットを使用するか、または Developer ツール 【コピー】 ボタンと 【貼り付け】 ボタンを使用します。 |

| オプション | 説明 |
|------------------------------|--|
| 手動でポートを追加する | 【新規】 > 【フィールド】 をクリックしてポートを追加します。 |
| 【階層の第 1 レベルの抽出】 を選択する | 【階層の第 1 レベルの抽出】 を選択します。Developer ツールにより、操作入力 of 第 1 レベルのノードが出力ポートおよび出力グループにマップされます。また、マッピングを実行するために必要な出力ポートと出力グループが作成されます。階層の第 1 レベルに、複数出現子ノードを 1 つ以上持つ複数出現親ノードが含まれている場合は、Developer ツールでポートの作成または階層の第 1 レベルのマッピングが行われません。 |

9. 以下のいずれかの方法を使用して、ポートの場所をクリアします。

| オプション | 説明 |
|---------------------------|---|
| 【クリア】 ボタンをクリックします。 | 【ポート】 領域でポートを 1 つ以上選択し、 【クリア】 をクリックします。 |
| ノードとポートを結ぶ線を削除する | 操作入力のノードを出力ポートに接続する線を 1 つ以上選択し、Delete キーを押します。 |

10. SOAP リクエスト全体を XML としてマップするには、**【操作入力】** 領域で **【要求】** ノードを右クリックし、**【XML としてマップ】** を選択します。

出力トランスフォーメーション

出力トランスフォーメーションは、Web サービス WSDL での出力要素およびヘッダー要素を表します。出力トランスフォーメーションは、操作マッピングでリレーショナルデータのグループから SOAP レスポンスメッセージを作成します。WSDL で大きな SOAP メッセージ階層が記述されていても、Web サービス操作からは SOAP レスポンスメッセージの一部分のデータが返される可能性があります。

Web サービスの操作出力を定義すると、Developer ツールによって出力トランスフォーメーションが作成されます。

出力トランスフォーメーションの **【ポート】** タブを使用して、操作出力階層の表示、トランスフォーメーション入力ポートの追加、および操作出力への入力ポートのマッピングを行います。操作出力階層によって、SOAP レスポンスメッセージ階層が定義されます。

1 つの文字列またはテキストの入力ポートから SOAP レスポンス全体に、XML データをマップできます。XML データを SOAP レスポンス全体にマップする場合は、操作出力のノードにポートをマップできません。

出力トランスフォーメーションの詳細プロパティは、**【詳細】** タブで設定します。

出力トランスフォーメーションの **【ポート】** タブ

【ポート】 タブでは、入力グループの定義、入力ポートの定義、および操作出力ノードへの入力ポートのマッピングを行います。

ポートを表示しているときは、グループおよびポートを手動で追加するか、または他のトランスフォーメーションから出力トランスフォーメーションにポートをコピーできます。その場合、キーボードショートカットを使用するか、または Developer ツールでコピーと貼り付けのボタンを使用します。

出力マッピングを表示しているときは、入力グループの定義、入力ポートの定義、および操作出力階層への入力ポートのマッピングを行うことができます。タブの左側は【ポート】領域、タブの右側は【操作出力】領域です。【操作出力】領域には、SOAP レスポンスメッセージ階層が表示されます。【ポート】領域では、入力グループおよび入力ポートを定義できます。【ポート】領域から【操作出力】領域のノードに入力ポートをマップすると、【操作出力】領域の【場所】カラムに入力ポートの場所が表示されます。

階層の第 1 レベルをマップすることを選択すると、Developer ツールによって操作出力の第 1 レベルのノードに入力ポートがマップされます。また、データをマップするために必要な入力ポートが作成されます。階層の第 1 レベルに、複数出現子ノードを 1 つ以上持つ複数出現親ノードが含まれている場合は、Developer ツールでポートの作成または階層の第 1 レベルのマッピングが行われません。

出力マッピングを表示する場合、入力ポートと操作入力内のノードを結ぶ線を表示するように選択できます。

出力トランスフォーメーションの【詳細】タブ

【詳細】タブでは、出力トランスフォーメーションの詳細プロパティを設定します。

以下の表に、出力トランスフォーメーションの【詳細】タブで設定できるプロパティを示します。

| プロパティ | 説明 |
|---------|--|
| ソート済み入力 | 入力データを必ずしもすべて処理しなくても、Data Integration Service が出力を生成できるようにします。入力データが操作入力階層内のキーを基準にしてソートされる場合に、ソート済み入力を有効にします。デフォルトでは無効になっています。 |

ポートを操作出力にマップするためのルールとガイドライン

出力トランスフォーメーションを設定する場合は、ポートを操作出力階層にマップします。

入力ポートを操作出力階層にマップする際には、以下のルールとガイドラインを考慮します。

- 入力ポートおよびノードのデータ型に互換性があることが必要です。
- 階層内の 1 個のノードに対して、1 個の入力ポートをマップできます。
- 1 つの入力グループから操作出力の同じ階層レベルにあるノードに、ポートをマップできます。
- 1 つの入力グループから操作出力の異なる階層レベルにあるノードに、異なるポートをマップできます。
- 操作出力のキーに入力ポートをマップします。キーにマップするポートのデータ型は、string、integer、または bigint であることが必要です。操作出力内の、SOAP メッセージに含める階層レベルより上位にある全レベルのキーに、データをマップします。マップするレベルと、その上位にある全レベルの外部キーを含めます。

注: 操作出力階層の最も低いレベルのみをマップする場合は、入力ポートをキーにマップする必要はありません。

- データ型が異なる複数の入力ポートをキーにマップできます。キーの【場所】フィールドをクリックすると、入力ポートの順序を変更したりポートの 1 つを削除したりすることができます。

出力トランスフォーメーションの設定

Web サービス操作から応答メッセージが返される場合は、出力トランスフォーメーションを設定します。各トランスフォーメーションで入力ポートを定義し、それらの入力ポートから操作出力階層のノードにデータをマップします。

1. エディタで出力トランスフォーメーションを選択します。
2. 【プロパティ】ビューの【ポート】タブをクリックします。

3. **【出力マッピング】** をクリックします。
【操作出力】 領域に操作出力階層が表示されます。入力ポートは **【ポート】** 領域で定義します。
4. 必要に応じて、**【行の表示】** をクリックして、入力ポートを操作出力のノードに接続する線を表示します。すべての線を表示することも、選択したポートの線を表示することもできます。
5. 操作出力に anyType 要素、any 要素、anyAttribute 属性、派生型要素、または置き換えグループが含まれている場合は、**【操作出力】** 領域でオブジェクトを選択します。ノードの **【タイプ】** カラムで **【選択】** をクリックし、リストから 1 つ以上のタイプ、要素、または属性を選択します。
6. 入力グループを追加するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|------------------------------|--|
| ノードをドラッグする | 【操作出力】 領域のグループノードまたは子ノードから、 【ポート】 領域の空のカラムまでポインタをドラッグします。グループノードの場合、Developer ツールによってポートのないグループが追加されます。 |
| 手動でグループを追加する | 【新規】 ボタンの横にある矢印をクリックし、 【新しいグループ】 をクリックします。 |
| 【階層の第 1 レベルの抽出】 を選択する | 【階層の第 1 レベルの抽出】 を選択します。Developer ツールにより、操作入力力の第 1 レベルのノードが出力ポートおよび出力グループにマップされます。また、データをマップするために必要な入力ポートと入力グループが作成されます。 |

7. 入力ポートを追加するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|------------------------------|--|
| 手動でポートを追加する | 【新規】 ボタンの横にある矢印をクリックし、 【新しいポート】 をクリックします。 |
| 別のトランスフォーメーションからポートをドラッグする | エディタで、別のトランスフォーメーションから出力トランスフォーメーションにポートをドラッグします。 |
| ポートをコピーする | 別のトランスフォーメーションからポートを選択し、 【操作出力】 領域にコピーします。ポートをコピーするには、キーボードショートカットを使用するか、または Developer ツール 【コピー】 ボタンと 【貼り付け】 ボタンを使用します。 |
| 【階層の第 1 レベルの抽出】 を選択する | 【階層の第 1 レベルの抽出】 を選択します。Developer ツールにより、操作出力の第 1 レベルのノードが入力ポートおよび入力グループにマップされます。また、マッピングを実行するために必要な入力ポートと入力グループが作成されます。 |

8. 入力ポートから操作出力階層のノードにデータをマップするには、各入力ポートまたはグループから操作出力の関連するノードにポインタをドラッグします。**【操作出力】** 領域のノードの横に、入力フィールドの場所が表示されます。
9. 入力ポートから SOAP レスポンス全体に XML データをマップするには、ポートを右クリックし、**【XML としてマップ】** を選択します。

10. 次のいずれかの方法を使用して、入力ポートを複合キーとしてマップします。

| オプション | 説明 |
|-------------------------------|---------------------------------------|
| 入力ポートをドラッグする | 2 つ以上の入力ポートを選択し、操作出力階層内のキーにドラッグします。 |
| 【場所の選択】 ダイアログボックスから入力ポートを選択する | 操作出力階層でキーの【場所】 カラムをクリックし、入力ポートを選択します。 |

11. 以下のいずれかの方法を使用して、ノードの場所をクリアします。

| オプション | 説明 |
|--------------------|--|
| 【クリア】 ボタンをクリックします。 | 【操作出力】 領域で 1 つ以上のノードを選択し、【クリア】 をクリックします。 |
| ポートとノードを結ぶ線を削除する | 入力ポートを操作出力のノードに接続する線を 1 つ以上選択し、Delete キーを押します。 |

フォールトトランスフォーメーション

フォールトトランスフォーメーションは、Web サービス WSDL でのフォールト要素を表します。フォールトトランスフォーメーションは、Web サービス操作内にユーザー定義フォールトを生成します。

フォールトトランスフォーメーションは、次のいずれかのユーザー定義フォールトを表します。

定義済みフォールト

定義済みフォールトでは、フォールトトランスフォーメーションが、Web サービス WSDL のフォールト要素を表します。また、Web サービス操作マッピングでリレーショナルデータからエラーメッセージを作成します。

汎用フォールト

汎用フォールトでは、フォールトトランスフォーメーションは、Web サービス操作の Web サービス WSDL で定義されたフォールト要素を表しません。汎用フォールトのフォールトトランスフォーメーションは、トランスフォーメーションでエラーが発生すると汎用エラーメッセージを返します。汎用フォールトのフォールトトランスフォーメーションは、Developer ツールから作成および設定できます。

操作マッピングには、複数のフォールトトランスフォーメーションを含めることができます。マッピングで同じフォールトトランスフォーメーションの複数のインスタンスを追加して、マッピングのさまざまな部分で同じメッセージを生成することができます。

マッピングに対しては、操作シグネチャを変更することなく、フォールトトランスフォーメーションを追加または削除することができます。マッピング実行時にフォールトエラーが発生しないことがわかっている場合は、マッピングからフォールトトランスフォーメーションを削除できます。フォールトトランスフォーメーションは、アップストリームトランスフォーメーションに接続する必要があります。そうしないとマッピングは有効となりません。

SOAP 1.1 バインディングを持つ操作でフォールトを作成すると、ウィザードによって faultcode 要素、faultstring 要素、および faultactor 要素が作成されます。SOAP 1.2 バインディングを持つ操作でフォールトを作成すると、ウィザードによって code 要素、reason 要素、node 要素、および role 要素が作成されます。フォールトに要素を追加すると、ウィザードによってフォールトの詳細グループに要素が追加されます。フォールトトランスフォーメーションの【ポート】 タブを使用して、操作フォールト階層の表示、トランスフォー

メーション入力ポートの追加、および操作フォールトへの入力ポートのマッピングを行います。操作フォールト階層によって、ユーザー定義フォールトから生じるエラーメッセージの SOAP レスponseメッセージ階層が定義されます。

1つの文字列またはテキストの入力ポートから SOAP レスponse全体に、XML データをマップできます。SOAP レスponse全体に XML データをマップする場合は、操作フォールトのノードにポートをマップできません。

フォールトトランスフォーメーションの詳細プロパティは、**【詳細】** タブで設定します。

フォールトトランスフォーメーションの **【ポート】** タブ

【ポート】 タブでは、入力グループの定義、入力ポートの定義、および操作フォールトノードへの入力ポートのマッピングを行います。

ポートを表示しているときは、手動でグループとポートを定義できます。あるいは、他のトランスフォーメーションからフォールトトランスフォーメーションにポートをコピーできます。その場合、キーボードショートカットを使用するか、または Developer ツールでコピーと貼り付けのボタンを使用します。

フォールトマッピングを表示しているときは、入力グループの定義、入力ポートの定義、および操作フォールト階層への入力ポートのマッピングを行うことができます。タブの左側は **【ポート】** 領域、タブの右側は **【操作フォールト】** 領域です。 **【操作フォールト】** 領域には、SOAP レスponseメッセージ階層が表示されます。 **【ポート】** 領域から **【操作フォールト】** 領域のノードに入力ポートをマップすると、 **【操作フォールト】** 領域の **【場所】** カラムに入力ポートの場所が表示されます。

階層の第 1 レベルをマップすることを選択すると、Developer ツールによって操作フォールトの第 1 レベルのノードに入力ポートがマップされます。また、データをマップするために必要なポートが作成されます。

フォールトトランスフォーメーションの **【詳細】** タブ

【詳細】 タブでは、フォールトトランスフォーメーションの詳細プロパティを設定します。

以下の表に、**【詳細】** タブで設定できるプロパティを示します。

| プロパティ | 説明 |
|---------|--|
| ソート済み入力 | Data Integration Service が入力データをすべて処理しなくても出力を生成できるようにします。操作フォールト階層で入力データがキーでソートされている場合は、ソート済み入力を有効にします。デフォルトでは無効になっています。 |

ポートを操作フォールトにマップするためのルールとガイドライン

フォールトトランスフォーメーションを設定する場合は、入力ポートを操作フォールト階層にマップします。

入力ポートを操作フォールト階層にマップする際には、以下のルールとガイドラインを考慮します。

- 入力ポートは、操作フォールト階層の 1 つのノードにマップできます。入力ポートおよびノードのデータ型に互換性があることが必要です。
- 1 つの入力グループから操作フォールトの同じ階層レベルにあるノードに、ポートをマップできます。
- 1 つの入力グループから操作フォールトの異なる階層レベルにあるノードに、異なるポートをマップできます。
- 入力ポートデータは、操作フォールト階層内のキーにマップする必要があります。キーにマップするポートのデータ型は、string、integer、または bigint であることが必要です。SOAP メッセージに組み込む階層レベルより上にある操作フォールトのすべてのレベルのキーに、データをマッピングします。マップするレベルと、その上位にある全レベルの外部キーを含めます。

- データ型が異なる複数の入力ポートをキーにマップできます。キーの【場所】フィールドをクリックすると、入力ポートの順序を変更したりポートの1つを削除したりすることができます。

フォールトのトランスフォーメーションの作成

汎用フォールトまたは定義済みフォールトを作成できます。汎用フォールトの場合、Web サービス WSDL はフォールト要素を定義しません。定義済みフォールトの場合、Web サービスはフォールト要素を使用してフォールトを定義します。

1. Web サービスを開きます。
2. 【アウトライン】ビューで、操作マッピングを選択します。
操作マッピングがエディタに表示されます。
3. エディタで【トランスフォーメーションの追加】を選択します。
【トランスフォーメーションの追加】ダイアログボックスが表示されます。
4. 【フォールト】を選択し、【OK】をクリックします。
【フォールトの追加】ダイアログボックスが表示されます。
5. フォールトトランスフォーメーションを作成するには、以下のいずれかの手順を実行します。
 - 【汎用フォールトとして作成】を選択します。
 - 【フォールト要素から定義済みフォールトとして作成】を選択します。
注: Web サービスにはフォールトを定義する要素が含まれます。
6. 【OK】をクリックします。
フォールトトランスフォーメーションが、汎用フォールトまたは定義済みフォールトとして表示されます。

フォールトトランスフォーメーションの設定

Web サービスの操作がフォールトを返すか、汎用フォールトのフォールトトランスフォーメーションを作成する場合、フォールトトランスフォーメーションをそれぞれ設定します。入力ポートを定義し、入力ポートから操作フォールトのノードにデータをマップします。

1. エディタでフォールトトランスフォーメーションを選択します。
2. 【プロパティ】ビューの【ポート】タブをクリックします。
3. 【フォールトマッピング】をクリックします。
【操作フォールト】領域に、応答またはフォールトメッセージ階層が表示されます。入力ポートは【ポート】領域で定義します。
4. 必要に応じて、【行の表示】をクリックして、入力ポートを操作出力のノードに接続する線を表示します。すべての線を表示することも、選択したポートの線を表示することもできます。
5. 操作フォールトに anyType 要素、any 要素、anyAttribute 属性、派生型要素、または置き換えグループが含まれている場合は、【操作フォールト】領域でオブジェクトを選択します。ノードの【タイプ】コラムで【選択】をクリックし、リストから1つ以上のタイプ、要素、または属性を選択します。

6. 入力グループを追加するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|----------------------|--|
| ノードをドラッグする | 【操作フォールト】領域のグループノードまたは子ノードから、【ポート】領域の空のカラムまでポインタをドラッグします。グループノードの場合、Developer ツールによってポートのないグループが追加されます。 |
| 手動でグループを追加する | 【新規】ボタンの横にある矢印をクリックし、【新しいグループ】をクリックします。 |
| 【階層の第 1 レベルの抽出】を選択する | 【階層の第 1 レベルの抽出】を選択します。Developer ツールにより、操作フォールトの第 1 レベルのノードが入力ポートおよび入力グループにマップされます。また、データをマップするために必要な入力ポートと入力グループが作成されます。 |

7. 入力ポートを追加するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|----------------------------|--|
| 手動でポートを追加する | 【新規】ボタンの横にある矢印をクリックし、【新しいポート】をクリックします。 |
| 他のトランスフォーメーションからポートをドラッグする | エディタで、別のトランスフォーメーションからフォールトトランスフォーメーションにポートをドラッグします。 |
| ポートをコピーする | 別のトランスフォーメーションからポートを選択し、【操作フォールト】領域にコピーします。ポートをコピーするには、キーボードショートカットを使用するか、または Developer ツール【コピー】ボタンと【貼り付け】ボタンを使用します。 |
| 【階層の第 1 レベルの抽出】を選択する | 【階層の第 1 レベルの抽出】を選択します。Developer ツールにより、操作フォールトの第 1 レベルのノードが入力ポートおよび入力グループにマップされます。また、データをマップするために必要な入力ポートと入力グループが作成されます。 |

8. 入力ポートから操作フォールト階層のノードにデータをマップするには、各入力ポートまたはグループから操作フォールトの関連するノードにポインタをドラッグします。
【操作フォールト】領域のノードの横に、入力ポートの場所が表示されます。
9. 入力ポートから SOAP レスポンス全体に XML データをマップするには、ポートを右クリックし、【XML としてマップ】を選択します。
10. 次のいずれかの方法を使用して、入力ポートを複合キーとしてマップします。

| オプション | 説明 |
|------------------------------|---|
| 入力ポートをドラッグする | 2 つ以上の入力ポートを選択し、操作フォールト階層内のキーにドラッグします。 |
| 【場所の選択】ダイアログボックスから入力ポートを選択する | 操作フォールト階層でキーの【場所】カラムをクリックし、入力ポートを選択します。 |

11. 以下のいずれかの方法を使用して、ノードの場所をクリアします。

| オプション | 説明 |
|--------------------|---|
| [クリア] ボタンをクリックします。 | [操作フォールト] 領域で 1 つ以上のノードを選択し、[クリア] をクリックします。 |
| ポートとノードを結ぶ線を削除する | 入力ポートを操作フォールトのノードに接続する線を 1 つ以上選択し、Delete キーを押します。 |

フォールト処理

Web サービスでエラーが発生すると、Data Integration Service はエラーメッセージを生成し、そのメッセージをフォールト内で Web サービスクライアントに返します。

SOAP 1.1 リクエストが SOAP 1.2 バインディングで操作に送られると、Web サービスは SOAP 1.1 を使用してフォールトを生成します。SOAP 1.2 リクエストが SOAP 1.1 バインディングで操作に送られると、Web サービスは SOAP 1.2 を使用してフォールトを生成します。Web サービスはシステム定義フォールトとユーザー定義フォールトを生成できます。

Data Integration Service は、ユーザー定義エラーが発生すると Web サービスクライアントにエラーメッセージを返します。ユーザー定義フォールトは接続は以下のいずれかのタイプにすることができます。

- 定義済み
- 汎用

操作マッピングに出力トランスフォーメーションが含まれている場合、Web サービスは出力トランスフォーメーションからデータを返すか、フォールトを返します。操作マッピングでターゲットトランスフォーメーションまたは外部アプリケーションにデータをコミットした後にフォールトが発生した場合、Data Integration Service はデータをロールバックできません。マッピングは停止し、Data Integration Service は Web サービスの出力トランスフォーメーションが受信したデータを破棄します。

SOAP 1.1 フォルト

SOAP 1.1 の場合、フォルトは以下の構造の SOAP メッセージです。

```
Fault (FaultName)
Key_Fault (FaultName)
faultcode      xs:QName
faultstring    xs:string
faultactor     xs:anyURI
detail
```

フォルトには以下の要素が含まれます。

フォールトコード

エラーメッセージ番号などのフォールト ID コード。

フォールト文字列

エラーの説明。

フォールトアクタ

フォールト発生の原因となったオブジェクトのオプションの情報。

詳細

フォールトによって異なるオプションの情報。

SOAP 1.2 フォルト

SOAP 1.2 の場合、フォールトは以下の構造の SOAP メッセージです。

```
Fault (FaultName)
Key_Fault (FaultName)
Code                tns:faultcodeEnum
Reason              tns:reasonText
Node                xs:anyURI
Role                xs:anyURI
detail
```

フォールトには以下の要素が含まれます。

コード

フォルト ID。

コードの Value 要素は、次のいずれかの値である必要があります。

- infasoapns:DataEncodingUnknown
- infasoapns:MustUnderstand
- infasoapns:Receiver
- infasoapns:Sender
- infasoapns:VersionMismatch

注: Code フォルト要素を 1 レベル拡張して、SubCode フォルト要素を抽出できます。SubCode のスキーマタイプは xsd:QName です。SubCode フォルト要素を使用してエラーメッセージ番号を定義できます。

理由

エラーの説明。

ノード

フォールトを生成した SOAP ノードの URI。

ロール

フォールト発生の原因となったオブジェクトのオプションの情報。

明細

フォールトによって異なるオプションの情報。

システム定義フォールト

Data Integration Service は、システムエラーを検出したときにシステム定義フォールトを生成します。システムエラーが発生すると、Data Integration Service はシステム定義のフォールト内でエラーメッセージを返します。

例えば、数値入力ポートが数値以外のデータを受信した場合、Data Integration Service は以下のフォールトを返す可能性があります。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
```

```

        <faultcode>WSCMN_10022</faultcode>
        <faultstring>[WSCMN_10022] Mapping execution failed:
[com.informatica.platform.ldtm.common.ExecutionException: [MPSVCCMN_10009] The Mapping Service Module
[MappingService] encountered an exception with the following details: [LDTM_0072] [ERROR] XML parsing
component [Input_S2R] message code: [66022], message body: A data conversion error occurred in field
[customerID] of the transformation group [Operation]. The field contains the following data: [1100AA].]</
faultstring>
        <detail>                </detail>
    </soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>

```

注: この例では SOAP 1.1 を使用しています。

フォールトトランスフォーメーションを使用せずに、エラーメッセージを Web サービスクライアントに返すことができます。式トランスフォーメーションで ABORT(msg)関数を呼び出して、フォールトを発生させることができます。ABORT メッセージを呼び出すと、Data Integration Service によってシステム定義フォールトメッセージが生成されます。ABORT 関数に渡すメッセージがシステム定義フォールトのフォールト文字列になります。

定義済みフォールト

定義済みフォールトの場合、Web サービスはフォールト要素を使用してフォールトを定義します。カスタムエラーメッセージを返すようにフォールトトランスフォーメーションを設定します。

Web サービスでフォールトトランスフォーメーションを設定するときに、エラー状態を返す操作マッピングロジックを定義する必要があります。フォールトトランスフォーメーションに渡すエラーメッセージを生成するようにトランスフォーメーションを定義する必要があります。

フォールトトランスフォーメーションを定義するときに、返すデータをフォールトコード、フォールト文字列、およびフォールトアクタで定義します。また、その他の情報を Web サービスクライアントに返すために、フォールトメッセージでその他の要素を追加することもできます。フォールトの要素を定義すると、ウィザードによって要素が detail グループに追加されます。複数出現要素を定義して、detail グループ内の要素のグループ間の階層リレーションを作成することができます。

エラーメッセージの構造が各エラーで同じ場合は、マッピングで同じフォールトトランスフォーメーションを複数回使用できます。構造が異なる場合は、Web サービスクライアントに返すエラーメッセージごとに異なるフォールトトランスフォーメーションを設定できます。

例えば、ルックアップで従業員が見つからない場合にメッセージを Web サービスクライアントに返すようにフォールトトランスフォーメーションを設定します。faultcode 要素にエラー番号を、faultstring 要素にエラーメッセージを渡します。また、Web サービスクライアントに部門 ID と従業員 ID を返す必要があります。フォールトを定義するときに、detail グループに DeptID 要素と EmployeeID 要素を追加します。

Data Integration Service から次のようなフォールトが返されます。

```

<infasoapns:Envelope xmlns:infasoapns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:infawsdlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.informatica.com/dis/ws/Get_Employee_Info_Web_Service">
  <infasoapns:Body>
    <infasoapns:Fault>
      <faultcode>ERR_12345</faultcode>
      <faultstring>Web service failed to retrieve employee information.</faultstring>
      <detail>
        <tns:Employee_Not_Found>
          <tns:DeptID>100</tns:DeptID>
          <tns:EmployeeID>2428</tns:EmployeeID>
        </tns:Employee_Not_Found>
      </detail>
    </infasoapns:Fault>
  </infasoapns:Body>
</infasoapns:Envelope>

```

注: この例では SOAP 1.1 を使用しています。

汎用フォールト

エラーが WSDL のフォールト要素で定義されていない場合に Web サービスにエラーメッセージを返す、汎用フォールトを定義できます。トランスフォーメーションでエラーが発生した際に汎用エラーメッセージを返すように、フォールトトランスフォーメーションを作成します。

Web サービスで汎用フォールトのフォールトトランスフォーメーションを作成するときに、エラー状態を返す操作マッピングロジックを定義する必要があります。

汎用フォールトのフォールトトランスフォーメーションを定義するときに、操作バインディングが SOAP 1.1 形式の場合は、返すデータをフォールトコード、フォールト文字列、およびフォールトアクタ要素で定義します。操作バインディングが SOAP 1.2 形式の場合は、返すデータをコード、理由、ノード、およびロール要素で定義します。オプションで、詳細文字列で返すデータも定義できます。詳細文字列はオプションであり、xsd:any 要素を持ちます。データを詳細文字列に送信する場合、Web サービス操作で定義したいいずれかのポートに詳細要素をマップします。また、その他の情報を Web サービスクライアントに返すために、フォールトメッセージでその他の要素を追加することもできます。

detail グループ内の要素のグループ間の階層リレーションを作成することができます。エラーメッセージの構造が各エラーで同じ場合は、マッピングで同じフォールトトランスフォーメーションを複数回使用できます。Web サービスクライアントに返すエラーメッセージごとに異なるフォールトトランスフォーメーションも作成できます。

注: フォールトトランスフォーメーションを作成した後は、汎用フォールトまたは定義済みフォールトのステータスを変更することはできません。

操作マッピングのテスト

希望する結果が得られたことを確認するには、操作マッピングの出力をプレビューします。マッピングにおけるトランスフォーメーションの出力をプレビューすることもできます。

操作マッピングの出力または操作マッピングのトランスフォーメーションの出力をプレビューするときに Developer ツールによって適用される設定を制御するには、Web サービスの設定を作成します。デフォルトの Web サービスの実行設定を構成するには、**【設定】** ダイアログボックスを使用します。**【実行】** ダイアログボックスを使用して、**【データビューア】** ビューで指定できる Web サービスの設定を作成することもできます。

操作マッピングのテスト

操作マッピングをテストして、操作に対する SOAP レスポンスをプレビューします。操作マッピングの出力またはマッピングでのトランスフォーメーションからの出力をプレビューできます。

ドメインに複数のサービスが設定されている場合は、データをプレビューする前に、デフォルトのデータ統合サービスを選択する必要があります。

1. Web サービスを開きます。
2. **アウトライン**ビューで操作マッピングを選択します。
操作マッピングがエディタに表示されます。
3. **【データビューア】** ビューを選択します。
4. **【入力】** ウィンドウで要求を入力します。
5. 特定のトランスフォーメーションの出力データを表示する場合は、エディタでトランスフォーメーションを選択します。選択しなかった場合は、データビューアを実行したときに操作マッピングの出力が表示されます。

6. **【実行】** をクリックします。
要求の結果が **【出力】** ウィンドウに表示されます。
7. 別の要求で操作マッピングをテストするには、**【リセット】** をクリックし、[4](#) から [6](#) の手順を繰り返します。

ビューをカスタマイズするためのオプション

入力、出力、フォールトの各トランスフォーメーションを設定する際に、SOAP メッセージ階層を変更して **【操作】** 領域にキーを表示することができます。ノードの順序を定義するグループ化構造を表示することもできます。

ビューをカスタマイズするオプションを表示するには、**【操作入力】** 領域、**【操作出力】** 領域、または **【操作フォールト】** 領域で **【ビューのカスタマイズ】** をクリックします。

以下のオプションを有効にすることができます。

シーケンス、選択、およびすべて

要素定義が、すべて、シーケンス、選択のどれであることを示す線を表示します。

全グループ内のすべてのノードを SOAP メッセージ内に含める必要があります。

シーケンスグループ内のノードは、WSDL 内に定義された順序で並んでいる必要があります。

選択グループ内の少なくとも 1 つ以上のノードが、SOAP メッセージに指定されている必要があります。

キー

各階層レベルのキーを表示します。

第 7 章

Web サービス SOAP メッセージの解析

この章では、以下の項目について説明します。

- [Web サービス SOAP メッセージの解析の概要, 64 ページ](#)
- [トランスフォーメーションのユーザーインターフェース, 65 ページ](#)
- [複数出現出力設定, 66 ページ](#)
- [anyType 要素の解析, 68 ページ](#)
- [派生型の解析, 69 ページ](#)
- [QName 要素の解析, 70 ページ](#)
- [代替グループの解析, 70 ページ](#)
- [SOAP メッセージ内の XML 構造の解析, 70 ページ](#)

Web サービス SOAP メッセージの解析の概要

Data Integration Service は、Web サービストランスフォーメーションで SOAP メッセージを解析するときに行データを生成します。

Web サービス入力トランスフォーメーションおよび Web サービスコンシューマトランスフォーメーションは、SOAP メッセージの解析を行う Web サービストランスフォーメーションです。

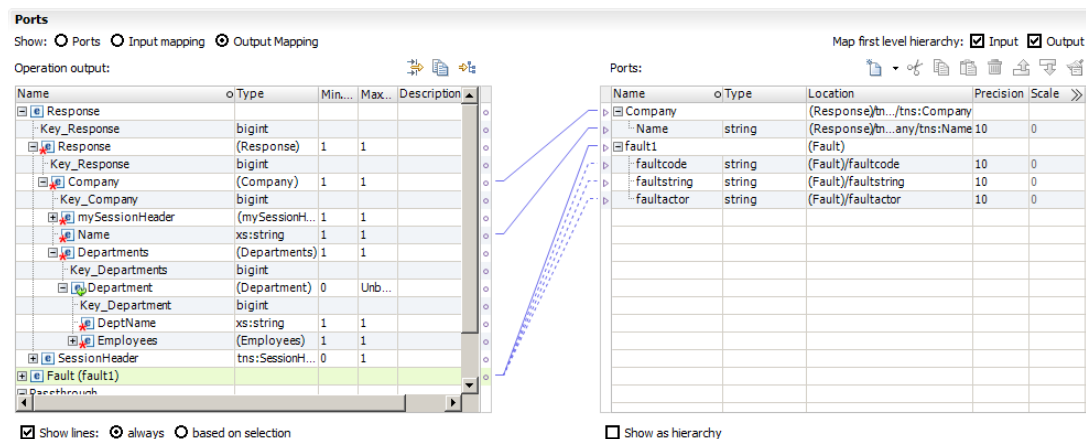
SOAP メッセージを解析するようにトランスフォーメーションを設定するには、SOAP メッセージ階層と同様の構造の出力ポートを作成します。SOAP メッセージ階層内のノードを、作成した出力ポートにマッピングします。

正規化した出力ポートのグループ、非正規化したグループ、ピボット化したポートのグループを設定することができます。SOAP メッセージに派生型、anyType 要素、または代替グループが含まれる場合は、SOAP メッセージインスタンス内に出現する可能性のある型に基づいて別の出力グループを設定することもできます。

トランスフォーメーションのユーザーインターフェース

Web サービスコンシューマトランスフォーメーションおよび Web サービス入力トランスフォーメーションでは、ユーザーインターフェースを介して、SOAP メッセージからトランスフォーメーション出力ポートにデータをマップすることができます。

次の図に、Web サービスコンシューマトランスフォーメーションにおける SOAP 1.1 メッセージノードと出力ポート間のマッピングを示します。



操作領域

操作領域には SOAP メッセージ階層が含まれています。階層構造内の階層レベルは、複合ノードまたは複数出現ノードによって設定されます。Developer ツールは階層レベルにキーを割り当て、そのキーによって階層レベル間の親子リレーションが設定されます。

前の図に示した SOAP メッセージ階層には、次の階層レベルが含まれます。

応答または要求

応答メッセージまたは要求メッセージのルートを表すレベル。

会社名

最上位レベルの要求データ。

部門

その会社内の複数出現の部門。

従業員

従業員は、部門内の複合要素です。

フォールトグループ

エラーメッセージを受信するフォールトメッセージグループ。

ポート領域

SOAP メッセージの階層レベルから出力ポートへ、データをマップすることができます。出力ポート内の各グループは、プライマリ外部キーリレーションによって、別の出力グループに関連付けられている場合もあります。

前の図に示したトランスフォーメーションには、SOAP メッセージ内のノードグループに対応する出力ポートのグループがあります。

複数出現出力設定

入力トランスフォーメーションまたは Web サービスコンシューマトランスフォーメーションから複数出現のデータが返される場合、出力ポートを別の構成で設定することもできます。

正規化した出力データ、ピボット化した出力データ、または非正規化した出力データを構成することができます。

例えば、SOAP メッセージに Departments（部門）と Employees（従業員）という複合要素が含まれているとします。各部門には、複数の従業員が含まれています。Departments は Employees の親です。

SOAP メッセージには、以下のような要素の階層が含まれます。

```
Departments
  Department_ID
  Department_Name
  Employees
    Employee_ID
    Employee_Name
```

正規化したリレーショナル出力

正規化した出力データを作成する場合、それらのデータ値は出力グループ内で反復されません。SOAP メッセージ内の階層レベルと、出力ポートのグループとの間に、1 対 1 リレーションを作成します。

SOAP メッセージに Departments という親階層レベルと、Employees という子階層レベルが含まれている場合に、以下のようなポートのグループを作成するとします。

```
Departments
  Department_Key
  Department_ID
  Department_Name
```

```
Employees
  Department_Key
  Employee_ID
  Employee_Name
```

Department_Key は、Employees 出力グループを Departments 出力グループに関連付ける生成キーです。

生成キー

出力グループを作成すると、その出力グループと別の出力グループとが、Developer ツールの生成キーを使用して関連付けられます。Developer ツールは、親グループと子グループの両方に bigint 値のキーを追加します。生成キーのキー値は、実行時に Data Integration Service によって作成されます。

例

SOAP 階層内に次のノードがあります。

```
Departments
  Dept_Key
  Dept_Num
  Dept_Name

  Employees
    Dept_FK
    Employee_Num
    Employee_Name
```

Departments の出力ポートグループを作成する場合、Departments ノードをポート領域の空のフィールドにマップします。Developer ツールによって、次の出力グループが作成されます。

```
Departments
  Dept_Num
  Dept_Name
```

Employees ノードをポート領域の空のフィールドにマップすると、Developer ツールによって、Employees グループを Departments グループに関連付けるよう求めるメッセージが表示されます。Employees グループを複数のグループに対して関連付けることができます。Developer ツールによって、各グループにキーが追加されます。

Developer ツールによって、以下のグループと生成キーが作成されます。

```
Departments
  Key_Departments
  Dept_Num
  Dept_Name

Employees
  Key_Departments
  Employee_Num
  Employee_Name
```

注: 生成キーにノードをマップする必要はありません。キー値は実行時に Data Integration Service によって作成されます。

Developer ツールは、1 個の出力グループ内の複数のレベルに対して生成キーを作成できます。Employees グループに、以下のようなポートが含まれるとします。

```
Employees
  Key_Employees
  Key_Departments
  Key_Managers
  Employee_Num
  Employee_Name
```

Key_Departments および Key_Managers は、親グループをポイントする生成キーです。Key_Employees は、Employees グループを示す生成キーです。Key_Employees は、子グループが Employees グループに関連付けられている場合に表示されます。

非正規化したリレーショナル出力

リレーショナル出力を非正規化することができます。出力データを非正規化すると、親グループ内の要素の値が、個々の子要素に対して繰り返し出現するようになります。

出力データを非正規化するには、親階層レベルから出力ポートの子グループにノードをマップします。

次の例に、Employees 出力グループ内の Department_ID および Department_Name を示します。

```
Employees
  Department_ID
  Department_Name
  Employee_ID
  Employee_Name
```

Department_ID および Department_Name は、部門内の全従業員に対して繰り返し表示されます。

| Department_ID | Department_Name | Employee_ID | Employee_Name |
|---------------|-----------------|-------------|---------------|
| 100 | 経理 | 56500 | Kathy Jones |
| 100 | 経理 | 56501 | Tom Lyons |
| 100 | 経理 | 56509 | Bob Smith |

ピボット化したリレーショナル出力

出力グループに表示される複数出現要素の数を指定することができます。

複数出現要素をピボット化するには、出力ポートの親グループに複数出現子要素をマップします。Developer ツールによって、親に含める子要素の数を指定するよう求めるメッセージが表示されます。

次の例に、親グループ部門内の 2 つの Employee_ID インスタンスを示します。

```
Departments
  Department_ID
  Department_Name
  Employee_ID1
  Employee_ID2
```

anyType 要素の解析

anyType 要素は、WSDL またはスキーマで使用されているグローバル型すべてを表します。ノードを Developer ツール内のポートにマップする場合、SOAP メッセージ内に anyType 要素の代わりに出現するデータ型を選択します。SOAP メッセージ内の anyType 要素は、複合型または xs:string に置き換える必要があります。選択したデータ型ごとに、ポートグループを作成します。

出力ポートにデータをマップする型を選択する必要があります。WSDL またはスキーマにグローバル型が含まれない場合は、Developer ツールにより anyType 要素が xs:string に置き換えられます。

操作領域内で要素型を選択するには、**[タイプ]** カラムで anyType 要素の **[選択]** をクリックします。利用可能な複合型および xs:string のリストが表示されます。

anyType 要素を派生型で置き換える場合、Data Integration Service は一度に 1 つの型に対して要素を設定します。SOAP メッセージは基本型と派生型のデータを同時に含むことはありません。

派生型の例

WSDL に anyType 要素が 1 つ含まれています。この要素を AddressType とその派生型である USAddressType に置き換えます。SOAP メッセージ階層に以下のグループがあります。

```
Address:AddressType (base type)
  Address: AddressType
    Street
    City

Address:USAddressType (derived type)
  Street
  City
  State
  ZipCode
```

SOAP メッセージには以下のデータが含まれています。

```
<address xsi:type="AddressType">
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>
```

Data Integration Service は xsi: AddressType に対して以下の 1 行を返します。

| Street | City |
|------------------|----------|
| 1002 Mission St. | San Jose |

Data Integration Service は、派生型 xsi: USAddressType に対して以下の 1 行を返します。

| Street | City | 状態 | 郵便番号 |
|-------------------|-----------|----|-------|
| 234 Fremont Blvd. | Sunnyvale | CA | 94556 |

型が xsi: USAddressType である場合、Data Integration Service は AddressType に値を設定しません。

派生型の解析

派生型を含む SOAP メッセージを解析することができます。SOAP メッセージからデータを受け取るポートを定義する場合、SOAP メッセージ内に出現する可能性のある型を選択します。選択した型の要素によって、作成する必要のあるポートが決まります。

例えば、WSDL に AddressType およびその派生型である USAddressType が含まれるとします。Developer ツールの操作領域で、以下のようなグループを作成できます。

```
Address
  Address: AddressType
    Street
    City

  Address:USAddressType
    Street
    City
    State
    ZipCode
```

SOAP メッセージに、以下のようなデータが含まれているとします。

```
<address>
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>

<address xsi:type="USAddressType">
<street>100 Cardinal Way</street>
<city>Redwood City</city>
<zip>94536</zip>
<state>CA</state>
</address>

<address>
<street>100 El Camino Real</street>
<city>Sunnyvale</city>
</address>
```

基本型である Address に対して、Data Integration Service から以下の行が返されます。

| Street | City |
|--------------------|--------------|
| 1002 Mission St. | San Jose |
| 234 Fremont Blvd | Sunnyvale |
| 100 Cardinal Way | Redwood City |
| 100 El Camino Real | Sunnyvale |

派生型である USAddress に対して、Data Integration Service から以下の行が返されます。

| Street | City | 状態 | 郵便番号 |
|-------------------|--------------|----|-------|
| 234 Fremont Blvd. | Sunnyvale | CA | 94556 |
| 100 Cardinal Way | Redwood City | CA | 94536 |

Data Integration Service は、すべての住所を基本型で返します。Data Integration Service は、米国式表記の住所データを派生型で返します。派生型には、USAddressType が基本型から継承した Street 要素と City 要素も含まれています。

QName 要素の解析

Data Integration Service で SOAP メッセージの QName 要素を解析すると、スキーマで定義されている名前空間プレフィックスを使用するためにスキーマの名前空間に属する QName 値が更新されます。そうでない場合は、Data Integration Service により要素の値は更新されません。

例えば、スキーマには名前空間"http://user/test"に対して定義されている名前空間プレフィックス tns があるとします。SOAP メッセージには、同じ名前空間に対して定義されている名前空間プレフィックス mytns があります。Data Integration Service が QName 値 mytns:myelement を解析するとき、値は tns:myElement に変更されます。

Data Integration Service で SOAP メッセージの QName 要素を生成するとき、要素の値は更新されません。

代替グループの解析

代替グループは、ある要素を同じグループ内の別の要素に置き替えます。代替グループは、派生型とほぼ同じですが、各要素定義に代替グループ名が含まれている点で異なります。

代替グループ内の特定の型から要素を受け取る、出力ポートのグループを設定することができます。代替グループ内の別の型から要素を受け取る、別の出力ポートグループを設定することもできます。

SOAP メッセージ内の XML 構造の解析

SOAP メッセージには、choice、list、union などの要素が含まれる場合があります。

Web サービストランスフォーメーションは、これらの構造を持つ SOAP メッセージを解析できますが、いくつかの制限を伴います。

choice 要素

choice 要素の子要素は、<choice>宣言内の要素のいずれかに限定されます。

以下に、従業員または契約社員の person 要素を示します。

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="contractor" type="contractor"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

choice 要素のマッピングは、以下の方法で行うことができます。

- 出力グループ内の個々の choice 要素に対して出力ポートを作成します。一部の要素では、出力行に NULL 値が表示されることがあります。
- 各 choice 要素に対して出力グループを作成します。上の例の場合、employee グループと contractor グループを作成します。Data Integration Service は、SOAP メッセージ内に表示される要素に基づいて行を生成します。

list 要素

list 要素は、"Monday Tuesday Wednesday"などのように、単純型の値が複数含まれている XML 要素です。

Data Integration Service は、list を文字列値として返すことができます。SOAP メッセージにリストが含まれている場合に、リストの項目を別々の出力行にマップすることはできません。マッピング内でリスト内の要素を抽出する必要がある場合には、式フォーメーションを設定することで、リスト内の要素を個別に取り出すことができます。

union 要素

union 要素は、複数の型の組み合わせから成る単純型です。

以下のテキストに、size_no と size_string という 2 つの単純型を組み合わせた、Size 要素を示します。

```
<xs:element name="Size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>
```

Size 要素を出力ポートにマップするには、その Size 要素用のポートを 1 つ作成します。作成したポートを文字列として設定します。マッピング内に別のトランスフォーメーションを設定し、データを別のデータ型に変換することができます。

第 8 章

Web サービス SOAP メッセージの生成

この章では、以下の項目について説明します。

- [Web サービス SOAP メッセージの生成の概要, 72 ページ](#)
- [トランスフォーメーションのユーザーインターフェース, 73 ページ](#)
- [ポートと階層レベルのリレーション, 74 ページ](#)
- [キー, 75 ページ](#)
- [ポートのマップ, 76 ページ](#)
- [複数出現ポートのピボット化, 78 ページ](#)
- [非正規化データのマップ, 79 ページ](#)
- [派生型および要素の置き換え, 81 ページ](#)
- [SOAP メッセージ内の XML 構造の生成, 82 ページ](#)

Web サービス SOAP メッセージの生成の概要

Data Integration Service では、SOAP メッセージを生成する際に、入力データのグループから XML データが生成されます。Web サービスコンシューマトランスフォーメーション、Web サービスの出力トランスフォーメーション、またはフォールトトランスフォーメーションを作成する場合は、SOAP メッセージ階層にマップする入力ポートを設定します。

SOAP メッセージを生成するようにトランスフォーメーションを設定するには、入力ポートのグループを作成し、各グループを SOAP メッセージ階層内のグループにマップします。SOAP メッセージの構造は、WSDL またはスキーマで定義されます。

SOAP メッセージ内のデータのグループを、非正規化入力データから設定することができます。また、複数出現入力データを、SOAP メッセージ内の複数出現ノードにピボット化することもできます。

データは、SOAP メッセージ内の派生型、anyType 要素、または置き換えグループにマップできます。トランスフォーメーションを定義するときに、SOAP メッセージに出現可能な型を選択する必要があります。選択した型によって、作成が必要となる入力ポートが決まります。

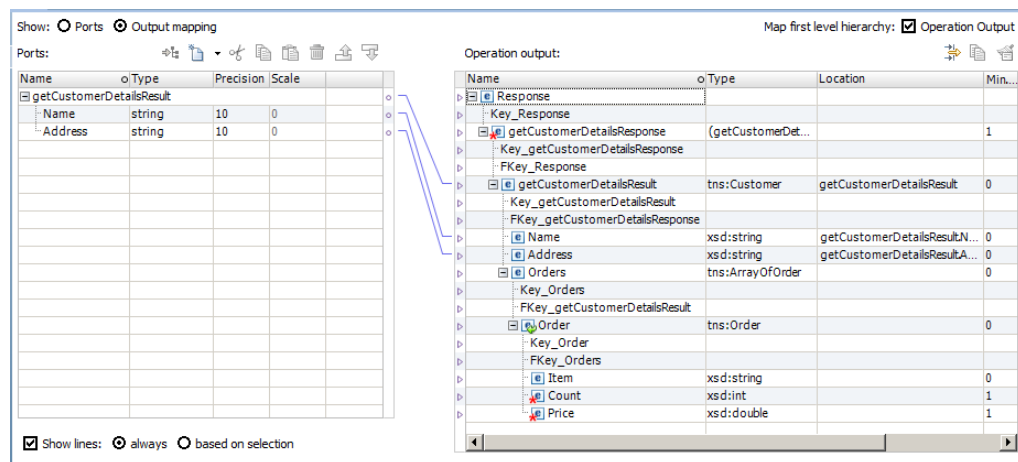
Developer ツールで SOAP メッセージ階層を表示すると、階層にキーが含まれていることがわかります。このキーは、SOAP メッセージには現れません。Data Integration Service では、このキーを使用して、SOAP メッセージ内のグループ間に親子リレーションが定義されます。キー値を設定するには、SOAP メッセージ内のキーに入力データをマップします。

トランスフォーメーションのユーザーインターフェース

Web サービスの出力トランスフォーメーション、フォールトトランスフォーメーション、および Web サービスコンシューマトランスフォーメーションには、SOAP メッセージを設定するために使用できるユーザーインターフェースが含まれています。

SOAP メッセージを生成するようにトランスフォーメーションを設定するには、SOAP メッセージ階層に似た構造内で入力ポートを作成します。階層の構造は、WSDL またはスキーマによって決まります。各入力ポートを SOAP メッセージ内のノードにマッピングします。

以下の図は、Web サービスの出力トランスフォーメーションにおける入力ポートと SOAP メッセージノードの間のマッピングを示しています。



[入力ポート] 領域

入力ポートのグループを【入力ポート】領域で作成します。SOAP メッセージ階層内の各レベルについて、マッピングが必要な入力ポートを組み込みます。

Response または Request 入力グループと、データを受け取る子グループを作成する必要があります。

入力ポートグループを作成する際に、各親グループでプライマリキーを定義します。各子グループで外部キーを定義します。外部キーによって、グループが親グループに関連付けられます。

WSDL ルートレベルでデータを渡すことがなければ、Response レベルまたは WSDL ルートレベルでキーを定義する必要はありません。例えば、ルートレベルに HTTP ヘッダーが含まれている場合があります。

以下に示す顧客と注文のグループに似たポートのグループを作成するとします。

```
Response
  Response_Key

  Customer_Details_Root
    Key_Cust_Det
    FK_Response_Key

    Customer
      Customer_ID
      FK_Cust_Det
      Name
      Address

      Orders
        Order_Num
        FK_Cust_ID
```

```
Order_Items
  Order_Num
  Item
  Count
  Price
```

操作領域

【操作】 領域には、WSDL またはスキーマで定義されている SOAP メッセージ階層内の要素が表示されます。SOAP メッセージに、WSDL またはスキーマの要素がすべて含まれているとは限りません。メッセージには、入力ポートからマップしたデータが含まれています。

複数出現ノードおよび複合ノードによって、SOAP メッセージ構造の階層レベルが定義されます。Developer ツールにより、これらのレベルにキーが追加されてレベル間の親子リレーションが作成されます。リーフレベルを除く階層内のすべてのレベルには、プライマリキーが設定されます。各子レベルには、親レベルへの外部キーが設定されます。SOAP メッセージ階層に見られるキーは、SOAP メッセージインスタンスには現れません。Data Integration Service では、SOAP メッセージ生成時にデータのレベルを関連付けるために、キーの値が必要となります。

【場所】 カラムには、グループ名および SOAP メッセージ内の要素のデータを含んでいる入力ポートが表示されます。入力ポートをノードにマップするまで、**【場所】** カラムは空白です。

前出の図では、SOAP メッセージに顧客詳細および注文の単一インスタンスが含まれています。Orders グループには、Order という複数出現要素が含まれています。SOAP メッセージ階層では、以下のレベルがキーで関連付けられています。

```
Response
  GetCustomerDetailsResponse
    GetCustomerDetailsResult
      Orders
        Order
```

Response レベルは、レスポンスメッセージのルートを表します。Data Integration Service では、SOAP メッセージにヘッダーを添付するためにこのレベルが必要となります。

GetCustomerDetailsResponse レベルは、メッセージのルートです。

ポートと階層レベルのリレーション

入力ポートを SOAP メッセージ階層にマップする場合は、入力グループと SOAP メッセージ階層レベルの間のリレーションを維持します。例えば、Department（部門）と Employee（従業員）の 2 つのグループがあるとしたします。

Department 入力グループは、以下の行を受け取ります。

| Dept_num | 名前 | 場所 |
|----------|---------|------------|
| 101 | HR | New York |
| 102 | Product | California |

Employee 入力グループは、以下の行を受け取ります。

| Dept_num | Employee |
|----------|----------|
| 101 | Alice |
| 101 | Bob |
| 102 | Carol |
| 102 | Dave |

Employee グループの部門番号を、Department グループと Employee グループの間にリレーションを確立する外部キーとしてマップします。部門番号は、部門階層レベルに出現しますが、従業員レベルでは出現しません。

SOAP メッセージには、以下の XML 構造が格納されます。

```
<department>
  <dept_num>101</dept_num>
  <name>HR</name>
  <location>New York</location>

  <employee>
    <name>Alice</name>
  </employee>

  <employee>
    <name>Bob</name>
  </employee>
</department>

<department>
  <dept_num>102</dept_num>
  <name>Product</name>
  <location>California</location>

  <employee>
    <name>Carol</name>
  </employee>

  <employee>
    <name>Dave</name>
  </employee>
</department>
```

キー

SOAP メッセージ階層にはキーが含まれています。Data Integration Service では、SOAP メッセージ内で XML 階層を構築するために、キー値が必要となります。

入力ポートデータを、SOAP メッセージ階層内のキーにマップする必要があります。データを提供する各レベルのキーに、データをマッピングします。複数出現ノードがある場合は、ノードを親に関連付ける必要があります。

SOAP メッセージ内では、キーが型のない状態で出現します。キーにマッピングするポートのデータ型は、string、integer、または bigint であることが必要です。親グループのプライマリキーと各子グループの外部キーは、データ型、精度、および位取りが同じである必要があります。生成されたキーを SOAP メッセージキーにマップできます。

ポートは、ノードおよび同じ階層レベルにあるキーにマップできます。例えば、Employee_ID を SOAP メッセージ内のノードにマップし、さらに Employee レベルのキーにマップします。

階層内の 2 つのグループノードに親子リレーションがある場合は、以下のタスクを実行します。

- 親ノードグループのプライマリキーにポートをマップします。
- 子ノードグループの外部キーにポートをマップします。

プライマリキーを入力ポートにマップして、プライマリキーが Null である行またはプライマリキーが重複している行を削除することもできます。

同じキーに複数のポートをマップすることで、SOAP メッセージ内に複合キーを作成できます。複合キーは、データを非正規化し、かつ複数出現する値の組み合わせに対して一意のキーを維持する必要がある場合に使用します。string 値、bigint 値、または integer 値を含む複合キーを作成できます。

注: 操作マッピングに式トランスフォーメーションを組み込んで、キー値を生成することができます。

複合キーの例

以下に示すポートのグループから、一意の事業部 (Division) -部門 (Department) キーを設定します。

```
Company
  Company_Num
  Company_Name

  Division
    Company_Num
    Divison_Num
    Division_Name

    Department
      Division_Num
      Dept_Num
      Dept_Name
      Location
```

Dept_Num は事業部内では一意ですが、会社内の全事業部で一意ではありません。

事業部と部門の情報を含んだ Department グループを設定することができます。事業部番号と部門番号を複合キーの一部として設定します。

```
Department
  Division_Num + Dept_Num (key)
  Dept_Name
  Location
```

ポートをマップした順序によって、キー値が決まります。

ポートのマップ

入力ポートを作成したら、各入力ポートを SOAP メッセージ階層にマップします。ポートの場所は、**[操作]** 領域でノードの横に表示されます。

ポートは以下のタイプのノードにマップできます。

アトミックノード

子がなく分割不可能な単純要素または属性。

複数出現アトミックノード

階層内の同じ場所で複数回出現する単純要素または属性。

複合ノード

他の要素を含んでいる要素。

親ノードに場所がない場合は、入力グループ名が親ノードの場所として受け取られます。親ノードに場所がある場合は、階層レベル内の各ノードが同じ場所から出力場所を得る必要があります。

入力グループ名を、階層レベルの親ノードにマップできます。Developer ツールにより、階層内の親ノードの場所フィールドが更新されます。階層内でグループに属している子ノードは更新されません。入力ポートを子ノードにマップする場合は、各入力ポートが親ノードと同じ場所であることが必要です。

入力グループを階層レベルにマッピングした後に、入力グループを変更することができます。【クリア】をクリックするか、【ポート】領域と【操作】領域の間の線を削除することができます。線を削除するには、線のポインタをドラッグして対象の線を選択します。【削除】をクリックします。

ポートのマップ

ポートを SOAP メッセージ内のノードにマップする場合は、ポートをマップするノードのタイプによって、Developer ツールでの結果が変わります。

以下の表に、【操作】領域で単一ポートを各種のターゲットノードにマップした場合の結果を示します。

| ターゲットノード | 結果 |
|------------------------------|---|
| アトミックノード | 単一ポートをノードおよび場所を持たない親ノードにマップした場合、そのノードはポートの場所を受け取ります。親ノードの場所には、単一ポートの入力グループの場所が入ります。単一ポートをノードおよび場所を持つ親ノードにマップした場合は、親ノードの場所を変更し、同じレベルにある他の子ノードの場所をクリアすることができます。階層レベルの場所は、ポートのグループ名に変わります。 |
| 複数出現アトミックノードまたはそのノードのプライマリキー | 単一ポートを複数出現アトミックノードにマップした場合は、Developer ツールによってアトミックノードの場所が選択したポートのグループに設定されます。 |
| 複合ノード | 単一ポートを複合ノードにマップした場合は、Developer ツールによって複合ノードの場所がポートを含んでいるグループの場所に設定されます。Developer ツールから、ポートの割り当て先となる単独出現アトミックノードを指定するように求められます。 すべての単独出現アトミックノードに場所がある場合は、複合ノードをマップできません。 |

グループのマップ

入力グループを SOAP メッセージ内のノードにマップする場合は、ポートをマップするノードのタイプによって、Developer ツールでの結果が変わります。

以下の表に、【操作】領域でグループをノードにマップした場合の結果を示します。

| ターゲットノード | 結果 |
|--------------|---|
| アトミックノード | アトミックノードにグループをマップすることはできません。 |
| 複数出現アトミックノード | ノードとプライマリキーの場所を更新するために、入力グループ内のポートを選択するように要求されます。 |
| 複数出現複合ノード | Developer ツールにより、複合ノードの場所がグループの場所に設定されます。 |

複数のポートのマッピング

複数のポートを SOAP メッセージ内のノードにマップする場合は、ポートをマップするノードのタイプによって、Developer ツールでの結果が変わります。同じグループからマッピングを行う場合は、複数のポートを同時にマップできます。

以下の表に、複数ポートをノードにマップした場合のノードの結果を示します。

| ターゲットノード | 結果 |
|--------------|---|
| 単一のアトミックノード | 複数のポートを単一ノードにマップした場合は、 【操作】 領域で複数の単一アトミックノードの場所を更新します。階層のレベルに十分な数の更新対象ノードがない場合は、Developer ツールによって存在するノードにのみポートがマップされます。 |
| 複数出現アトミックノード | 複数のポートを複数出現アトミックノードにマップする場合は、ポートをそのノードの複数の出現箇所にピボット化します。Developer ツールにより、マップしたポートの数に基づいてノードのインスタンスが作成されます。投影したポートの数を表示メッセージが表示されます。 |
| 複数出現複合ノード | 複数のポートを複合ノードにマップする場合は、更新対象となる単独出現アトミックノードを選択する必要があります。ポートをそのノードの複数の出現箇所にピボット化します。Developer ツールにより、マップしたポートの数に基づいてノードのインスタンスが作成されます。 |

複数出現ポートのピボット化

複数の入力ポートを、SOAP メッセージ内の複数出現ノードにマップできます。Developer ツールにより、入力データが SOAP メッセージ内の複数のノードにピボット化されます。

ピボット化する要素の数を変更するには、**【マップオプション】** ダイアログボックスで **【既存のピボット化のオーバーライド】** を選択します。

ピボット化されたポートインスタンスの 1 つを **【ポート】** 領域から削除すると、Developer ツールによってすべてのインスタンスが **【操作】** 領域から削除されます。

ピボット化の例

入力グループに以下の行があるとします。

| Num | 名前 | 場所 | emp_name1 | emp_name2 | emp_name3 |
|-----|----|------------|-----------|-----------|-----------|
| 101 | HR | New York | Alice | Tom | Bob |
| 102 | 製品 | California | Carol | Tim | Dave |

各行には、部門番号と 3 人の従業員名が含まれています。

Employee は、SOAP メッセージ階層内の複数出現ノードです。Employee のすべてのインスタンスを、入力行から SOAP メッセージ階層にマップできます。Employee のすべての出現箇所を選択します。 **【マップ】** をクリックします。 **【マップオプション】** ダイアログボックスで、一覧からノードを選択するように求められます。

Developer ツールにより、SOAP メッセージ階層で Employee ノードが複数の name ノードを含むように変更されます。

```
Department
  num
  name
```

```
location
Employee      (unbounded)
  emp_name1
  emp_name2
  emp_name3
```

SOAP メッセージから以下の階層が返されます。

```
<department>
  <num>101</num>
  <name>HR</name>
  <location>New York</location>
  <employee>
    <emp_name>Alice</name>
  </employee>
  <employee>
    <emp_name>Tom</name>
  </employee>
  <employee>
    <emp_name>Bob</name>
  </employee>
</department>

<department>
  <num>102</num>
  <name>Product</name>
  <location>California</location>
  <employee>
    <emp_name>Carol</name>
  </employee>
  <employee>
    <emp_name>Tim</name>
  </employee>
  <employee>
    <emp_name>Dave</name>
  </employee>
</department>
```

非正規化データのマップ

非正規化データをマップし、SOAP メッセージ内の正規化ノードに渡すことができます。

非正規化データをマップする場合は、1つの入力グループから SOAP メッセージ階層内の複数のノードにデータを渡します。SOAP メッセージ内に、以下のタイプのリレーションに似たグループリレーションを作成することができます。

線形ノードリレーション

ノード A はノード B の親であり、ノード B はノード C の親であり、ノード C はノード D の親です。

階層ノードリレーション

ノード A はノード B の親であり、ノード C の親でもあります。ノード B とノード C の間に関連はありません。

以下の表に、事業部と部門の非正規化データを含んでいる入力行を示します。

| Division | Dept_Num | Dept_Name | Phone | Employee_Num | Employee_Name |
|----------|----------|-------------|-------|--------------|---------------|
| 01 | 100 | 経理 | 3580 | 2110 | Amir |
| 01 | 100 | 経理 | 3580 | 2113 | Robert |
| 01 | 101 | Engineering | 3582 | 2114 | Stan |

| Division | Dept_Num | Dept_Name | Phone | Employee_Num | Employee_Name |
|----------|----------|-------------|-------|--------------|---------------|
| 01 | 101 | Engineering | 3582 | 2115 | Jim |
| 02 | 102 | Facilities | 3583 | 2116 | Jose |

入力データには、一意の従業員番号および名前が含まれています。事業所データと部門データは、同じ部門および事業部に所属する従業員ごとに繰り返されます。

線形グループリレーション

ポートを設定する際に、Division、Department、および Employee に別々のグループを設定できます。Division は Department の親であり、Department は Employee の親です。以下の線形構造でグループを設定できます。

```

Division
  Division_Key
  Division_Num
  Division Name

  Department
    Department_Key
    Division_FKey
    Dept_Num
    Dept_Name
    Phone

    Employee
      Department_Fkey
      Employee_Num
      Employee_Name

```

SOAP メッセージには Division および Department の一意のインスタンスが含まれますが、Division_Num と Dept_Num は入力データ内で繰り返し出現します。Division_Num を Division グループのプライマリキーとして定義します。Dept_Num を Department グループのプライマリキーとして定義します。

階層グループリレーション

Division 親グループおよび Department と Employee の各子グループで構成されるグループ階層を作成することができます。Department と Employee の間には、プライマリキーと外部キーのリレーションがありません。Department と Employee は、Division の子です。これらのグループは、以下の構造で設定することができます。

```

Division
  Division_Key
  Division_Num
  Division_Name

  Department
    Division_FKey
    Dept_Num
    Dept_Name

  Employee
    Division_FKey
    Employee_Num
    Employee_Name

```


派生型および要素の置き換え

入力ポートは、SOAP メッセージ内の派生複合型、anyType 要素、および置き換えグループにマップできます。SOAP メッセージには、基本型および派生型の要素を組み込むことができます。

型リレーションでは、基本型は別の型の派生元となる型です。派生型は、基本型から要素を継承します。拡張複合型は基本型から要素を継承する派生型であり、追加の要素を含んでいます。限定複合型は、基本型からの要素の一部を制限している派生型です。

派生型の生成

WSDL またはスキーマに派生型が含まれている場合は、SOAP メッセージに追加する型を選択する必要があります。

例えば、WSDL で基本型 AddressType が定義されているとします。この WSDL には、AddressTypes から派生した USAddressType と UKAddressType も含まれています。

各型には以下の要素が含まれています。

- AddressType:street、city
- USAddressType (AddressType を拡張したもの):state、zipCode
- UKAddressType (AddressType を拡張したもの):postalCode、country

[操作] 領域で USAddressType を選択すると、Developer ツールによって USAddressType 要素のグループが SOAP メッセージ内に作成されます。このグループには、基本の住所から street と city が取り込まれ、USAddress の state と zipCode が取り込まれます。基本型を拡張した派生型には、必ず基本型の要素が含まれます。

選択可能なすべての派生型を SOAP メッセージ用に選択すると、Developer ツールによって以下のようなグループが SOAP 階層内に作成されます。

```
Address
  Address: Address
    Street
    City

  Address:USAddressType
    Street
    City
    State
    ZipCode

  Address: UKAddressType
    Street
    City
    PostalCode
    Country
```

Address、USAddress、および UKAddress 用に入力ポートグループを定義する必要があります。

anyType 要素および属性の生成

スキーマ要素および属性のなかには、SOAP メッセージに対して Any 型のデータを許可するものがあります。

anyType 要素は、グローバルに認識されるすべての型を表す選択肢です。ポートを SOAP メッセージの anyType 要素にマップする前に、利用可能な複合型または xs:string を選択します。WSDL またはスキーマに複合型が含まれない場合は、Developer ツールにより anyType 要素型が xs:string に置き換えられます。

操作領域内で要素型を選択するには、**[タイプ]** カラムで anyType 要素の**[選択]** をクリックします。利用可能な複合型および xs:string のリストが表示されます。

以下の要素および属性は、Any 型のデータを許可します。

anyType 要素

要素に関連する XML ファイル内の任意のデータ型にすることができます。

anySimpleType 要素

要素に関連する XML ファイル内の任意の simpleType にすることができます。

ANY 内容要素

要素をスキーマで定義されている任意のグローバル要素にすることができます。

anyAttribute 属性

要素をスキーマで定義されている任意の属性にすることができます。

置き換えグループの生成

置き換えグループを使用して、SOAP メッセージ内のある要素を別の要素に置き換えることができます。置き換えグループは派生型とほぼ同じように機能しますが、要素定義に置き換えグループ名が組み込まれる点が異なります。

例えば、基本型 Address および派生型 USAddress と UKAddress があるとします。

```
xs:element name="Address" type="xs:string"/>
<xs:element name="USAddress" substitutionGroup="Address"/>
<xs:element name="UKAddress" substitutionGroup="Address"/>
```

SOAP メッセージ階層を設定する際に、SOAP メッセージ内で Address に置き換わる要素を選択することができます。

SOAP メッセージ内の XML 構造の生成

WSDL またはスキーマには、choice 要素、list 要素、または union 要素が含まれている場合があります。Web サービスのトランスフォーメーションで、これらの要素を含んだ SOAP メッセージを生成することができます。

choice 要素

choice 要素の子要素は、<choice>宣言内の要素のいずれかに限定されます。

choice 要素を含んでいる SOAP メッセージにポートをマップするには、choice 構造のすべての要素が含まれている入力グループを 1 つ作成します。例えば、品目の説明が寸法または重量であるとします。

item: description, choice {dimension, weight}

説明が寸法である場合、その説明は、長さ、幅、高さで構成される複合型です。

説明が重量である場合、その説明は単純な文字列型です。

この入力データのカラムと行は以下のとおりです。

| description | length | width | height | weight |
|-------------|--------|-------|--------|--------|
| box | 20cm | 18cm | 15cm | NULL |
| coffee | NULL | NULL | NULL | 500g |

SOAP メッセージには、寸法または重量の説明が入った Item グループが含まれています。

```
Item
  Description
    Dimension
      Length
      Width
      Height
    Weight
```

入力データが NULL 値であると、XML 出力の要素が欠落します。

SOAP メッセージには以下のデータが含まれています。

```
<item>
  <desc>box</desc>
  <dimension>
    <length>20cm</length>
    <width>18cm</width>
    <height>15cm</height>
  </dimension>
</item>

<item>
  <desc>coffee</desc>
  <weight>500g</weight>
</item>
```

list 要素

list は、同じ要素または属性に複数の単純型値を入れることのできる XML 要素です。Data Integration Service では、入力データ内の list が統合された一連のデータとして表されている場合に、list を処理できません。

list 内の各項目が個別の要素である場合（ClassDates1、ClassDates2、ClassDates3 など）、Data Integration Service ではそれらの項目を list として処理できません。SOAP メッセージで list を返す必要がある場合は、式トランスフォーメーションを使用してこれらの項目を文字列へと統合できます。

以下の入力行には、曜日で構成される ClassDates という list 要素が含まれています。

| CourseID | Name | ClassDates |
|-----------|-------------------|-------------|
| Math 1 | Beginning Algebra | Mon Wed Fri |
| History 1 | World History | Tue Thu |

Data Integration Service では、以下の XML 構造の SOAP メッセージが返されます。

```
<class>
  <courseId>Math 1</courseId>
  <name>Beginning Algebra</name>
  <classDates>Mon Wed Fri</classDates>
</class>
<class>
  <courseId>History 1</courseId>
  <name>World History</name>
  <classDates>Tue Thu</classDates>
</class>
```

union 要素

union 要素は、複数の型の組み合わせから成る単純型です。SOAP メッセージに union 要素が含まれている場合は、文字列のデータを含んだ単一の入力ポートをマップする必要があります。

例えば、SOAP メッセージに size という要素が含まれているとします。size は、整数と文字列の union です。

```
<xs:element name="size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>
```

入力行には、説明とサイズが指定された品目が含まれています。品目には、42 などの数値のサイズが指定されている場合があります。あるいは、large、medium、small などの文字列値のサイズが指定されている場合もあります。

以下の表に、数値のサイズと文字列のサイズが含まれている入力行を示します。

| Desc | Size |
|-------|-------|
| shoes | 42 |
| shirt | large |

品目サイズに対して 1 つのポートを作成します。ポートを文字列としてマップします。SOAP メッセージには以下の要素が格納されます。

```
<item>
  <desc>shoes</desc>
  <size>42</size>
</item>

<item>
  <desc>shirt</desc>
  <size>large</size>
</item>
```

第 9 章

Web サービスコンシューマトランスフォーメーション

この章では、以下の項目について説明します。

- [Web サービスコンシューマトランスフォーメーションの概要, 85 ページ](#)
- [WSDL の選択, 88 ページ](#)
- [Web サービスコンシューマトランスフォーメーションのポート, 89 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの入力マッピング, 91 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの出力マッピング, 94 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの詳細プロパティ, 97 ページ](#)
- [フィルタの最適化, 101 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの作成, 103 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの例, 105 ページ](#)

Web サービスコンシューマトランスフォーメーションの概要

Web サービスコンシューマトランスフォーメーションは、Web サービスクライアントとして Web サービスに接続し、データへのアクセスまたはデータの変換を行います。Web サービスコンシューマトランスフォーメーションは複数のグループのトランスフォーメーションです。

Web サービスは、SOAP、WSDL、XML などのオープンスタンダードを使用します。SOAP は、Web サービス用の通信プロトコルです。Web サービスクライアントの要求および Web サービスの応答は、SOAP メッセージです。WSDL は、Web サービス操作のプロトコル、形式、およびシグネチャを記述する XML スキーマです。

Web サービス操作には、情報の要求、データ更新の要求、タスク実行の要求などがあります。例えば、`getCustomerOrders` という Web サービス操作を実行するための SOAP リクエストを Web サービスコンシューマトランスフォーメーションが送信とします。トランスフォーメーションは、リクエストで顧客 ID を渡します。Web サービスは、顧客情報と注文情報を取得し、その情報を SOAP レスポンスでトランスフォーメーションに返します。

Web サービスコンシューマトランスフォーメーションは、WSDL、Web サービス接続、またはエンドポイント URL 入力ポートで定義されたエンドポイント URL を使用して Web サービスに接続します。Web サービス接続で Web サービスのセキュリティを有効にします。

SOAP メッセージ

Web サービスコンシューマトランスフォーメーションは、Web サービスプロバイダと情報をやり取りしたり、Web サービスを要求したりするために、Simple Object Access Protocol (SOAP) を使用します。SOAP は、Web サービスのリクエストメッセージとレスポンスメッセージの形式を定義します。

Web サービスコンシューマトランスフォーメーションでデータを変換すると、トランスフォーメーションによって SOAP リクエストが生成され、Web サービスへの接続が行われます。トランスフォーメーションは、WSDL オブジェクト、Web サービス接続、またはエンドポイント URL 入力ポートで定義されたエンドポイント URL を使用して Web サービスに接続します。SOAP リクエストには、要求された操作を実行するために Web サービス側で必要となる情報が含まれています。その Web サービス操作から、SOAP 応答としてデータがトランスフォーメーションに返されます。トランスフォーメーションは、SOAP レスポンスからデータをマップし、出力ポートでデータを返します。

Web サービスコンシューマトランスフォーメーションは、SOAP メッセージヘッダを ISO-8859-1 でエンコードします。

トランスフォーメーションは、ドキュメント/リテラルのエンコーディングを使用する SOAP メッセージを処理できます。ドキュメント/リテラルのスタイルには SOAP メッセージを記述する XML スキーマが必要です。SOAP メッセージは XML から形成されます。SOAP メッセージに複数出現要素が含まれている場合は、要素のグループによって XML 階層のレベルが形成されます。あるレベルが別のレベル内にネストされている場合、グループは関連しています。

SOAP リクエストメッセージには、階層データを含めることができます。例えば、販売データベースに顧客の注文を追加する要求を Web サービスコンシューマトランスフォーメーションが送信するとします。トランスフォーメーションは、2 つのデータグループを SOAP リクエストメッセージで渡します。一方のグループには顧客の ID と名前が含まれ、もう一方のグループには注文情報が含まれています。注文情報は複数回出現します。

SOAP レスポンスメッセージには、階層データを含めることができます。例えば、Web サービスコンシューマトランスフォーメーションが顧客注文に対する SOAP リクエストを生成するとします。Web サービスは、注文ヘッダーと複数出現の注文詳細要素を、SOAP レスポンスで返します。

WSDL ファイル

WSDL ファイルには、送信側と受信側が交換するデータを認識できるように、Web サービスに渡されるデータが記述されています。Web サービスコンシューマトランスフォーメーションを作成する前に、WSDL ファイルをリポジトリにインポートする必要があります。

WSDL ファイルには、Web サービスコンシューマが正しい形式で要求メッセージを送信できるよう、データに対して実行する操作と、プロトコルやトランスポートへのバインドについて記述されています。WSDL ファイルには、Web サービスに接続するネットワークアドレスが記述されています。

WSDL には、SOAP リクエストメッセージと SOAP レスポンスメッセージのエンコード方法に関する情報が含まれています。SOAP エンコーディングにより SOAP メッセージ本体の形式が決まります。SOAP エンコーディングには、Web サービスが Web サービスコンシューマとの通信に使用する要求と応答のメッセージの形式が記述されています。Web サービス開発者は、さまざまなツールキットを使用して Web サービスを作成できます。ツールキットによって、サポートされる SOAP メッセージのエンコーディング方法が異なります。

Web サービスコンシューマトランスフォーメーションは、ドキュメント/リテラル SOAP エンコーディングスタイルをサポートします。Web サービスコンシューマトランスフォーメーションでは、開発者は WSDL 1.1 を使用できます。MIME、DIME、MTOM メッセージなど、WSDL アタッチメントは使用できません。

操作

Web サービスには、その Web サービスがサポートする各アクションの操作が含まれています。

例えば、顧客名を受信して顧客の詳細を応答で返す `getcustomerid` という操作を Web サービスに含めることができます。操作入力には、顧客名の要素が含まれています。操作出力には、顧客名に基づいた顧客詳細の要素が含まれています。

Web サービスコンシューマトランスフォーメーションを設定するときに、トランスフォーメーションが操作入力にデータをマップする方法と、トランスフォーメーションが操作出力からデータをマップする方法を定義します。トランスフォーメーションでは以下の情報を設定します。

入力マッピング

トランスフォーメーションの入力ポートを Web サービスの操作入力ノードにマップする方法を定義します。操作入力では、SOAP リクエスト内の操作の要素を定義します。

出力マッピング

Web サービスの操作出力ノードをトランスフォーメーションの出力ポートにマップする方法を定義します。操作出力では、SOAP レスポンス内の操作の要素を定義します。

Web サービスのセキュリティ

Web サービス接続で Web サービスのセキュリティを有効にします。次のタイプのセキュリティを設定できます。

Web サービスのセキュリティ

データ統合サービスは、Web サービスプロバイダに SOAP リクエストを送信するときに Web サービスセキュリティヘッダーを含めることができます。Web サービスセキュリティヘッダーには、Web サービスプロバイダがデータ統合サービスを認証できるように、認証情報が含まれています。

Web サービスコンシューマトランスフォーメーションによってユーザー名トークンが提供されます。データ統合サービスは、SOAP リクエスト内に個別のセキュリティ SOAP ヘッダーを作成し、その SOAP リクエストを Web サービスプロバイダに渡します。

Web サービス接続では、以下のタイプの Web サービスセキュリティを使用できます。

- PasswordText。データ統合サービスは WS-Security SOAP ヘッダー内のパスワードを変更しません。
- PasswordDigest。データ統合サービスが、ナンス値とタイムスタンプをパスワードに組み合わせます。データ統合サービスでは、そのパスワードに SHA ハッシュを適用して base64 エンコーディングでエンコードし、エンコードしたパスワードを SOAP ヘッダー内で使用します。

トランスポートレイヤセキュリティ

Secure Sockets Layer (SSL) を使用して TCP/IP のトランスポートレイヤ (TCP レイヤ) の上に実装されるセキュリティ。Web サービスは、Web アドレスとして Hypertext Transfer Protocol over SSL (HTTPS) を使用し、メッセージをセキュリティ保護して転送します。Web サービスコンシューマトランスフォーメーションでは、TLS 1.2、TLS 1.1、TLS 1.0 を使用できます。トランスポートレイヤセキュリティに、HTTP 認証、プロキシサーバー認証、SSL 証明書を使用できます。

SSL 認証

HTTPS プロトコルを介して接続する場合は、SSL 認証を使用できます。

使用できる SSL 認証には、次の種類があります。

- 1 方向 SSL 認証
- 2 方向 SSL 認証

HTTP 認証

HTTP プロトコルを介して接続する場合は、HTTP 認証を使用できます。

使用できる HTTP 認証メソッドは、次のとおりです。

- 基本認証
- ダイジェスト認証
- NT LAN Manager (NTLM) 認証

WSDL の選択

Web サービスコンシューマトランスフォーメーションを作成する前に、WSDL ファイルをモデルリポジトリにインポートする必要があります。WSDL ファイルは、実行する Web サービスの操作シグネチャを定義します。WSDL ファイルをインポートすると、Developer ツールによって、他のトランスフォーメーションで再利用可能な物理データオブジェクトが作成されます。

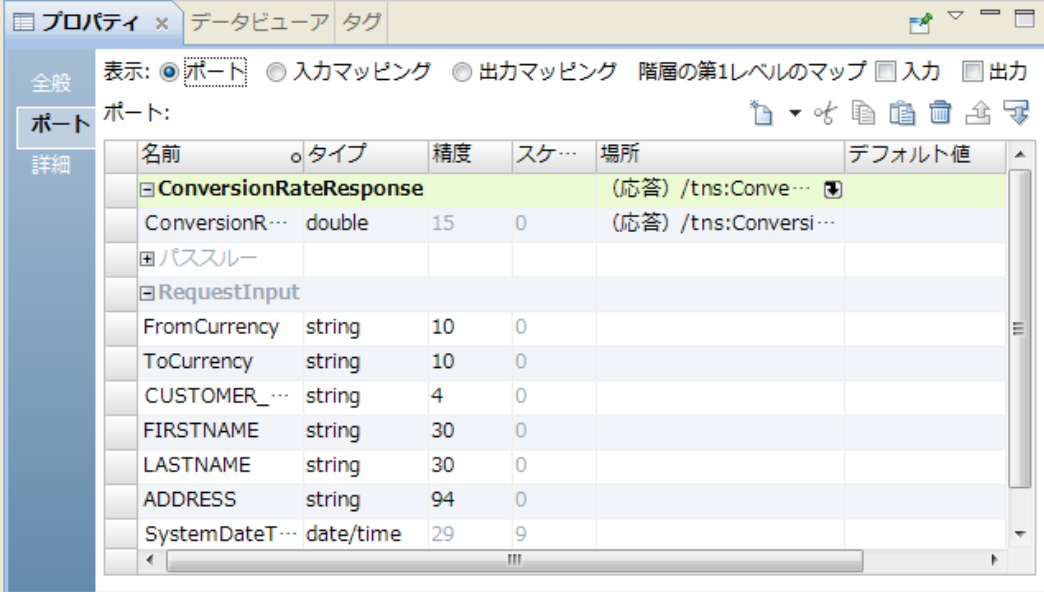
WSDL ファイルには複数の操作を定義できます。Web サービスコンシューマトランスフォーメーションを作成する場合は、実行する操作を選択します。操作入力と操作出力の階層を Web サービスコンシューマトランスフォーメーションに表示できます。これらの階層によって、SOAP リクエストメッセージと SOAP レスポンスメッセージの構造が定義されます。

1 方向入力処理を使用して WSDL をインポートすることもできます。1 方向入力処理を使用して WSDL をインポートする場合は、ダミー出力ポートを作成する必要があります。

Web サービスコンシューマトランスフォーメーションのポート

トランスフォーメーションのポートを表示するときに、操作階層を確認する必要がない場合はポートを表示します。ポートを表示している場合は、グループの定義、ポートの定義、操作出力から出力ポートへのノードのマップを行うことができます。

以下の図に、再利用不可能な Web サービスコンシューマトランスフォーメーションのポートを示します。



| 名前 | タイプ | 精度 | スケ... | 場所 | デフォルト値 |
|------------------------|-----------|----|-------|-----------------------|--------|
| ConversionRateResponse | | | | (応答) /tns:Conve... | |
| ConversionR... | double | 15 | 0 | (応答) /tns:Conversi... | |
| パススルー | | | | | |
| RequestInput | | | | | |
| FromCurrency | string | 10 | 0 | | |
| ToCurrency | string | 10 | 0 | | |
| CUSTOMER_... | string | 4 | 0 | | |
| FIRSTNAME | string | 30 | 0 | | |
| LASTNAME | string | 30 | 0 | | |
| ADDRESS | string | 94 | 0 | | |
| SystemDateT... | date/time | 29 | 9 | | |

Web サービスコンシューマトランスフォーメーションには、複数の入力グループと複数の出力グループを設定できます。ポートを作成するときは、グループを作成して、そのグループにポートを追加します。操作入力階層または操作出力階層の構造に基づいて、ポートをグループ階層で定義します。キーを追加して、子グループを親グループに関連付けます。プライマリキーは、階層内の最下位のグループを除くすべてのグループに必要です。外部キーは、ルートグループを除き階層内のすべてのグループに必要です。

このトランスフォーメーションには、RequestInput という名前のルート入力グループがあります。プライマリキーをルート入力グループに追加する必要があります。プライマリキーは、string 型、bigint 型、integer 型のいずれかにする必要があります。

ルート入力グループにはパススルーポートを追加できます。パススルーポートは、データを変更せずにトランスフォーメーションを介してデータを渡します。パススルーポートは、入力データ内で 1 回しか指定できません。パススルーポートは、任意の出力グループに追加できます。出力ポートを入力ポートに関連付けます。SOAP リクエストを通じて渡した入力値は、SOAP レスポンスからの出力行に繰り返し出現します。

さらに、HTTP ヘッダー、クッキーポート、動的 URL ポート、Web サービスセキュリティ認証用ポートを、ルート入力グループに追加することができます。ルートグループ内のデータは 1 回だけ出現します。

操作出力ノードを出力ポートへマップするために、[場所] カラム内のフィールドをクリックし、[場所の選択] ダイアログボックスで階層を展開します。次に、階層からノードを選択します。

HTTP ヘッダー入力ポート

Web サービスには追加の HTTP ヘッダーが必要な場合があります。ルート入力グループ内に入力ポートを作成して、Web サービスプロバイダに追加のヘッダー情報を渡すことができます。

HTTP ヘッダーと HTTP ポートを追加するには、追加先のルート入力グループを選択し、**[新規]** ボタンの隣の矢印をクリックします。次に、**[HTTP ヘッダー]** をクリックします。ヘッダー名とポート名を入力します。

複数の HTTP ヘッダーを作成することができます。

その他の入力ポート

事前定義済みの入力ポートを Web サービスコンシューマトランスフォーメーションに追加できます。

以下の事前定義済み入力ポートを追加できます。

クッキーポート

クッキー認証を使用するよう Web サービスコンシューマトランスフォーメーションを設定することができます。リモート Web サービスサーバーは、クッキーに基づいて、Web サービスコンシューマユーザーを追跡します。マッピングで Web サービスの呼び出しを繰り返し行う場合に、パフォーマンスが向上します。

クッキーポートを Web サービス要求メッセージに対して投影する場合、Web サービスプロバイダは応答メッセージでクッキー値を返します。クッキー値は、マッピング内の他のトランスフォーメーションダウンストリームに渡すことも、ファイル内に保存することもできます。クッキー値をファイルに保存する場合、そのクッキー値を Web サービスコンシューマトランスフォーメーションに対する入力として設定できます。

クッキー出力ポートは、Web サービスコンシューマトランスフォーメーションの任意の出力グループに投影できます。

エンドポイント URL ポート

Web サービスコンシューマトランスフォーメーションは、エンドポイント URL を使用して Web サービスに接続します。エンドポイント URL は、WSDL ファイル、Web サービス接続、またはエンドポイント URL 入力ポートで定義できます。トランスフォーメーションが URL をポートで動的に受信すると、Data Integration Service は WSDL ファイルまたは Web サービス接続で定義されている URL をオーバーライドします。

Web サービスコンシューマトランスフォーメーションには、Web サービス要求ごとに URL ポート値を 1 個設定できます。エンドポイント URL ポートはルート入力グループに追加します。

WS-Security ポート

Web サービスのセキュリティは、Web サービス接続で有効にします。Web サービスのセキュリティを有効にするときは、Web サービス接続または WS-Security 入力ポートでユーザー名とパスワードを定義する必要があります。

WS-Security ポートを追加するときは、トランスフォーメーションの入力ポートを通じてユーザー名とパスワードを渡します。トランスフォーメーションがユーザー名とパスワードをポートで動的に受信すると、Data Integration Service は Web サービス接続で定義されている値をオーバーライドします。

注: Web サービス接続には、HTTP と WS-Security の認証用のユーザー名とパスワードが 1 組あります。

事前定義済みの入力ポートを追加するには、**[ポート]** 領域でルート入力グループをクリックします。**[新規]** ボタンの隣の矢印をクリックし、**[その他のポート]** をクリックします。追加するポートを選択します。

Web サービスコンシューマトランスフォーメーションの入力マッピング

トランスフォーメーションのポートを表示するときに、操作入力階層を確認するには入力マッピングを表示します。入力マッピングを表示している場合は、入力グループの定義、入力ポートの定義、操作入力ノードへの入力ポートのマッピングを行うことができます。

入力マッピングは以下の領域で構成されています。

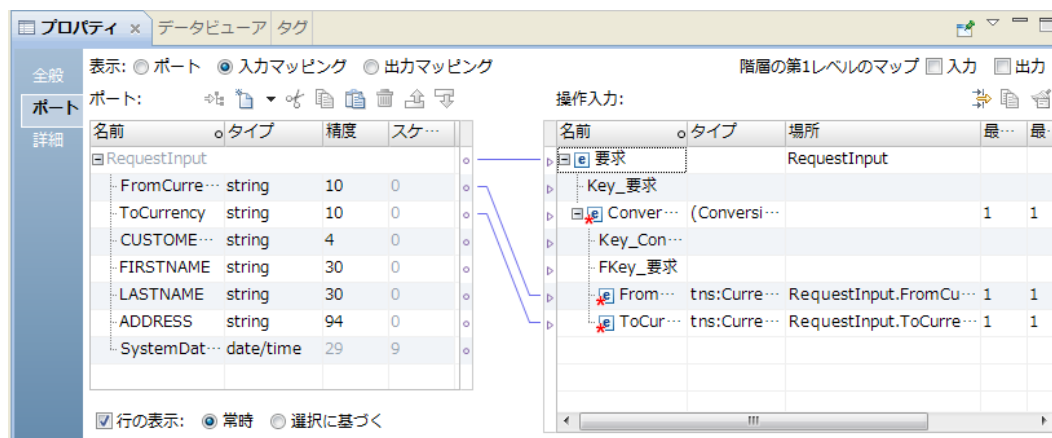
ポート

【ポート】領域では、トランスフォーメーションの入力グループと入力ポートを作成します。

操作入力

【操作入力】領域には、Web サービスコンシューマトランスフォーメーションから Web サービスに伝送される SOAP リクエストメッセージ内のノードが表示されます。トランスフォーメーションの作成に使用する WSDL データオブジェクトにより、操作入力階層が定義されます。

以下の図に、再利用不可能な Web サービスコンシューマトランスフォーメーションの入力マッピングを示します。



入力ポートを作成したら、入力ポートを【ポート】領域から【操作入力】領域のノードにマップします。入力ポートを操作入力内のノードにマップすると、ポートの場所が【操作入力】領域の【場所】カラムに表示されます。

入力階層の第 1 レベルをマップするよう選択すると、Developer ツールは操作入力の第 1 レベルのノードを入力ポートにマップします。また、Developer ツールはマッピングを実行するためのポートも作成します。階層の第 1 レベルに複数回出現する親ノードが含まれ、その親ノードに 1 回または複数回出現する子ノードがある場合、Developer ツールは階層の第 1 レベルをマップしません。

1 つの文字列またはテキストの入力ポート内の XML データを、SOAP リクエストのメッセージ全体にマップすることができます。XML データを SOAP リクエストのメッセージ全体にマップする場合、その操作入力のポートをノードにマップすることはできません。

入力ポートと操作入力内のノードを結ぶ線が表示されるように設定することができます。


入力ポートをノードにマップするためのルールとガイドライン

入力ポートを操作入力階層内のノードにマップする場合には、以下のルールを確認します。

- 階層内の 1 個のノードに対して、1 個の入力ポートをマップできます。同じポートを階層内の任意の数のキーにマップできます。
- 入力ポートとノードのデータ型には互換性が必要です。
- 1 つの入力グループ内のノードを、操作入力内の複数の階層レベルにマップできます。
- 操作入力のキーに入力ポートをマップする必要があります。キーにマップするポートのデータ型は、string、integer、または bigint であることが必要です。操作入力内の、SOAP メッセージに含める階層レベルより上位にある全レベルのキーに、データをマップします。マップするレベルと、その上位にある全レベルの外部キーを含めます。
注: 操作入力階層の最下位レベルだけをマップする場合は、入力ポートをキーにマップする必要はありません。
- string 型、bigint 型、または integer 型の複数の入力ポートを **【操作入力】** 領域のキーにマップして、複合キーを作成できます。複合キーの **【場所】** フィールドをクリックして、入力ポートの順序を変更したり、いずれかのポートを削除したりすることができます。

[ビューのカスタマイズ] のオプション

【操作入力】 領域にキーを表示するように、操作入力階層を変更することができます。ノードの順序を定義するグループ化構造を表示することもできます。

【ビューのカスタマイズ】 ボタン () (**【操作入力】** 領域内) をクリックします。以下のいずれかのオプションを有効にします。

シーケンス、選択、およびすべて

要素定義が、シーケンス、選択、すべてのいずれかであるかどうかを示す線を表示します。

全グループ内のすべてのノードを SOAP メッセージ内に含める必要があります。

シーケンスグループ内のノードは、WSDL 内に定義された順序で並んでいる必要があります。

選択グループ内の少なくとも 1 つ以上のノードが、SOAP メッセージに指定されている必要があります。

キー

【操作入力】 領域にキーを表示します。 **【操作入力】** 領域には、各グループ用のキーが含まれています。
【ポート】 領域内の入力ポートに、キーを追加できます。

操作入力への入力ポートのマッピング

操作入力マッピングを表示している場合は、入力グループの定義、入力ポートの定義、操作入力ノードへの入力ポートのマッピングを行うことができます。

1. Web サービスコンシューマトランスフォーメーションを開きます。
2. トランスフォーメーション入力マッピングを表示するには、以下のいずれかの方法を使用します。
 - 再利用可能なトランスフォーメーションの場合は、**【概要】** ビューをクリックします。入力マッピングの表示を選択します。
 - 再利用不可能なトランスフォーメーションの場合は、**【ポート】** タブ (**【プロパティ】** ビュー内) をクリックします。入力マッピングの表示を選択します。
3. ルート入力グループのプライマリキーを定義します。

4. 次のいずれかの方法を使用して、入力グループまたは入力ポートを【ポート】領域に追加します。

| オプション | 説明 |
|----------------------------|---|
| ノードをドラッグする | 【操作入力】領域のグループノードまたは子ノードを、【ポート】領域の空のカラムにドラッグします。グループノードの場合、Developer ツールによってポートのないグループが追加されます。 |
| グループまたはポートを手動で追加する | グループを追加するには、【新規】ボタンの隣の矢印をクリックし、【グループ】をクリックします。ポートを追加するには、【新規】ボタンの隣の矢印をクリックし、【フィールド】をクリックします。 |
| 別のトランスフォーメーションからポートをドラッグする | エディタ内で、別のトランスフォーメーションから Web サービスコンシューマトランスフォーメーションへ、ポートをドラッグします。 |
| ポートをコピーする | 別のトランスフォーメーションからポートを選択し、【ポート】領域へコピーします。ポートをコピーするには、キーボードショートカットを使用するか、Developer ツールの【コピー】ボタンと【貼り付け】ボタンを使用します。 |
| 【階層の第 1 レベルのマップ】を選択する | 【階層の第 1 レベルのマップ】を選択します。Developer ツールによって、操作入力の第 1 レベル内のノードが、入力ポートおよび入力グループにマップされます。また、マッピングを実行するための入力ポートと入力グループが作成されます。 |

5. ポートを手動で作成するか、別のトランスフォーメーションからコピーした場合は、【操作入力】領域の【場所】カラムをクリックし、リストからポートを選択します。
6. 以下のいずれかの方法を使用して、入力ポートを複合キーとしてマップします。

| オプション | 説明 |
|------------------------------|--------------------------------------|
| 入力ポートをドラッグする | 2 つ以上の入力ポートを選択し、操作入力階層内のキーにドラッグします。 |
| 【場所の選択】ダイアログボックスから入力ポートを選択する | 操作入力階層でキーの【場所】カラムをクリックし、入力ポートを選択します。 |

7. 以下のいずれかの方法を使用して、ノードの場所をクリアします。

| オプション | 説明 |
|------------------|--|
| 【クリア】をクリックする | 【操作入力】領域でノードを 1 つ以上選択し、【クリア】をクリックします。 |
| ポートとノードを結ぶ線を削除する | 操作入力領域で入力ポートとノードを結ぶ線を 1 本以上選択し、Delete キーを押します。 |

8. 関連付けられている WSDL データオブジェクト内に、anyType 要素、任意の要素、anyAttribute 属性、派生型の要素、または置き換えグループが含まれている場合は、【操作入力】領域でオブジェクトを選択します。ノードの【タイプ】カラムで【選択】をクリックし、リストから 1 つ以上のタイプ、要素、または属性を選択します。
9. 文字列入力ポートまたはテキスト入力ポートから完全な SOAP リクエストに XML データをマップするには、ポートを右クリックし、【XML としてマップ】を選択します。

Web サービスコンシューマトランスフォーメーションの出力マッピング

トランスフォーメーションのポートを表示するときに、操作出力階層を確認するには出力マッピングを表示します。出力マッピングを表示している場合は、出力グループの定義、出力ポートの定義、出力ポートへの操作出力ノードのマッピングを行うことができます。

出力マッピングは、次に挙げる領域で構成されています。

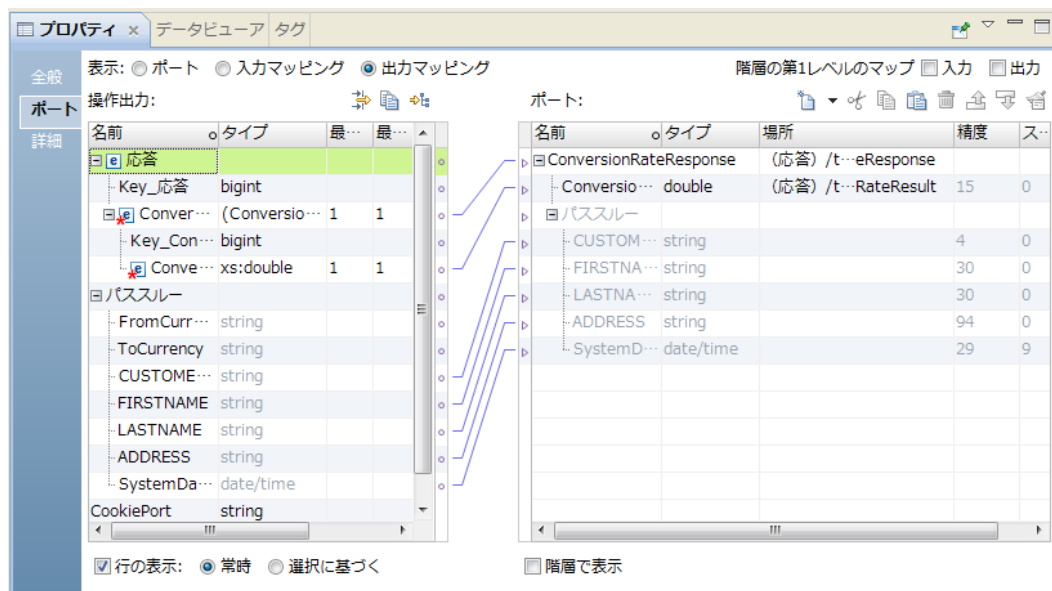
操作出力

【操作出力】領域には、Web サービスから Web サービスコンシューマトランスフォーメーションに返される SOAP レスポンスメッセージ内のノードが表示されます。トランスフォーメーションの作成に使用する WSDL データオブジェクトにより、操作出力階層が定義されます。

ポート

【ポート】領域では、トランスフォーメーションの出力グループと出力ポートを作成します。

以下の図に、再利用不可能な Web サービスコンシューマトランスフォーメーションの出力マッピングを示します。



出力ポートを作成したら、【操作出力】領域から【ポート】領域のポートに、ノードをマップします。操作出力から出力ポートにノードをマップすると、【ポート】領域の【場所】カラムにノードの場所が表示されます。

出力階層の第 1 レベルをマップするよう選択すると、Developer ツールは操作出力の第 1 レベルのノードを出力ポートにマップします。また、Developer ツールはマッピングを実行するためのポートも作成します。階層の第 1 レベルに複数回出現する親ノードが含まれ、その親ノードに 1 回または複数回出現する子ノードがある場合、Developer ツールは階層の第 1 レベルをマップしません。

出力ポートを階層構造で表示するように設定することもできます。子グループはそれぞれ、その親グループの下に表示されます。操作出力内のノードと出力ポートを結ぶ線が表示されるように設定することができます。

関連付けられた WSDL データオブジェクトがリポジトリから削除された場合、Developer ツールは操作ノードの場所を出力マッピングに保持します。出力マッピングを表示すると、【ポート】領域の出力ポートの【場所】カラムに、操作ノードの場所が表示されています。別の WSDL をトランスフォーメーションに関連付けると、Developer ツールがそれぞれの場所が有効であるかどうかを調べます。場所が有効でなくなった場合、出力マッピングの【ポート】領域にある操作ノードの場所を Developer ツールがクリアします。

ノードを出力ポートにマップするためのルールとガイドライン

操作出力階層内のノードを出力ポートにマップする際には、以下のルールを確認します。

- 操作出力ノードおよび出力ポートのデータ型には互換性が必要です。
- グループ内の複数の出力ポートにノードをマップすることはできません。
- パススルーポートを除く各出力ポートには、有効な場所を指定する必要があります。
- 複数出現子ノードを空の出力ポートにドラッグした場合、グループを他の出力グループに関連付ける必要があります。関連付けるグループを選択すると、Developer ツールにより、グループ間に関連付けるキーが作成されます。
- 複数出現要素を、親要素が含まれているグループにドラッグする場合、含める子要素の出現回数を設定することができます。または、親グループを、トランスフォーメーション出力内の複数出現子グループで置き替えることもできます。

SOAP メッセージを XML としてマップ

データを個別の出力ポートに返す代わりに、完全な SOAP メッセージを XML としてマップできます。

SOAP メッセージを XML としてマップすると、Data Integration Service により完全な SOAP メッセージが 1 個のポートに返されます。出力ポートは作成しないでください。

完全なメッセージをマップするには、**【操作出力】** 領域でルートグループを右クリックします。**【XML としてマップ】** を選択します。

Developer ツールにより、文字列出力ポートが作成されます。精度は 65535 バイトです。

【ビューのカスタマイズ】 のオプション

【操作出力】 領域内にクッキーポート、パススルーポート、およびキーを表示するように、操作出力階層を変更することができます。ノードの順序を定義するグループ化構造を表示することもできます。

【ビューのカスタマイズ】 ボタン（**【操作出力】** 領域内）をクリックします。以下のいずれかのオプションを有効にします。

シーケンス、選択、およびすべて

要素定義が、シーケンス、選択、すべてのいずれかであるかどうかを示す線を表示します。

全グループ内のすべてのノードを SOAP メッセージ内に含める必要があります。

シーケンスグループ内のノードは、WSDL 内に定義された順序で並んでいる必要があります。

選択グループ内の少なくとも 1 つ以上のノードが、SOAP メッセージに指定されている必要があります。

キー

【操作出力】 領域にキーを表示します。**【操作出力】** 領域には、各グループのキーが含まれています。

【ポート】 領域内の出力ポートに、キーを追加できます。

パススルーポート

【操作出力】 領域には、パススルーポートが表示されます。パススルーポートとは、トランスフォーメーションを通じて、変更を加えずにデータを渡すポートのことです。操作出力内のパススルーポートを、Web サービスコンシューマトランスフォーメーションの任意の出力グループへ投影することができます。パススルーポートはデータを 1 回だけ受け取るため、SOAP メッセージ内のルートレベルにあります。

クッキーポート

クッキーポートを表示します。クッキー認証を設定するときに、リモート Web サービスサーバーは、クッキーに基づいて、Web サービスコンシューマユーザーを追跡します。Web サービスクッキーを要求メ

メッセージ内に投射する場合、Web サービスは応答メッセージでクッキー値を返します。操作出力内のクッキー値を、Web サービスコンシューマトランスフォーメーション任意の出力グループへ投影することができます。

出力ポートへの操作出力のマッピング

操作出力マッピングを表示している場合は、出力グループの定義、出力ポートの定義、出力ポートへの操作出力ノードのマッピングを行うことができます。

- Web サービスコンシューマトランスフォーメーションを開きます。
- トランスフォーメーション出力マッピングを表示するには、以下のいずれかの方法を使用します。
 - 再利用可能なトランスフォーメーションの場合は、**【概要】** ビューをクリックします。出力マッピングの表示を選択します。
 - 再利用不可能なトランスフォーメーションの場合は、**【ポート】** タブ（**【プロパティ】** ビュー内）をクリックします。出力マッピングの表示を選択します。
- 以下のいずれかの方法を使用して、出力グループまたは出力ポートを **【ポート】** 領域に追加します。

| オプション | 説明 |
|-------------------------------|--|
| ノードをドラッグする | 【操作出力】 領域のグループノードまたは子ノードを、 【ポート】 領域の空のカラムにドラッグします。グループノードの場合、Developer ツールによってポートのないグループが追加されます。 |
| グループまたはポートを手動で追加する | グループを追加するには、 【新規】 ボタンの隣の矢印をクリックし、 【グループ】 をクリックします。ポートを追加するには、 【新規】 ボタンの隣の矢印をクリックし、 【フィールド】 をクリックします。 |
| 別のトランスフォーメーションからポートをドラッグする | エディタ内で、別のトランスフォーメーションから Web サービスコンシューマトランスフォーメーションへ、ポートをドラッグします。 |
| ポートをコピーする | 別のトランスフォーメーションからポートを選択し、 【ポート】 領域へコピーします。ポートをコピーするには、キーボードショートカットを使用するか、Developer ツールの 【コピー】 ボタンと 【貼り付け】 ボタンを使用します。 |
| 【階層の第 1 レベルのマップ】 を選択する | 【階層の第 1 レベルのマップ】 を選択します。Developer ツールによって、操作出力の第 1 レベル内のノードが、出力ポートおよび出力グループにマップされます。また、マッピングを実行するための出力ポートと出力グループが作成されます。 |

- ポートを手動で作成するか、別のトランスフォーメーションからコピーした場合は、**【ポート】** 領域の **【場所】** カラムをクリックし、リストからノードを選択します。
- 以下のいずれかの方法を使用して、ポートの場所をクリアします。

| オプション | 説明 |
|----------------------|---|
| 【クリア】 をクリックする | 【ポート】 領域でポートを 1 つ以上選択し、 【クリア】 をクリックします。 |
| ノードとポートを結ぶ線を削除する | 操作出力のノードと出力ポートを結ぶ線を 1 本以上選択し、Delete キーを押します。 |

6. 関連付けられている WSDL データオブジェクト内に、anyType 要素、任意の要素、anyAttribute 属性、派生型の要素、または置き換えグループが含まれている場合、**【操作出力】** 領域でオブジェクトを選択します。ノードの **【タイプ】** カラムで **【選択】** をクリックし、リストから 1 つ以上のタイプ、要素、または属性を選択します。
7. 完全な SOAP レスポンスメッセージを XML としてマップするには、**【操作出力】** 領域でルートグループを右クリックし、**【XML としてマップ】** を選択します。

Web サービスコンシューマトランスフォーメーションの詳細プロパティ

Web サービスコンシューマトランスフォーメーションの詳細プロパティには、追跡レベル、汎用フォールトポート、Web サービス接続、および同時 Web サービス要求のメッセージが含まれます。

【詳細】 タブで、Web サービスコンシューマトランスフォーメーションの以下の詳細プロパティを定義できます。

トレースレベル

このトランスフォーメーションのログに表示される情報の詳細度。Terse、Normal、Verbose Initialization、Verbose data から選択できます。デフォルトは **【Normal】** です。

SOAP アクション

WSDL に定義されている SOAP アクション値を、Web サービスコンシューマトランスフォーメーションの定数値でオーバーライドします。

汎用 SOAP フォールト処理を有効にする

WSDL で定義されていないフォールトメッセージを返します。フォールトのコードとメッセージを処理する出力ポートを GenericFault 出力グループ内に作成します。

以下の表に、SOAP 1.1 および SOAP 1.2 のフォールト出力ポートを示します。

| フォールト出力ポート (SOAP 1.1) | フォールト出力ポート (SOAP 1.2) | 説明 |
|--------------------------|--------------------------|---|
| フォールトコード | コード* | フォールトの ID コードを返します。 |
| フォールト文字列 | 理由* | フォールトメッセージ内のエラーの説明を返します。 |
| フォールト詳細 | 明細 | Web サービスプロバイダから Web サービスコンシューマトランスフォーメーションへ、汎用フォールトメッセージで渡されるカスタム情報を返します。 |
| フォールトアクタ | ロール | フォールト発生の原因となったオブジェクトに関する情報を返します。 |

| フォールト出力ポート (SOAP 1.1) | フォールト出力ポート (SOAP 1.2) | 説明 |
|------------------------------|--------------------------|----------------------------------|
| - | ノード | フォールトが生成された SOAP ノードの URI を返します。 |
| *コード出力ポートと理由出力ポートは階層になっています。 | | |

注: コードフォールト出力ポートを展開して、SubCode フォールト出力ポートを 1 つのレベルに抽出することができます。

HTTP エラー処理を有効にする

Web サービスから HTTP エラーを返します。GenericFault 出力グループ内に HTTP エラー出力ポートを作成します。

フォールトをエラーとして扱う

フォールトメッセージをマッピングログに記録します。フォルトが発生すると、Data Integration Service によってマッピングのエラー数が 1 件加算されます。このプロパティを無効にすると、初期選択の最適化および最適化にプッシュインを利用できるようになります。デフォルトでは有効になっています。

接続

Web サービスに接続する Web サービス接続オブジェクトを特定します。Developer ツールで Web サービス接続を作成します。Developer ツールまたは Administrator ツールで Web サービス接続を編集します。Web サービス接続を設定するときは、エンドポイント URL、Web サービスに必要なセキュリティのタイプ、および接続タイムアウト期間を設定します。

Web サービスコンシューマトランスフォーメーションは、エンドポイント URL を使用して Web サービスに接続します。エンドポイント URL は、WSDL ファイル、Web サービス接続、またはエンドポイント URL 入力ポートで定義できます。

以下のガイドラインを使用して、Web サービス接続をいつ設定するか判断します。

- WSDL ファイル内の URL と異なるエンドポイント URL を使用する場合、およびエンドポイント URL 入力ポートを使用しない場合に、接続を設定します。
- 接続先の Web サービスに Web サービスセキュリティ、HTTP 認証、または SSL 証明書が必要な場合に、接続を設定します。
- デフォルトの接続タイムアウト時間を変更する場合に、接続を設定します。

注: リポジトリ内の WSDL データオブジェクトは、Web サービス接続に関連付けることができます。関連付けられた接続は、その WSDL から作成される各 Web サービスコンシューマトランスフォーメーションのデフォルト接続になります。

圧縮を有効にする

gzip 圧縮方式による SOAP リクエストのエンコードを有効にし、gzip または deflate による SOAP レスポンスのデコードを有効にします。

XML スキーマ検証

実行時に SOAP レスポンスメッセージを検証します。**【無効な XML でのエラー】** または **【検証なし】** を選択します。

ソート済み入力

入力データを必ずしもすべて処理しなくても、Data Integration Service が出力を生成できるようにします。入力データが操作入力階層内のキーを基準にしてソートされる場合に、ソート済み入力を有効にします。

最適化にプッシュイン

最適化にプッシュインを有効にします。【最適化にプッシュイン】プロパティの【開く】ボタンをクリックして、フィルタ値を受け取るフィルタポートを選択します。フィルタポートごとに、Web サービス応答にフィルタリングされたカラムを含める出力ポートを選択します。

副次作用あり

Web サービスが行を返す以外にいずれかの関数を実行することを示すチェックボックス。Web サービスが、行を返すことに加えて、オブジェクトを変更したり他のオブジェクトまたは関数と対話したりすると、Web サービスコンシューマトランスフォーメーションに副次作用が生じます。副次作用により、Web サービスで、データベースの変更、合計の増加、例外の発生、電子メールへの書き込み、または他の Web サービスの呼び出しが行われる場合があります。最適化にプッシュインまたは初期選択の最適化を可能にするには、【副次作用あり】プロパティを無効にします。デフォルトでは有効になっています。

並行処理の有効化

複数の Web サービス要求を同時に送信できるように、Web サービスに対して複数の同時接続を作成するには、Web サービスコンシューマトランスフォーメーションを有効にします。Web サービスコンシューマトランスフォーメーションを有効にして、Web サービスに対して複数の同時接続を作成するときには、合計メモリ消費量の制限と同時接続制限を設定することができます。

以下の表に、オプションを示します。

| オプション | 説明 |
|-------------------|---|
| 並行処理の有効化 | Web サービスに対して複数の同時接続を作成します。 |
| 同時接続制限 | Web サービスの同時接続の数。デフォルトは 20 です。 |
| 並行処理の合計メモリ制限 (MB) | すべての同時接続の合計メモリ割り当ての制限。デフォルトは 100 MB です。 |

Web サービスのエラー処理

Web サービスコンシューマトランスフォーメーションは、SOAP フォールトと HTTP エラーをマッピング内のダウンストリームに渡すように設定できます。フォールトが発生したときに、そのフォールトをエラー数に加算することができます。Web サービスのエラー処理は、トランスフォーメーションの詳細プロパティで設定します。

Web サービスは、応答メッセージを返すか、またはフォールトを返します。フォールトはエラーの一種です。Web サービスでは、発生するエラーに基づいて各種のフォールトを生成できます。

Web サービスコンシューマトランスフォーメーションは、以下のタイプのフォールトを返すことができます。

SOAP フォールト

WSDL によって定義された SOAP エラー。フォールトを Web サービス応答メッセージに含めて返す出力エラーポートを設定します。SOAP 1.1 バインディングの場合、Data Integration Service は、フォールトのフォールトメッセージ、フォールトコード、フォールト文字列、フォールトアクタ要素を返します。SOAP 1.2 バインディングの場合、Data Integration Service は、フォールトのフォールトメッセージ、コード、理由、ノード、ロール要素を返します。

汎用 SOAP フォールト

Web サービスは実行時に汎用 SOAP フォールトを生成します。フォールト要素は、SOAP 1.1 バインディングと SOAP 1.2 バインディングで異なります。WSDL は、汎用 SOAP フォールトを定義しません。汎用 SOAP フォールトには、認証エラーや SOAP リクエストエラーなどがあります。

HTTP エラー

トランスフォーメーション内で HTTP エラー処理を有効にすると、Developer ツールによって HTTP フォールト出力ポートが追加されます。Data Integration Service は、1 個の文字列ポート内の Web サービスから HTTP エラーを返します。HTTP エラーには、エラーコードとメッセージが含まれます。

Web サービスからの SOAP レスポンスに無効な XML データが含まれていた場合、Web サービスコンシューマトランスフォーメーションはエラーを返します。

SOAP フォールトをエラーとして扱うかどうか、設定することができます。[フォールトをエラーとして扱う]を有効にした場合に SOAP フォールトが発生すると、Data Integration Service によってマッピングのエラー数が 1 件加算されます。フォールトはメッセージログに記録されます。

メッセージの圧縮

SOAP メッセージ圧縮を有効にした場合、Web サービスコンシューマトランスフォーメーションは Web サービス要求メッセージを圧縮し、圧縮済みの Web サービス応答メッセージを受け取ります。

Web サービスコンシューマトランスフォーメーションは、SOAP リクエストを gzip 圧縮でエンコードします。このトランスフォーメーションでは、gzip 圧縮または deflate 圧縮によってエンコードされた応答メッセージを受け付けます。

Web サービスからの応答を受信すると、Data Integration Service は SOAP メッセージ内の Content-Encoding HTTP ヘッダーを調べ、SOAP メッセージをデコードします。

デフォルトは、圧縮なしのエンコードです。Web サービスは SOAP レスポンスを圧縮しません。

以下の表に、圧縮がオンになっている場合とオフになっている場合の要求メッセージと応答メッセージのヘッダーを示します。

| 圧縮 | ヘッダー |
|----|---|
| オン | Content-Encoding ヘッダー: gzip Accept-Encoding ヘッダー: gzip、deflate |
| オフ | 空の Content-Encoding ヘッダー 空の Accept-Encoding ヘッダー |

応答メッセージは Web サービスによってデフォルトの圧縮方式でエンコードされていることもあります。メッセージが gzip または deflate でエンコードされている場合、Web サービスコンシューマトランスフォーメーションはそのメッセージをデコードします。Web サービスが応答メッセージを予期せずエンコードしていた場合、Web サービスコンシューマトランスフォーメーションはマッピングログにメッセージを記録します。

圧縮はトランスフォーメーションの詳細プロパティで有効にします。

並行処理

複数の Web サービス要求を同時に送信できるように、Web サービスコンシューマトランスフォーメーションを有効にして、Web サービスに対して複数の同時接続を作成することができます。

例えば、銀行情報を問い合わせているときに、複数の行が同時に送信される並行処理が有効になるように Web サービスコンシューマトランスフォーメーションを設定することができます。入力行が 20 行ある場合は、処理を高速化するために 20 個の要求を同時に送信できます。

Web サービスコンシューマトランスフォーメーションで並行処理を有効にするときは、合計メモリ消費量の制限を設定できます。

Web サービスコンシューマトランスフォーメーションで並行処理を有効にするときは、Web サービスの同時接続の数を設定できます。

並行処理に関するルールとガイドライン

並行処理の使用中は以下のルールおよびガイドラインを使用します。

- 並行処理では、ソートされた入力がある Web サービスに対する複数の同時接続としてサポートされます。順序付けられた出力行はサポートされません。
- 並行処理はデータセットが 100 行を超える場合に使用します。
- Web サービスの同時接続の数が増えないようにすることをお勧めします。Web サービスの同時接続の数は、オペレーティングシステムによって使用されるソケットの数に関係します。ソケットの数が増えると費用がかかります。
- 並行処理機能の使用中は、最適なパフォーマンスのために、RAM が 100 MB 以上のマルチコアプロセッサを備えたシステムを使用します。
- 並行処理のメモリ制限は、Web サービスの起動中に並行処理のワークフローによって消費されるメモリを表します。
- Web サービスコンシューマトランスフォーメーションで並行処理を有効にするときは、メモリ消費量の制限を設定できます。メモリ消費量はサーバーの物理 RAM を超えないようにしてください。

並行処理のベストプラクティス

並行処理使用時の最適なパフォーマンスを実現するために、次のベストプラクティスに従ってください。

- 並行処理の合計メモリの制限と同時接続制限のデフォルト値の変更を避ける。
- 100 行未満のデータセットに対する並行処理の使用を避ける。
- 並行処理使用時のマッピングでのパススルーポートを避ける。

フィルタの最適化

フィルタを最適化すると、マッピングを通過する行数が減り、パフォーマンスが向上します。Data Integration Service は、初期選択の最適化と最適化にプッシュインを適用することができます。

Data Integration Service でフィルタ最適化方式が適用されると、マッピング内でフィルタができるだけソースの近くに移動されます。Data Integration Service がマッピング内のトランスフォーメーションの前にフィルタを移動できない場合、フィルタロジックをトランスフォーメーションにプッシュできる可能性があります。

Web サービスコンシューマトランスフォーメーションでの初期選択の最適化の有効化

トランスフォーメーションに副次作用がなく、フォルトをエラーとして扱うこともない場合は、Web サービスコンシューマトランスフォーメーションで初期選択の最適化を有効にします。

1. Web サービスコンシューマトランスフォーメーションの **【詳細プロパティ】** ビューを開きます。
2. **【フォルトをエラーとして扱う】** をクリアします。
3. **【副次作用あり】** をクリアします。

Web サービスコンシューマトランスフォーメーションによるプッシュイン最適化

トランスフォーメーションが SQL データサービスの仮想テーブルにある場合は、プッシュイン最適化を Web サービスコンシューマトランスフォーメーションで設定できます。

データオブジェクトマッピングは、Web サービスを呼び出して、エンドユーザーの SQL クエリの文に基づいてデータのセットまたはデータのサブセットを取得します。エンドユーザーの SQL クエリには、オプションのフィルタ条件が含まれます。

Web サービスコンシューマトランスフォーメーションは、最適化にプッシュインによって、フィルタポートのフィルタ値を受け取ります。フィルタポートは、最適化にプッシュインを設定するときにフィルタポートとして特定する接続されていない入力ポートです。フィルタポートには、エンドユーザーのクエリにフィルタが含まれていない場合に Web サービスがすべての行を返すようにするデフォルト値があります。フィルタポートはパススルーポートではありません。

注: フィルタフィールドは、Web サービス要求のルートグループの一部である必要があります。

フィルタポートの設定時に、Web サービスコンシューマトランスフォーメーションで Web サービス応答からカラムデータを受け取る出力ポートを特定します。例えば、フィルタポートが EmployeeID という名前の入力ポートである場合、応答の出力ポートのポート名は EmployeeNum になります。Developer ツールは、読み込まれた仮想テーブルから Web サービスコンシューマ要求にフィルタロジックをプッシュするために、入力フィルタポートと出力ポートを関連付ける必要があります。Web サービス要求の入力ポートは、通常、Web サービス応答の出力ポートとは異なります。

フィルタフィールドをパススルーポートにすることはできません。フィルタポートの設定時、ポートのデフォルト値がフィルタ条件の値に変わるため、パススルー出力ポートの値が変わります。出力パススルーポートに基づくフィルタは、予期しない結果を返します。

Web サービスコンシューマトランスフォーメーションには複数の式をプッシュすることができます。各フィルタ条件は次の形式である必要があります。

<Field> = <Constant>

フィルタ条件は AND で結合する必要があります。フィルタ条件を OR で結合することはできません。

Web サービスコンシューマトランスフォーメーションによるプッシュイン最適化の例

SQL データサービスは、すべての顧客の注文を返すか、ユーザーから受け取る SQL クエリに基づいて特定の顧客の注文を返します。

SQL データサービスには、次のコンポーネントから構成される論理データオブジェクトが含まれています。

顧客テーブル

顧客情報が含まれる Oracle データベーステーブル。

Web サービスコンシューマトランスフォーメーション

Web サービスを呼び出して顧客の最新の注文を取得するトランスフォーメーション。Web サービスコンシューマトランスフォーメーションには、customerID と orderNum の入力ポートがあります。このトランスフォーメーションには、顧客テーブルから受け取る顧客データが含まれるパススルーポートがあります。orderNum ポートはフィルタポートで、接続されていません。OrderNum には、デフォルト値 "*" があります。Web サービスが Web サービス要求でこの値を受け取ると、すべての注文を返します。

注文仮想テーブル

Web サービスから顧客データと注文データを受け取る仮想テーブル。エンドユーザーはこのテーブルを参照します。注文には、顧客カラム、orderId カラム、および顧客データと注文データが含まれます。

エンドユーザーは次の SQL クエリを SQL データサービスに渡します。

```
SELECT * from OrdersID where customer = 23 and orderID = 56
```

クエリはマッピングを最適化するために分割されます。初期選択の最適化が使用され、フィルタロジック、customer = 23 が読み込まれた顧客テーブルに移動します。最適化にプッシュインが使用され、フィルタロジック、orderID = 56 が Web サービスコンシューマトランスフォーメーションのフィルタポートにプッシュされます。Web サービスコンシューマトランスフォーメーションでは、顧客 23 の ordersID 56 が取得されます。

Web サービスコンシューマトランスフォーメーションによるプッシュイン最適化の有効化

トランスフォーメーションに副次作用がなく、フォルトをエラーとして扱うこともない場合は、Web サービスコンシューマトランスフォーメーションで最適化にプッシュインを有効にします。

1. Web サービスコンシューマトランスフォーメーションの **【詳細プロパティ】** ビューを開きます。
2. **【フォルトをエラーとして扱う】** をクリアします。
3. **【副次作用あり】** をクリアします。
4. **【最適化にプッシュイン】** プロパティの **【開く】** ボタンをクリックします。
5. **【最適化された入力】** ダイアログボックスで、フィルタポート名を選択します。
複数のフィルタポートを選択できます。
6. **【出力】** カラムをクリックします。
7. フィルタポートごとに、Web サービス応答にフィルタリングされたカラムを含める出力ポートを選択します。
8. フィルタポートのデフォルト値を入力します。

注: Web サービスコンシューマポートは、フィルタポートではない限り、デフォルト値を設定できません。

Web サービスコンシューマトランスフォーメーションの作成

再利用可能または再利用不可能な Web サービスコンシューマトランスフォーメーションを作成できます。再利用可能なトランスフォーメーションは、複数のマッピングで使用できます。再利用不可能なトランスフォーメーションは、単一のマッピングで使用されます。

単一の WSDL オブジェクトから SOAP 1.1 バインディングと SOAP 1.2 バインディングの Web サービスコンシューマトランスフォーメーションを作成することができます。

1. トランスフォーメーションを作成するには、次のいずれかの方法を使用します。

| オプション | 説明 |
|-------|---|
| 再利用可能 | Object Explorer ビューで、プロジェクトまたはフォルダーを選択します。 【ファイル】 > 【新規】 > 【トランスフォーメーション】 をクリックします。Web サービスコンシューマトランスフォーメーションを選択し、 【次へ】 をクリックします。 |
| 再利用不可 | マッピングまたはマップレットで、トランスフォーメーションパレットからエディタに Web サービスコンシューマフォーメーションをドラッグします。 |

【新しい Web サービスコンシューマトランスフォーメーション】 ダイアログボックスが表示されます。

2. [参照] をクリックし、Web サービスの要求メッセージと応答メッセージを定義する WSDL データオブジェクトを選択します。

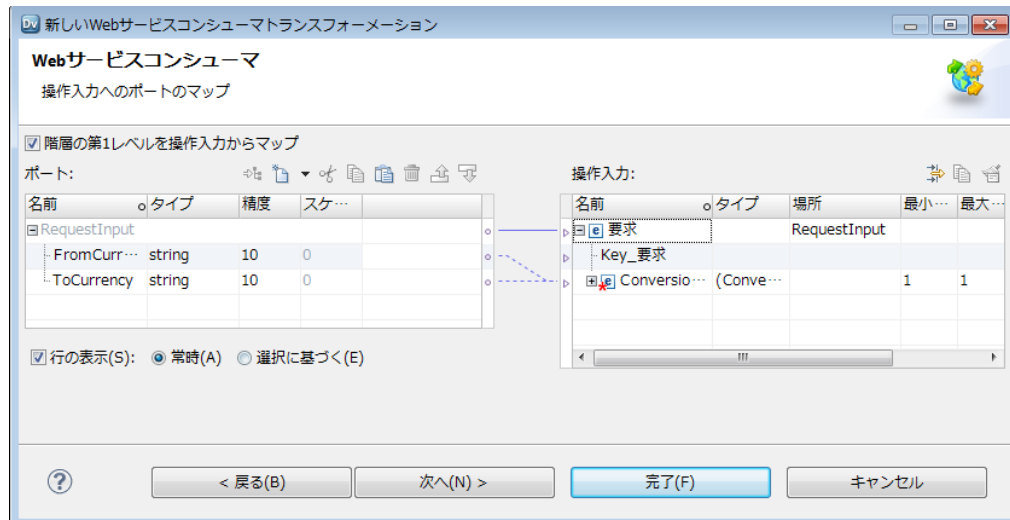
リポジトリ内に WSDL が存在しない場合、[新しい Web サービスコンシューマトランスフォーメーション] ダイアログボックスから WSDL をインポートすることができます。

3. [参照] をクリックし、WSDL から操作を選択します。

SOAP 1.1 バインディングまたは SOAP 1.2 バインディングのある操作を選択できます。

4. [次へ] をクリックします。

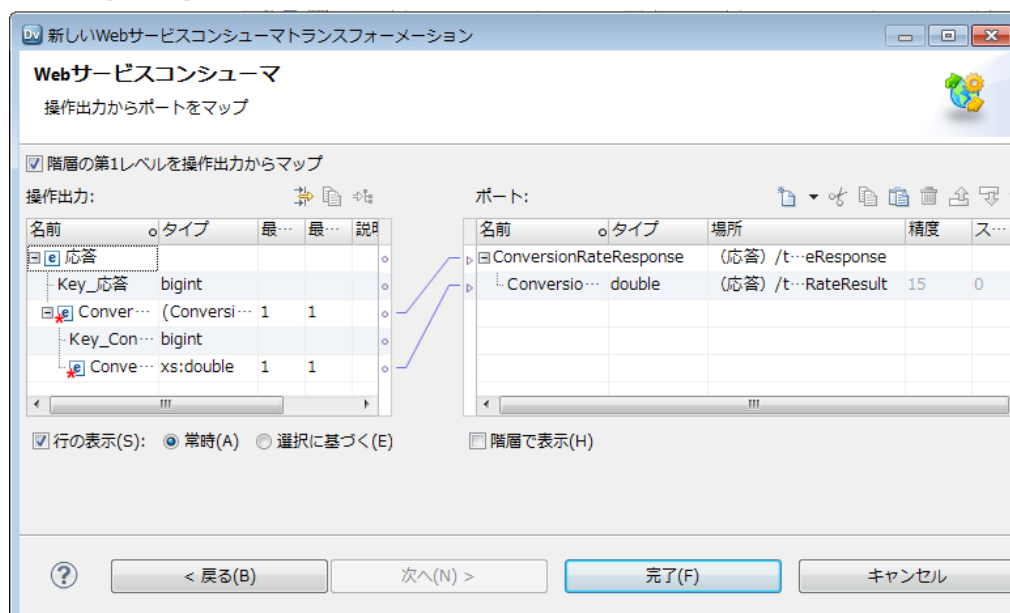
[操作入力へのポートのマッピング] 画面が表示されます。[ポート] 領域には、トランスフォーメーションの入力グループと入力ポートが表示されます。[操作入力] 領域には、要求メッセージ階層が表示されます。



5. 入力グループと入力ポートを定義し、操作入力ノードに入力ポートをマッピングします。

6. [次へ] をクリックします。

[操作出力からポートをマッピング] 画面が表示されます。[操作出力] 領域に、応答メッセージ階層が表示されます。[ポート] 領域に、トランスフォーメーションの出力グループと出力ポートが表示されます。



7. 出力グループと出力ポートを定義し、出力ポートに操作用出力ノードをマップします。
8. **【完了】** をクリックします。
9. **【詳細】** ビューをクリックし、トランスフォーメーションプロパティと Web サービス接続を設定します。

Web サービスコンシューマトランスフォーメーションの例

例えば、RT100 製品ラインの注文情報を営業部門へ公開する必要があるとします。営業チームでは、注文の概要と注文の詳細を日常的に照会する必要があります。

仮想テーブル内に毎日の注文情報を公開する論理データオブジェクトを作成します。読み取りマッピングには、最新の RT100 の注文を返す Web サービスコンシューマトランスフォーメーションが含まれます。この Web サービスコンシューマトランスフォーメーションは、RT100 製品ラインに対する毎日の注文の概要と詳細をそれぞれ返す Web サービスを使用します。

入力ファイル

入力ファイルは、製品ライン番号を含むフラットファイルです。

この入力ファイルを定義する物理データオブジェクトを作成します。ファイルには Product_Line というフィールドが 1 個含まれます。フィールド値は RT100 です。 **プロパティビューのランタイムビュー**で、物理データオブジェクトの場所を定義します。

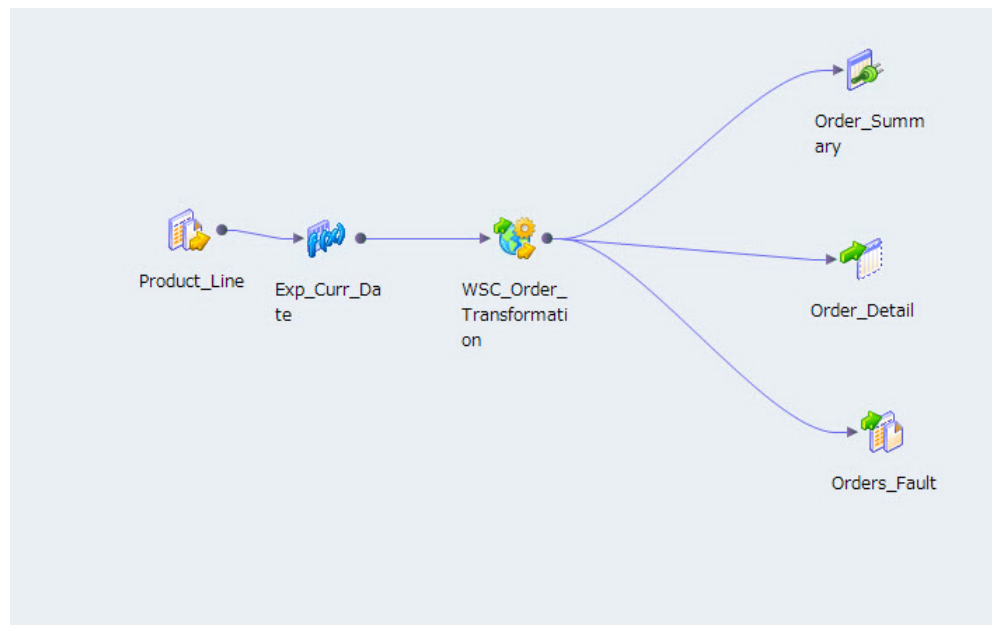
論理データオブジェクトモデル

所属部門のビジネスアナリストは、注文の概要と注文の詳細のテーブル構造を説明する論理データモデルを作成します。この論理データモデルには、Order_Summary と Order_Detail の論理データオブジェクトが含まれます。

論理データモデルを定義するスキーマが、アナリストによってモデリングツールで作成されます。このスキーマから論理データモデルをインポートし、Order_Summary と Order_Detail の論理データオブジェクトを作成します。

論理データオブジェクトマッピング

論理データオブジェクトマッピングは、論理データオブジェクトを通じてデータにアクセスする方法を記述したものです。



読み取りマッピングには以下のオブジェクトが含まれます。

Product_Line

製品ライン番号を含む入力フラットファイル。

Exp_Curr_Date トランスフォーメーション

現在の日付と、Web サービスコンシューマトランスフォーメーションのルートレベル入力グループのプライマリーを返す式トランスフォーメーション。

WSC_Order トランスフォーメーション

Web サービスを使用して注文情報を取得する Web サービスコンシューマトランスフォーメーション。トランスフォーメーションは、製品ラインと現在の日付を応答メッセージに含めて Web サービスに渡します。トランスフォーメーションは、Web サービスから注文情報を応答メッセージとして受け取ります。

Order_Summary テーブル

Order_No、Customer_Id、Qty、Order_Date などの注文情報を含む論理データオブジェクト。

Order_Detail テーブル

Order_No、Product_Id、Qty、Status などの詳細な注文情報を含む論理データオブジェクト。

Orders_Fault

汎用フォールトメッセージを受け取る出力フラットファイル。

Web サービスコンシューマトランスフォーメーション

Web サービスコンシューマトランスフォーメーションは、製品ライン、日付、シーケンス番号を入力として受け取ります。このトランスフォーメーションは、Get_Order_Info Web サービス操作を使用して、注文情報を取得します。

Web サービスコンシューマトランスフォーメーションを作成する場合、要求と応答の Web サービスメッセージを記述する WSDL データオブジェクトを選択します。Web サービスメッセージには、XML 要素の階層グループが含まれます。要素には他の要素を含めることができます。要素によっては、複数回出現するものもあります。トランスフォーメーションを、リポジトリ内の Order_Info WSDL オブジェクトから作成します。

トランスフォーメーションの入力ポートを設定し、ポートを操作入力階層にマップします。操作出力階層から出力ポートにノードをマップします。Web サービス接続および実行時プロパティを設定します。

トランスフォーメーション入力マッピング

ポートビューに入力マッピングを表示している場合、入力ポートを定義し、それらのポートを操作入力のノードにマップできます。

トランスフォーメーションの【**ポート**】領域には、ルートグループと Order グループがあります。ルートグループは、Request 入力グループです。Request 入力グループに、プライマリキーを表すポートを 1 個追加します。

Order グループには **Select_Date** 入力ポートと **Select_Product_Line** 入力ポートがあります。

これらの入力ポートを、【**操作入力**】領域の **Order_Date** ノードと **Product_Line** ノードにマップします。

【**操作入力**】領域は、Web サービスコンシューマトランスフォーメーションが Web サービスに渡す要求メッセージを定義します。【**操作入力**】領域には、デフォルトでノードが表示されます。

トランスフォーメーション出力マッピング

ポートビューに出力マッピングが表示されている場合、操作出力のノードをトランスフォーメーション出力グループにマップすることで、出力ポートを定義できます。

Web サービスは、以下の階層を Web サービス応答メッセージに含めて返します。

```
Response
  Orders
    Order
      Key_Order
      Order_ID
      Order_Date
      Customer_ID
      Total_Qty
      Order_Details
        Order_Detail
          Product_ID
          Description
          Qty
          Status
```

Web サービスから複数の注文が返されます。Order は、Orders レベルの複数出現ノードです。個々の注文に対して、Web サービスは注文の詳細を複数返すことができます。Order_Detail は、Order_Detail レベルの複数出現ノードです。

注: Developer ツールにより、ユーザーインターフェースに Key_Order ノードが追加されます。これらのキーを出力グループにマッピングすることで、グループ間のリレーションを定義できます。この例では、Order_ID は、Order のプライマリキーであると同時に Order_Details の外部キーでもあります。

[ポート] 領域で、以下の出力グループを作成します。

```
Order
  Order_ID
  Order_Date
  Customer_ID
  Total_Qty

Order_Detail
  Order_ID
  Product_ID
  Description
  Qty
  Status
```

Data Integration Service は、Order_ID の値が変更されると必ず、Order グループから行を書き出します。

Data Integration Service は、Order_ID と Product_ID の値が変更されると必ず、Order_Detail グループから行を書き出します。

トランスフォーメーションの詳細プロパティ

Web サービスコンシューマトランスフォーメーションの以下の詳細プロパティを設定します。

汎用 SOAP フォールト処理を有効にする

SOAP フォールトメッセージを受け取る出力ポートを追加します。

接続

Web サービスにアクセスする Web サービス接続を選択します。

圧縮を有効にする

Web サービスコンシューマトランスフォーメーションにより、Web メッセージが gzip で圧縮されます。

第 10 章

REST Web サービス

この章では、以下の項目について説明します。

- [REST Web サービスの概要, 109 ページ](#)
- [REST Web サービスの処理, 110 ページ](#)
- [Web サービスコンシューマトランスフォーメーションの処理, 110 ページ](#)
- [REST Web サービスリソース, 111 ページ](#)
- [リソースマッピング, 114 ページ](#)
- [REST Web サービスの出力トランスフォーメーション, 116 ページ](#)
- [要求メッセージ, 118 ページ](#)
- [応答メッセージ形式, 120 ページ](#)
- [応答のデータレビュー, 121 ページ](#)

REST Web サービスの概要

Informatica REST Web サービスはデータの HTTP 要求を処理し、JSON ファイルまたは XML ファイルの応答を返します。

外部アプリケーション、Web ブラウザ、または REST Web サービスのコンシューマトランスフォーメーションによって、REST Web サービスに接続し、要求の送信ができます。REST Web サービスによって、要求が処理され、クライアントに応答が返されます。

例えば、Web サービス操作を実行する要求を Web サービスクライアントが送信するとします。Web サービスクライアントは、要求で顧客 ID を渡します。注文テーブルから顧客情報と注文情報を取得します。情報をクライアントに JSON ファイルで返します。

Developer tool でデータサービスを定義して、Informatica REST Web サービスを作成します。データオブジェクトを REST Web サービスとしてデプロイすることもできます。

Informatica REST Web サービスには、次のコンポーネントがあります。

リソース

リソースには、REST Web サービスが実行するマッピングと Web サービスが返す応答メッセージの定義が含まれています。出力データのキーフィールドであるリソース ID も含まれています。データオブジェクトからリソースを作成するか、手動で定義できます。オブジェクトを REST Web サービスとしてデプロイした場合は、Developer tool によってリソースが作成されます。Web サービスには複数のリソースを設定できます。

要求メッセージ

タスクを実行するための、Web サービスクライアントから Web サービスへの要求。Informatica Web サービスで HTTP GET メソッドが実行できます。要求メッセージは Web サービス名、タスクを実行するためのリソースの名前とネットワークの場所、および出力をフィルタリングするためのパラメータを含む文字列です。

リソースマッピング

Web サービスクライアントに返すデータを返すマッピング。デフォルトまたはカスタムのリソースマッピングを作成できます。デフォルトのリソースマッピングには、同じポートを設定した読み取りトランスフォーメーションと出力トランスフォーメーションが含まれています。カスタムマッピングには、読み取りトランスフォーメーションおよび出力トランスフォーメーションと一緒に他のトランスフォーメーションが含まれる場合があります。カスタムリソースマッピングには、出力トランスフォーメーションとは異なるポートを設定した読み取りトランスフォーメーションが含まれている場合があります。

応答メッセージ

Web サービスクライアントに返すデータが含まれている JSON または XML ファイル。応答メッセージには、要素および複数出現データの階層を含めることができます。

REST Web サービスの処理

REST Web サービスでは、外部アプリケーション、Web ブラウザ、REST Web サービスのコンシューマトランスフォーメーションなどの Web サービスクライアントからの要求が処理されます。

以下のプロセスは、Web サービスクライアントからの Web サービス要求が、データ統合サービスでどのように処理されるかを示しています。

1. データ統合サービスは、Web サービスクライアントから要求を受け取ります。
2. データ統合サービスの REST Web サービスモジュールでは、リソースマッピングを実行し、データオブジェクトから行を取得して、要求が処理されます。
3. データ統合サービスでは、要求にフィルタパラメータが含まれている場合、出力行がフィルタリングされます。
4. REST Web サービスモジュールによって応答メッセージは Web サービスクライアントに送信されます。

Web サービスコンシューマトランスフォーメーションの処理

外部アプリケーションや Web サービスコンシューマトランスフォーメーションは、Web サービスクライアントとして Web サービスに接続できます。

以下のプロセスは、Web サービスコンシューマトランスフォーメーションがどのように要求を送信し、Web サービスからの応答を受信するかを示しています。

1. Web サービスコンシューマトランスフォーメーションが要求を生成し、接続オブジェクトがある Web サービスに接続します。
2. Web サービスコンシューマトランスフォーメーションが Web サービスから応答を受信します。
3. Web サービスコンシューマトランスフォーメーションが応答からデータを抽出し、トランスフォーメーション出力ポートでデータを返します。

REST Web サービスリソース

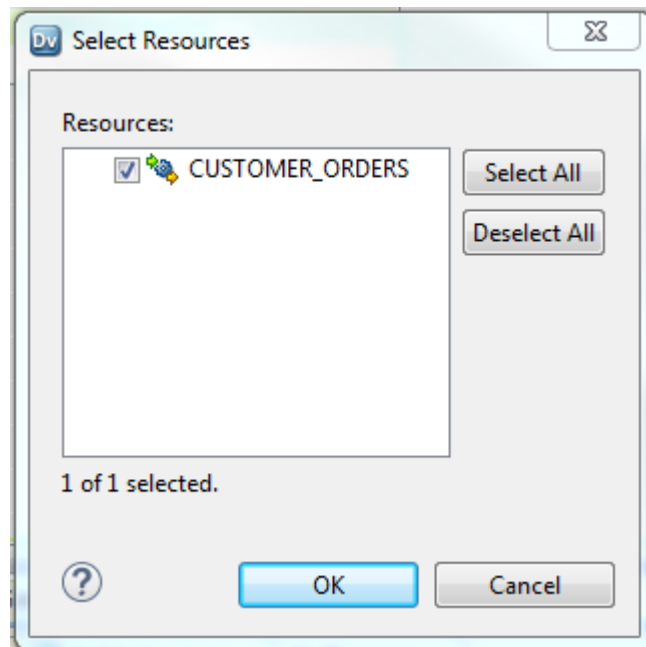
リソースには、応答を返すための REST 応答メッセージおよびデータアクセス方式の定義が含まれています。Informatica REST Web サービスの作成時、手動でリソースを定義するか、データオブジェクトからリソースを作成できます。

リレーショナルデータオブジェクトまたはフラットファイルデータオブジェクトからリソースを作成できます。データオブジェクトからリソースを作成する場合、Developer tool によって、読み取りトランスフォーメーションと出力トランスフォーメーションを設定したデフォルトリソースマッピングが作成されます。出力トランスフォーメーションには、読み取りトランスフォーメーションと同じポートが含まれています。

出力応答メッセージの構造を手動で定義できます。リソースを手動で作成する場合は、応答メッセージで要素を定義します。次に、データを応答メッセージに返すマッピングを定義します。

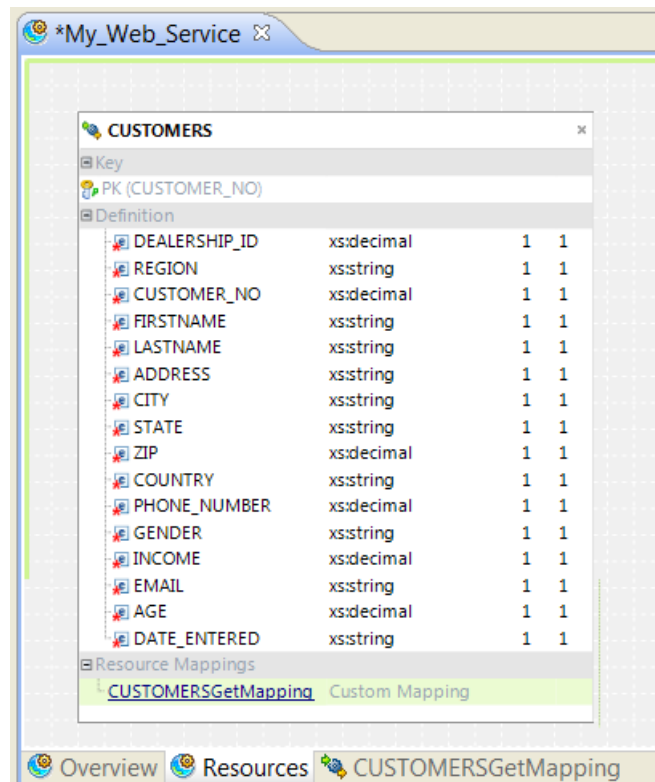
REST Web サービスには複数のリソースを設定できます。各リソースには、データおよび出力の応答の定義を取得するためのリソースマッピングがあります。[リソース] タブで、リソースを選択して表示できます。エディタで右クリックして、**[リソースの表示]** を選択します。

次の図は、[リソースの選択] ダイアログボックスです。



[リソース] タブでリソースのコンポーネントを表示できます。

次の図は、REST Web サービスの **【リソース】** タブのリソースコンポーネントです。



リソースには次のコンポーネントが含まれます。

キー

応答メッセージのデータのインデックス。Web サービスクライアントは特定のキーのデータを要求できます。出力で任意の単純型要素をキーとして設定できます。

定義

出力応答メッセージの要素。リソースの定義を展開するか、**【リソース】** プロパティの **【スキーマ】** ビューに移動して、要素を表示できます。

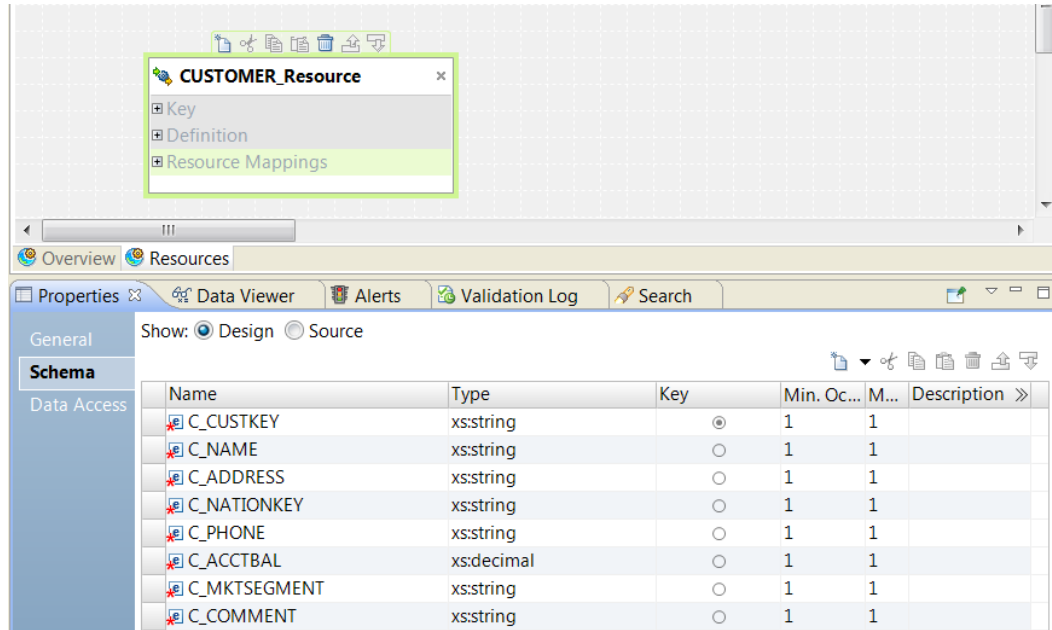
リソースマッピング

データを取得し、変換し、応答メッセージで返すマッピング。デフォルトでは、リソースマッピングには読み取りトランスフォーメーションと出力トランスフォーメーションが含まれています。任意のトランスフォーメーションをマッピングに追加できます。リソースのリンクをクリックして、リソースマッピングを表示します。

REST Web サービスのスキーマビュー

【リソース】プロパティの【スキーマ】ビューで REST リソース定義の構造を表示または変更します。

次の図は、【スキーマ】ビューです。



スキーマの【デザイン】または【ソース】ビューを選択できます。【ソース】ビューを選択して XML 形式のスキーマを表示します。

【デザイン】ビューを使用して、スキーマ内の要素を変更するか、要素型を変更します。要素の順序は変更が可能です。【デザイン】ビューでキーを変更できます。複数出現フィールドを定義することもできます。フィールドが出現する必要がある回数の最小値と最大値を入力します。【バインドなし】を選択すると、出現回数が無制限の複数出現フィールドを作成できます。

スキーマの形式を変更すると、出力トランスフォーメーションにその変更が表示されます。出力トランスフォーメーションでポートを更新する必要がある場合があります。

データオブジェクトの同期

リソースにデフォルトマッピングが含まれる場合は、Developer tool でリソース定義をデータオブジェクトに同期できます。

リソースキーがデータオブジェクトのプライマリキーと異なる場合、Developer tool はリソース定義をデータオブジェクトと同期できません。

リソースキー

リソース定義でキーを定義できます。キーを定義すると、データ統合サービスは出力データのカラムをインデックス処理します。Web サービスクライアントによって、キーで特定の出力行を要求できます。

データオブジェクトからリソースを作成すると、デフォルトでは Developer tool によってデータオブジェクトのプライマリキーがリソースキーとして使用されます。プライマリキーに複数のカラムが含まれる場合、Developer tool によってカラムデータが「+」で区切られ、複数のプライマリキーが作成されます。

Developer tool を設定して、ソースのプライマリキーとは異なる出力カラムをプライマリキーとして使用できます。

手動でリソースを作成する場合、リソースキーは単純型の要素である必要があります。

リソースマッピング

リソースマッピングはデータオブジェクトを読み取り、REST 応答メッセージにデータを返すマッピングです。Informatica リソースマッピングでは、GET メソッドが実行されます。このマッピングは Web サービス要求メッセージは解析されません。

リソースマッピングには、クライアント要求のデータをフィルタリングするフィルタトランスフォーメーションやルックアップトランスフォーメーションは含まれません。Web サービス要求メッセージにフィルタクエリが含まれている場合、マッピングでデータを取得後、データ統合サービスによってフィルタリングが行われます。結果セットキャッシュプロパティが Administrator ツールでデータ統合サービスに対して有効になっている場合、データ統合サービスは応答メッセージのデータをキャッシュします。Web サービス要求にフィルタパラメータが含まれている場合、データ統合サービスによってキャッシュの行がフィルタリングされます。

リソースマッピングを定義する場合、リソースマッピングの次のタイプのいずれかを作成します。

デフォルトリソースマッピング

デフォルトリソースマッピングには、読み取りトランスフォーメーションと出力トランスフォーメーションが含まれています。このマッピングには他のトランスフォーメーションは含まれていません。マッピングはデータオブジェクト内のすべての行を変更せずに返します。デフォルトリソースマッピングを作成するには、データオブジェクトから Web サービスを作成するか、Web サービスとしてデータオブジェクトをデプロイします。

カスタムリソースマッピング

カスタムリソースマッピングはデフォルトマッピング以外のすべてのマッピングです。カスタムリソースマッピングには、出力トランスフォーメーションとは異なるカラムを持つ読み取りトランスフォーメーションが含まれる場合があります。カスタムリソースマッピングには、マッピングの読み取りトランスフォーメーションと出力トランスフォーメーション間にトランスフォーメーションが含まれている場合があります。このトランスフォーメーションでは、パイプラインにカラムを追加したり、カラムを変更したりすることができます。

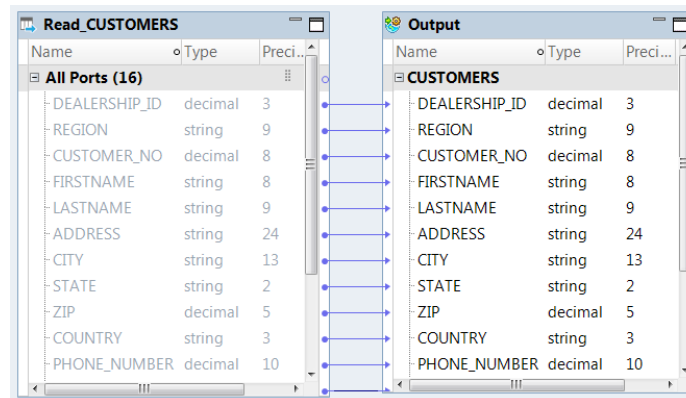
例えば、リソースマッピングに、2つのカラムを結合して第3のカラムを返す式トランスフォーメーションが含まれる場合があります。または、複数出現の行を返すトランスフォーメーションが含まれる場合があります。リソースを手動で作成するとき、カスタムリソースマッピングを作成できます。読み取りトランスフォーメーションを変更して、またはマッピングにトランスフォーメーションを追加して、デフォルトのマッピングを変更することもできます。

デフォルトリソースマッピング

デフォルトリソースマッピングには、読み取りトランスフォーメーションと出力トランスフォーメーションが含まれています。データオブジェクトから Web サービスを作成し、作成した Web サービスを変更しない場合は、デフォルトリソースマッピングを作成します。

データオブジェクトからリソースを作成するとき、ウィザードにより、読み取りトランスフォーメーションと出力トランスフォーメーションを含むデフォルトのリソースマッピングが作成されます。読み取りトランスフォーメーションは、リソース作成元のデータオブジェクトを読み取ります。出力トランスフォーメーションには、デフォルトのリソースマッピングの読み取りトランスフォーメーションと同じカラムが含まれます。

次の図は、顧客データオブジェクトのすべてのカラムを返すデフォルトリソースマッピングを示しています。



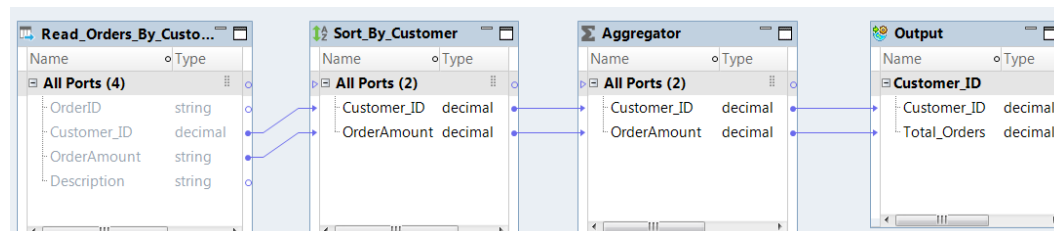
このリソースマッピングには、フィルタトランスフォーメーションやルックアップトランスフォーメーションは含まれておらず、データはフィルタリングされません。ただし、REST Web サービスクライアントは、フィルタパラメータを含む要求を、デフォルトリソースマッピングを使用する Web サービスに送信できます。データ統合サービスは、リソースマッピングで生成された出力データからのデータをフィルタリングします。

カスタムリソースマッピング

カスタムリソースマッピングには、出力トランスフォーメーションとは異なるポートを持つ読み取りトランスフォーメーションを含めることができます。カスタムリソースマッピングには、読み取りトランスフォーメーションと出力トランスフォーメーション以外のトランスフォーメーションを含めることもできます。

カスタムリソースマッピングは、デフォルトのリソースマッピングを変更するか、手動で REST Web サービスを作成することで、作成できます。

次の図は、カスタムリソースマッピングを示しています。



マッピングには次のトランスフォーメーションが含まれます。

読み取りトランスフォーメーション

注文のファイルを読み取ります。各注文には、顧客 ID が含まれています。顧客 ID は、複数回出現することができます。

ソータートランスフォーメーション

顧客 ID で注文をソートします。

アグリゲータトランスフォーメーション

顧客ごとに注文総額を合計します。

出力トランスフォーメーション

顧客別に注文総額を返します。

カスタムリソースマッピングを作成するには、REST Web サービスリソース定義を手動で定義します。リソース定義を手動で定義するときは、応答メッセージの要素を定義します。この例では、応答メッセージに顧客 ID と注文総額のみが含まれています。

リソース定義を定義すると、Developer tool により、出力トランスフォーメーションを含むリソースマッピングが作成されます。次に、読み取りトランスフォーメーションとその他のトランスフォーメーションをマッピングに追加します。

前の図の顧客 ID は、キーです。Web サービスクライアントが特定の顧客の注文数を要求する場合があります。データ統合サービスは、出力データをキーでフィルタリングします。このマッピングには、フィルタトランスフォーメーションは含まれません。

REST Web サービスの出力トランスフォーメーション

出力トランスフォーメーションでは、操作マッピングのリレーショナルデータのグループから REST Web サービスの応答メッセージが作成されます。REST Web サービスを定義するとき、Developer tool によって出力トランスフォーメーションが作成されます。

REST Web サービスを作成するとき、Developer tool によって、定義済みのリソース定義に基づいて出力トランスフォーメーションが作成されます。Developer tool は応答メッセージ構造に基づいてそのトランスフォーメーションの入力ポートを作成します。出力トランスフォーメーションには、入力ポートを応答メッセージのノードにマップするマッピングが含まれています。

スキーマを変更して出力メッセージ階層の要素を変更できます。スキーマの変更に対応するため、トランスフォーメーションの入力ポートを変更できます。

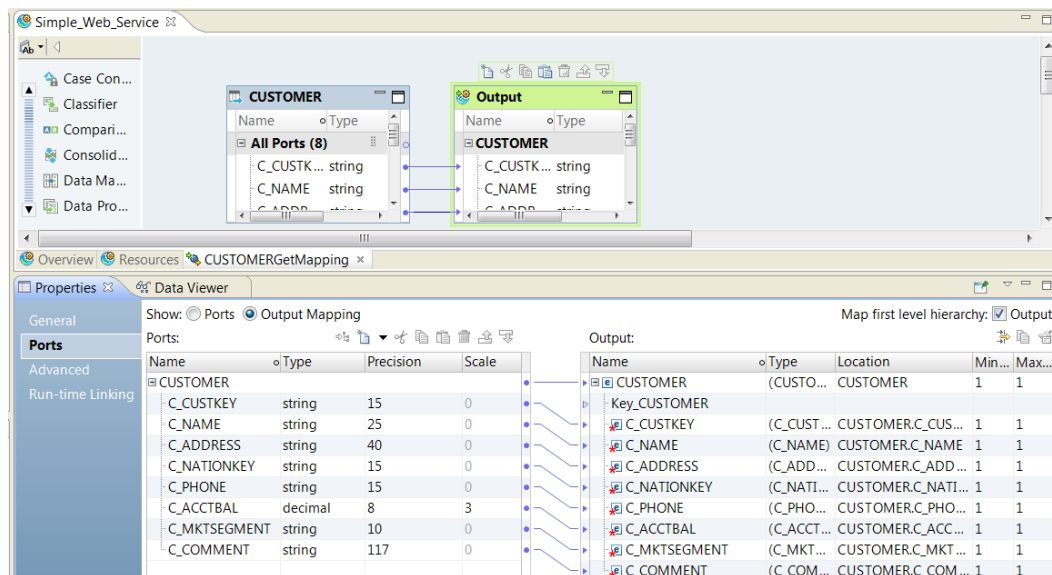
出力トランスフォーメーションの【ポート】タブで REST 応答メッセージ階層を表示します。トランスフォーメーションポートの表示または、トランスフォーメーション入力ポートと応答メッセージ階層間のマッピングの表示を選択できます。

ポートを表示しているときは、グループおよびポートを手動で追加するか、または他のトランスフォーメーションから出力トランスフォーメーションにポートをコピーできます。その場合、キーボードショートカットを使用するか、または Developer tool でコピーと貼り付けのボタンを使用します。

出力マッピングを表示しているとき、入力グループの定義、入力ポートの定義、および応答メッセージ要素への入力ポートのマップを行うことができます。タブの左側は【ポート】領域、タブの右側は【出力】領域です。【出力】領域に、応答メッセージ階層が表示されます。【ポート】領域で入力グループおよびポートを定義します。【ポート】領域から【出力】領域のノードに入力ポートをマップすると、【出力】領域の【場所】カラムに入力ポートの場所が表示されます。

出力マッピングを表示する場合、入力ポートと操作入力内のノードを結ぶ線を表示するように選択できます。

次の図は、出力トランスフォーメーションの単純な出力マッピングです。



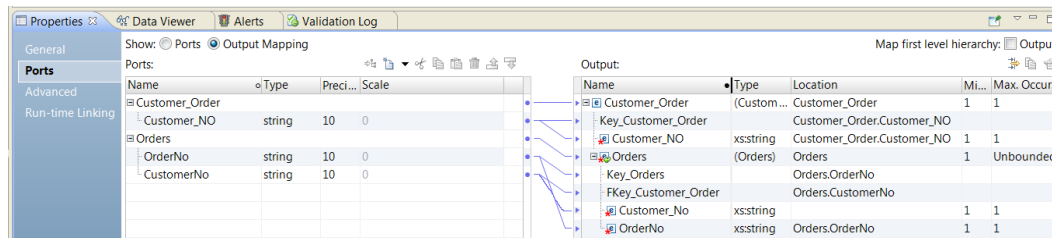
この図では、トランスフォーメーションは1人の顧客について1つのポートグループを受信します。1回出現する要素が返されます。

階層の第1レベルをマップすることを選択すると、Developer toolによって操作出力の第1レベルのノードに入力ポートがマップされます。**[第1レベル階層のマップ]** オプションはデフォルトで有効になっています。また、データをマップするために必要な入力ポートが作成されます。階層の第1レベルに、複数出現の子ノードを1つ以上持つ複数出現の親ノードが含まれている場合は、Developer toolでポートの作成または階層の第1レベルのマッピングが行われません。

REST 出力トランスフォーメーションの複数出現データ

REST 出力トランスフォーメーションでは複数グループのデータを受信し、応答メッセージで複数レベルの出力データを返すことができます。応答メッセージのデータのグループは、それぞれキーによってリンクされています。

次の図は、1人の顧客の複数の注文を返す出力マッピングです。



出力マッピングの**【ポート】**領域には、Customer_OrderおよびOrdersの2つのグループがあります。各グループには顧客番号が含まれています。顧客番号は応答メッセージで顧客データと注文データをリンクするキーになります。

リソース定義のプライマリキーは、Customer_Noです。Customer_NoはCustomer_Orderのキーです。Ordersでは外部キーです。プライマリキーには一意の値が含まれている必要があります。データ統合サービスが複数出現の注文を処理する場合、キーに基づいて1人の顧客のすべての注文を返すことができます。プライマリキーと外部キーは応答メッセージに表示されません。

出力マッピングの**【出力】**領域には応答メッセージが表示されます。入力ポートを出力の各キーにリンクする必要があります。

注: REST 出力トランスフォーメーションで重複したプライマリキー値が受信されると、エラーが発生してレビューが失敗します。

要求メッセージ

Web サービスクライアントによって、URI 文字列内の要求メッセージが Web サービスに送信されます。URI によって、Web サービスでアクセスするホストポート、Web サービス名、およびリソースが識別されます。要求メッセージにクエリパラメータを追加して、特定の行の Web サービス出力をフィルタリングできます。

URI に識別子やパラメータがない場合、REST Web サービスは REST Web サービスの応答のデータオブジェクトからすべての行のリストを返します。Web サービスによって、すべての顧客行を含む JSON ファイルまたは XML ファイルが返されます。

次の例は、すべての顧客のリストを返す要求です。

http://myhost:8095/DataIntegrationService/RESTSERVICE/Rsrc_CUSTOMER/CUSTOMER

次のテキストは JSON 応答メッセージです。

```
{"CUSTOMERS": {"CUSTOMER": [  
  {  
    "C_ACCTBAL": 9331.13,  
    "C_ADDRESS": "38 Summit Drive",  
    "C_COMMENT": "Call immediately if delay",  
    "C_CUSTKEY": {  
      "@url": "http:\u002F\u002FHostName:8095\u002FDataIntegrationService\u002FRestService  
\u002FRsrc_CUSTOMER\u002FCUSTOMER\u002F63",  
      "$": 63  
    },  
    "C_MKTSEGMENT": "AUTOMOBILE",  
    "C_NAME": "Customer#000000063",  
    "C_NATIONKEY": 21,  
    "C_PHONE": "31-952-552-9584"  
  },  
  {  
    "C_ACCTBAL": -646.64,  
    "C_ADDRESS": "44 Ocean Avenue",  
    "C_COMMENT": "Has dangerous animal in the house",  
    "C_CUSTKEY": {  
      "@url": "http:\u002F\u002FHostName:8095\u002FDataIntegrationService\u002FRestService  
\u002FRsrc_CUSTOMER\u002FCUSTOMER\u002F64",  
      "$": 64  
    },  
    "C_MKTSEGMENT": "BUILDING",  
    "C_NAME": "Customer#000000064",  
    "C_NATIONKEY": 3,  
    "C_PHONE": "13-558-731-7204"  
  }  
]}]}
```

リソースマッピングでのデータのフィルタリング

要求メッセージにクエリパラメータが含まれている場合、データ統合サービスは、クエリパラメータの値を使用して、出力データをフィルタリングします。

データ統合サービスは、デフォルトリソースマッピングおよびカスタムリソースマッピングで出力行をフィルタリングできます。

要求メッセージクエリには、次の条件を含めることができます。

<> = != >= <=

URI でパラメータを設定するには、次の句を含めます。

```
?filtercondition=<column name> <operand><value>
```

例えば、次の要求メッセージでは、市場区分の値が「BUILDING」のすべての顧客行が検索されます。

```
http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/?  
filterCondition=C_MKTSEGMENT='BUILDING'
```

カラムのデータ型が文字列の場合は、検索値を単一引用符 (') で囲みます。カラムのデータ型が数値の場合は、検索値を引用符で囲まないでください。

例えば、次の要求には、数値の検索パラメータが含まれています。

```
http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/?  
filterCondition=C_ACCTBAL=9331.13
```

REST Web サービスは、貯金残高が 9331.13 の顧客を返します。

次のテキストは、XML 形式の REST 応答メッセージを示しています。

```
<CUSTOMERs>  
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">  
<tns:C_CUSTKEY url="http://myServer:8095/DataIntegrationService/RestService/REST_Web_Service/CUSTOMER/  
63">63</tns:C_CUSTKEY>  
<tns:C_NAME>Customer#000000063</tns:C_NAME>  
<tns:C_ADDRESS>IXRSpVwWZraKII</tns:C_ADDRESS>  
<tns:C_NATIONKEY>21</tns:C_NATIONKEY>  
<tns:C_PHONE>31-952-552-9584</tns:C_PHONE>  
<tns:C_ACCTBAL>9331.13</tns:C_ACCTBAL>  
<tns:C_MKTSEGMENT>AUTOMOBILE</tns:C_MKTSEGMENT>  
<tns:C_COMMENT>Apply discount</tns:C_COMMENT>  
</tns:CUSTOMER>  
</CUSTOMERs>
```

複数のパラメータをクエリに含める場合は、「AND」でパラメータを結合します。

```
http://usw1mj02ee4j:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service0/CUSTOMER/?  
filterCondition=C_ACCTBAL=9331.13 AND C_NATIONKEY='21'
```

カラムのデータ型が日付/時間の場合は、URI のパラメータ文字列を日付/時間形式に変換し、パラメータ文字列の形式を指定できます。

例えば、次の文は、パラメータ文字列を日付/時間形式に変換します。

```
?filterCondition= O_ORDERDATE=TO_DATE('1994-11-17 00:00:00.000000000','YYYY-MM-DD HH24:MI:SS.NS')
```

キーで検索

リソースキーに対する REST Web サービスクエリを設定できます。要求にキー値が含まれる場合、クエリでカラム名を参照する必要はありません。

次の形式を使用して、リソース「CUSTOMER」のリソースキーにある番号「64」を検索します。

```
http://myServer:8095/DataIntegrationService/RESTSERVICE/REST_Web_Service/CUSTOMER/64
```

この URI にはリソース名とリソースキー値が含まれます。

```
/CUSTOMER/64
```

このクエリで、キーとなるカラムの名前を参照する必要はありません。

リソースキーで問い合わせると、Web サービスはキーに関連付けられた URI によって顧客を取得できます。応答メッセージのキー値で特定の行の URI を表示できます。次の例で、キーは C_CUSTKEY であり、キー値は 64 です。

```
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">  
<tns:C_CUSTKEY url="http://my Server:8095/DataIntegrationService/RestService/REST_Web_Service/CUSTOMER/
```



```

64">64</tns:C_CUSTKEY>
<tns:C_NAME>Customer#000000064</tns:C_NAME>
<tns:C_ADDRESS>MbCeGY20kaKK3oa1JD,OT</tns:C_ADDRESS>
<tns:C_NATIONKEY>3</tns:C_NATIONKEY>
<tns:C_PHONE>13-558-731-7204</tns:C_PHONE>
<tns:C_ACCTBAL>-646.64</tns:C_ACCTBAL>
<tns:C_MKTSEGMENT>BUILDING</tns:C_MKTSEGMENT>
<tns:C_COMMENT>
Customer has an angry dog in the yard
</tns:C_COMMENT>
</tns:CUSTOMER>

```

応答メッセージ形式

REST Web サービスを JSON または XML 形式で応答メッセージを返すように設定できます。

応答メッセージでは、C_CUSTKEY フィールドがリソースキーです。C_CUSTKEY フィールドにはキー値と、アプリケーションで出力データの特定キーによる顧客へのアクセスに使用できる、キー値への URL が含まれています。

JSON 形式では、URL のバックスラッシュは\u002f に Unicode エンコードされます。

URL のリソースキー値にも、次の理由のため、エンコードされた値が含まれている場合があります。

- キー値に `http://www.informatica.com` などの URL アドレスが含まれている。
- キー値に「20%」のようにパーセント記号 (%) が含まれている。
- キー値にスペースが含まれている。

次のテキストは JSON 形式の応答メッセージです。

```

{"CUSTOMERS": {"CUSTOMER": [
  {
    "C_ACCTBAL": 9331.13,
    "C_ADDRESS": "IXRSpVwWZraKII",
    "C_COMMENT": "Good customer",
    "C_CUSTKEY": {
      "@url": "http:\u002f\u002fHostName:8095\u002fDataIntegrationService\u002fSimple_Web_Service
\u002fRsrc_CUSTOMER\u002fCUSTOMER\u002f63",
      "$": 63
    },
    "C_MKTSEGMENT": "AUTOMOBILE",
    "C_NAME": "Customer#000000063",
    "C_NATIONKEY": 21,
    "C_PHONE": "31-952-552-9584"
  }
]}

```

次のテキストは XML 形式の応答メッセージです。

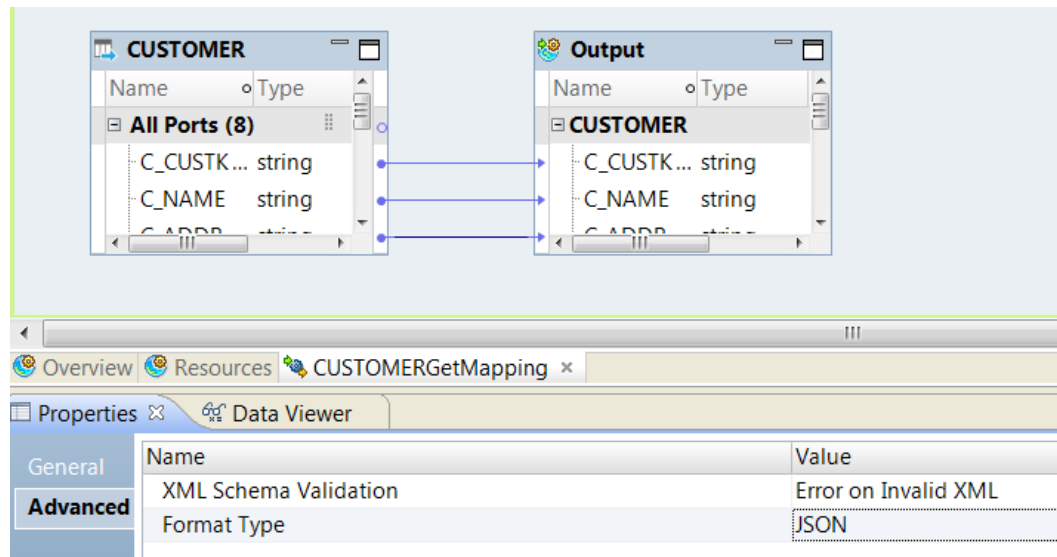
```

<?xml version="1.0" encoding="UTF-16LE" standalone="no"?>
<tns:CUSTOMER xmlns:tns="http://www.informatica.com">
  <tns:C_CUSTKEY url="Simple_Web_Service/CUSTOMER/63">63</tns:C_CUSTKEY>
  <tns:C_NAME>Customer#000000063 </tns:C_NAME>
  <tns:C_ADDRESS>IXRSpVwWZraKII </tns:C_ADDRESS>
  <tns:C_NATIONKEY>21</tns:C_NATIONKEY>
  <tns:C_PHONE>31-952-552-9584</tns:C_PHONE>
  <tns:C_ACCTBAL>9331.13</tns:C_ACCTBAL>
  <tns:C_MKTSEGMENT>AUTOMOBILE</tns:C_MKTSEGMENT>
  <tns:C_COMMENT>Good customer
</tns:C_COMMENT>
</tns:CUSTOMER>

```


応答メッセージの形式を設定できます。リソースマッピングの【プロパティ】ビューの【詳細】タブで応答メッセージの形式を変更します。【形式の種類】で JSON または XML を選択します。

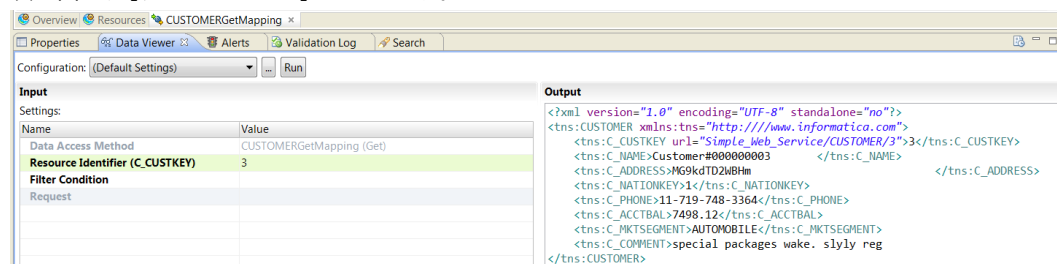
次の図は、リソースマッピングの【プロパティ】ビューの【詳細】タブです。



応答のデータプレビュー

Developer tool の【データビューア】ビューで Web サービスをテストできます。リソースキーでデータをフィルタリングするか、式を設定してデータをフィルタリングできます。

次の図は、【データビューア】ビューです。



次の入力設定を行って、【データビューア】ビューで応答メッセージをフィルタリングします。

データアクセス方式

実行するリソースマッピング。Web サービスには複数のリソースマッピングを含めることができます。

リソース識別子

検索に使用するキー値。Developer tool はリソースキーの名前を識別します。

フィルタ条件

式エディタで設定できるフィルタ式。式では複数の出力カラムを参照できます。出力に階層データがある場合、フィルタ条件では親グループの要素を参照する必要があります。フィルタ条件とリソース識別子を同時に入力できます。

第 11 章

REST Web サービスの作成方法

この章では、以下の項目について説明します。

- [REST Web サービスの作成, 122 ページ](#)
- [REST Web サービスを手動で作成する方法, 123 ページ](#)
- [手順 1. REST Web サービスリソースの作成, 123 ページ](#)
- [手順 2. リソースマッピングの定義, 125 ページ](#)
- [手順 3. 出力マッピングの設定, 130 ページ](#)
- [手順 4. \[データビューア\] ビューでのマッピングのテスト, 131 ページ](#)
- [手順 5. アプリケーションのデプロイ, 133 ページ](#)
- [手順 6. ブラウザからの Web サービスのクエリ, 135 ページ](#)
- [データオブジェクトから REST Web サービスを作成する方法, 137 ページ](#)
- [データオブジェクトを REST Web サービスとしてデプロイする方法, 140 ページ](#)

REST Web サービスの作成

データオブジェクトを REST Web サービスとしてデプロイすることも、ウィザードで Web サービスを定義することもできます。

Web サービスを作成する前に、応答メッセージに含める要素を決定します。応答メッセージの構造に基づいて Web サービスを作成するには、次のいずれかの方法を使用します。

手動で Web サービスを作成する。

リソースマッピングで計算またはフォーマットする必要があるデータが応答メッセージに含まれる場合は、手動で Web サービスを作成します。複数のソースがマッピングに含まれる場合は、手動で Web サービスを作成します。階層データまたは複数出現データが応答メッセージに含まれる場合は、手動で構造を定義する必要があります。

データオブジェクトから Web サービスを作成する。

データオブジェクトからのデータが応答メッセージに含まれる場合は、そのデータオブジェクトに基づいて応答定義を作成できます。Web サービスを作成した後、データオブジェクトから特定の列を選択して、リソースマッピングを変更できます。

データオブジェクトを Web サービスとしてデプロイする。

データオブジェクトを Web サービスとしてデプロイするのは、データオブジェクトからのすべての列を応答メッセージに含めることができる場合です。

REST Web サービスを手動で作成する方法

REST Web サービスリソースのカラムを手動で定義できます。リソースを手動で定義すると、Developer tool によって、応答メッセージを定義する出力トランスフォーメーションを設定したリソースマッピングが作成されます。Developer tool では、リソースマッピングに読み取りトランスフォーメーションは作成されません。リソースの定義後にマッピングに読み取りトランスフォーメーションを追加する必要があります。応答メッセージ構造がソースデータと異なる場合は、手動でリソースを作成します。

次の手順を使用して、REST Web サービスを作成します。

1. REST Web サービスリソースを作成します。
2. データを取得するリソースマッピングを定義します。
3. REST 出力トランスフォーメーションマッピングを設定して、入力ポートから出力階層の要素にデータをマップします。
4. マッピングをテストします。
5. 作成したアプリケーションをデータ統合サービスにデプロイします。
6. ブラウザから Web サービスにアクセスします。

REST Web サービスの例

Hypostores では、顧客のすべての注文を返す REST Web サービスを利用しています。販売員は、Web ブラウザから Web サービスに注文の要求を送信できます。要求には、1 つ以上の顧客番号が含まれています。Web サービスは、各顧客名と顧客のすべての注文リストを JSON ファイルで返します。

REST Web サービスには、顧客テーブルを読み取って顧客名を取得するマッピングが含まれています。マッピングには、ルックアップトランスフォーメーションが含まれています。ルックアップトランスフォーメーションは、各顧客のすべての注文を注文テーブルから取得します。マッピングには、階層構造の JSON ファイルを返す REST 出力トランスフォーメーションが含まれています。JSON ファイルの構造では、親グループに顧客番号と顧客名が含まれています。JSON ファイルでは、顧客内に注文の子グループが含まれています。子グループには、複数出現の注文が含まれています。各注文には、注文番号、価格、および注文日が含まれています。

手順 1. REST Web サービスリソースの作成

REST Web サービスリソースには、REST Web サービスの応答メッセージの定義と、応答を返すためのデータアクセス方式が含まれます。Informatica REST Web サービスを作成するときに、データオブジェクトからリソースを定義するか、手動でリソースを定義することができます。

この例では、応答メッセージの形式が階層的で複数出現データが含まれるため、リソースを手動で作成します。リポジトリのデータオブジェクトからリソースを作成することはできません。

REST Web サービスリソースの作成

Web サービスリソースを作成するとき、応答メッセージの構造を定義します。

1. Developer tool で、[ファイル] > [新規] > [データサービス] をクリックします。
2. [REST Web サービス] を選択し、[次へ] をクリックします。
3. Web サービス名として「Orders_Web_Service」と入力します。[次へ] をクリックします。
4. [REST リソース] ダイアログボックスで、[空の状態から作成] をクリックします。

デフォルトのリソースが表示されます。

REST Resource
Create REST resources and define mappings on REST resources.

Name: type filter text
Resource

Name: Resource
Description:

Definition:

| Name | Type | Key | Min. Occurs | Max. Occurs | Description |
|------|------|-----|-------------|-------------|-------------|
|------|------|-----|-------------|-------------|-------------|

HTTP method(s):
☒ Get

< Back Next > Finish Cancel

5. **【名前】** フィールドで、リソース名を [Resource] から [Orders_Resource] に変更します。
6. **【定義】** パネルで、**【新規】** > **【要素】** をクリックします。
7. 次の要素を入力します。

| 名前 | タイプ | キー | 最小出現数 | 最大出現数 |
|---------------|--------|----|-------|--------|
| Customer_Key | string | はい | 1 | 1 |
| Customer_Name | string | | 1 | 1 |
| Orders | string | | 1 | バインドなし |

8. [Orders] 要素を選択し、**【新規】** > **【子】** をクリックします。
9. [Orders] の下に次の子要素を入力します。

| 名前 | タイプ | キー | 最小出現数 | 最大出現数 |
|-------------|---------|----|-------|-------|
| Order_Key | integer | | 1 | 1 |
| Order_Price | decimal | | 1 | 1 |
| Order_Date | integer | | 1 | 1 |

10. Get HTTP メソッドが有効であることを確認します。

11. **【完了】** をクリックします。

次の図は、[Orders_Resource] の要素を示しています。

REST Resource
Create REST resources and define mappings on REST resources.

Name: type filter text
Orders_Resource

Name: Orders_Resource
Description: Returns orders by customer

Definition:

| Name | Type | Key | Min... | Max. Occurs | Description |
|---------------|-----------|-----|--------|-------------|-------------|
| Customer_Key | xs:string | ⊙ | 1 | 1 | |
| Customer_Name | xs:string | ○ | 1 | 1 | |
| Orders | (Orders) | ○ | 1 | Unbounded | |
| Order_Key | xs:int... | ○ | 1 | 1 | |
| Order_Price | xs:dec... | ○ | 1 | 1 | |
| Order_Date | xs:date | ○ | 1 | 1 | |

HTTP method(s):
☒ Get

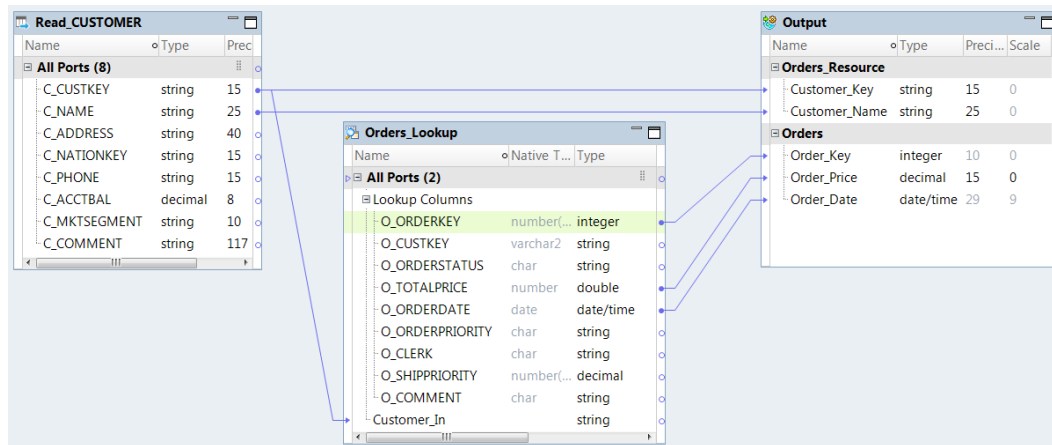
< Back Next > Finish Cancel

手順 2. リソースマッピングの定義

リソースを作成すると、Developer tool によって REST 出力トランスフォーメーションが含まれるリソースマッピングが作成されます。REST 出力トランスフォーメーションの構造によって、Web サービスがクライアント

トに返す応答メッセージが定義されます。読み取りトランスフォーメーションおよびルックアップトランスフォーメーションをマッピングに追加します。

次の図は、作成するリソースマッピングを示しています。



マッピングには次のオブジェクトが含まれます。

Read_Customer

Read_Customer トランスフォーメーションは顧客テーブルを読み取ります。このトランスフォーメーションはテーブルのすべての顧客行を返します。

Orders_Lookup

このルックアップトランスフォーメーションは各顧客の注文を取得します。

アウトプット

REST 出力トランスフォーメーションは Read_Customer トランスフォーメーションから顧客情報を取得します。Orders_Lookup トランスフォーメーションから注文情報を取得します。REST 出力トランスフォーメーションは、各顧客の注文をリストする階層的な JSON ファイルを生成します。

注: リソースマッピングには、顧客テーブルから取得する顧客を制限するフィルタトランスフォーメーションは含まれていません。

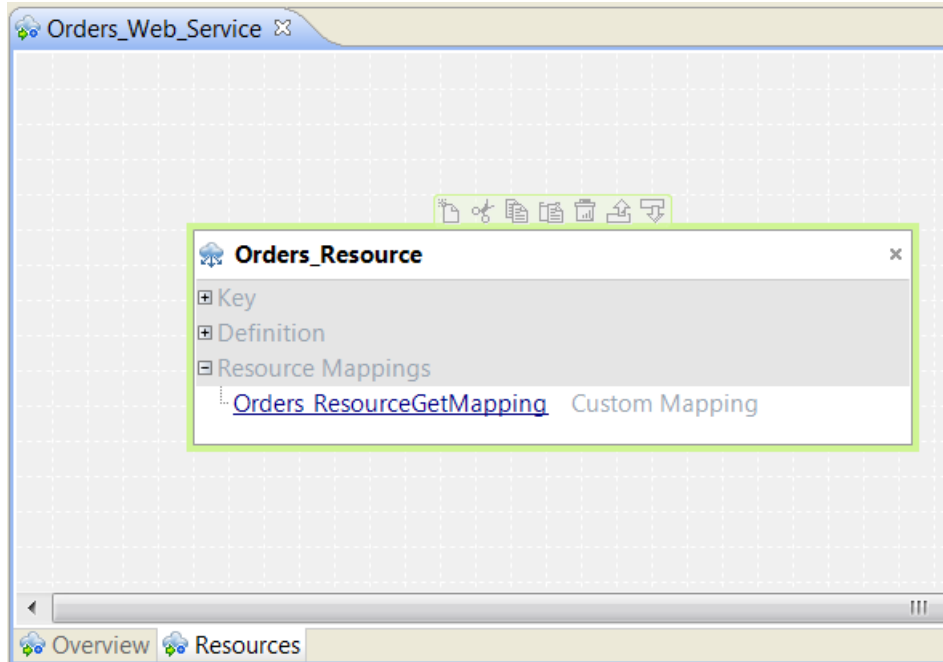
データ統合サービスは、クライアント要求のフィルタに基づいて出力データをフィルタリングします。

リソースマッピングの定義

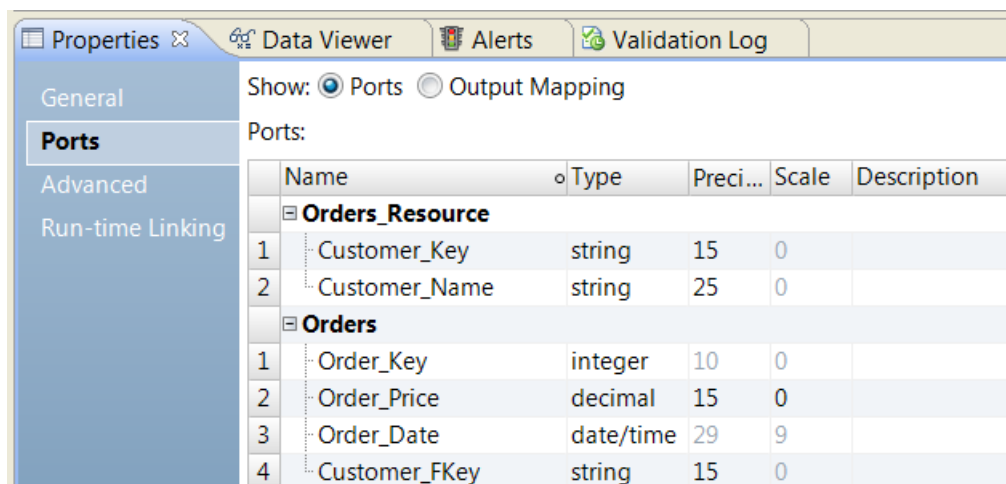
Developer tool により、作成した REST 出力トランスフォーメーションを含むリソースマッピングが作成されます。リソースマッピングの読み取りトランスフォーメーションとルックアップトランスフォーメーションを定義します。

1. **【リソース】** タブで、**[Orders_Resource]** の **【リソースマッピング】** リンクを展開します。

次の図は、**[Orders_Resource]** の **[Orders_Resource]** リンクを示しています。

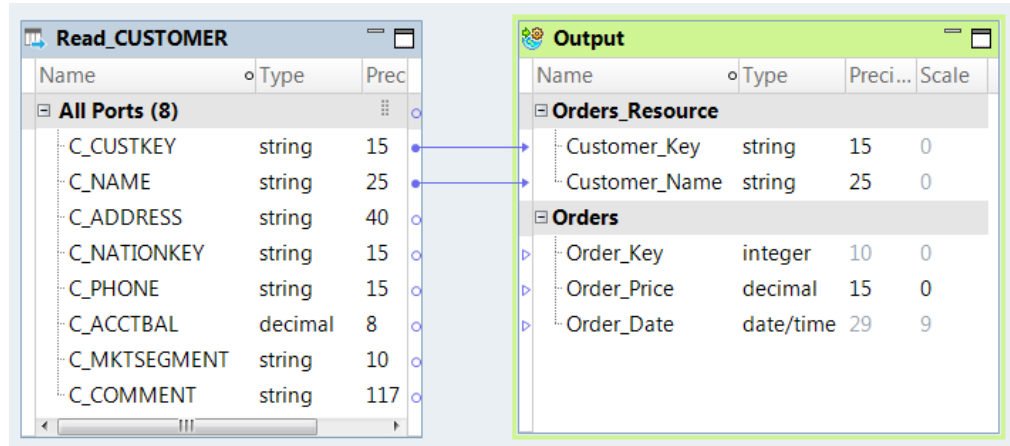


2. **[Orders_Resource]** の **[Orders_ResourceGetMapping]** リンクをクリックします。
マッピングがエディタに表示されます。
3. 顧客データオブジェクトを読み取りトランスフォーメーションとしてマッピングに追加します。
4. **【REST 出力トランスフォーメーション】** をクリックします。
5. REST 出力トランスフォーメーションの **【プロパティ】** タブで、**【ポート】** ビューをクリックします。
6. **[Customer_Key]** の精度を 15 に変更し、**[Customer_Name]** の精度を 25 に変更します。

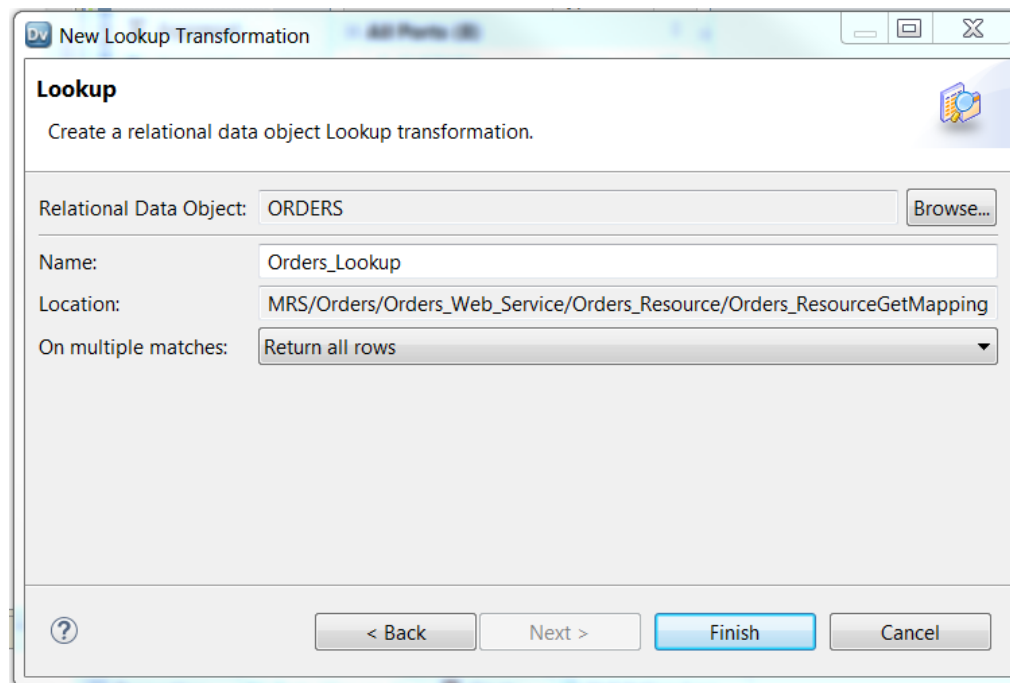


7. [Read_Customer] の顧客キーポートと顧客名ポートを REST 出力トランスフォーメーションに接続します。

次の図は、Read_Customer トランスフォーメーションと REST 出力トランスフォーメーション間のリンクを示しています。

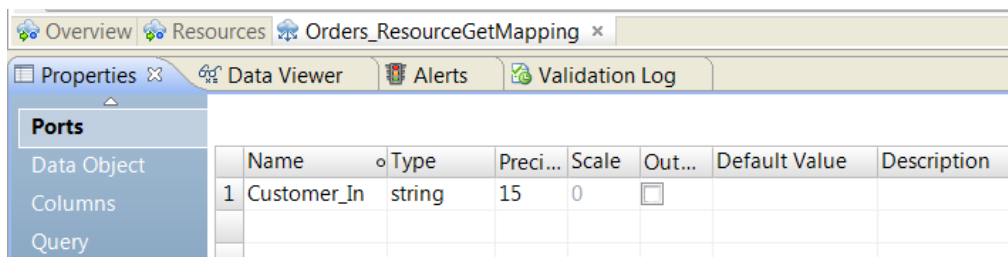


8. ルックアップトランスフォーメーションを追加するには、マッピングを右クリックして、**[トランスフォーメーションの追加]** をクリックします。
9. ルックアップトランスフォーメーションを選択します。
10. [リレーショナルデータオブジェクトルックアップ] を選択して、**[次へ]** をクリックします。
11. **[新しいルックアップトランスフォーメーション]** ダイアログボックスで、注文物理データオブジェクトを参照して選択します。すべての行を返すことを選択します。



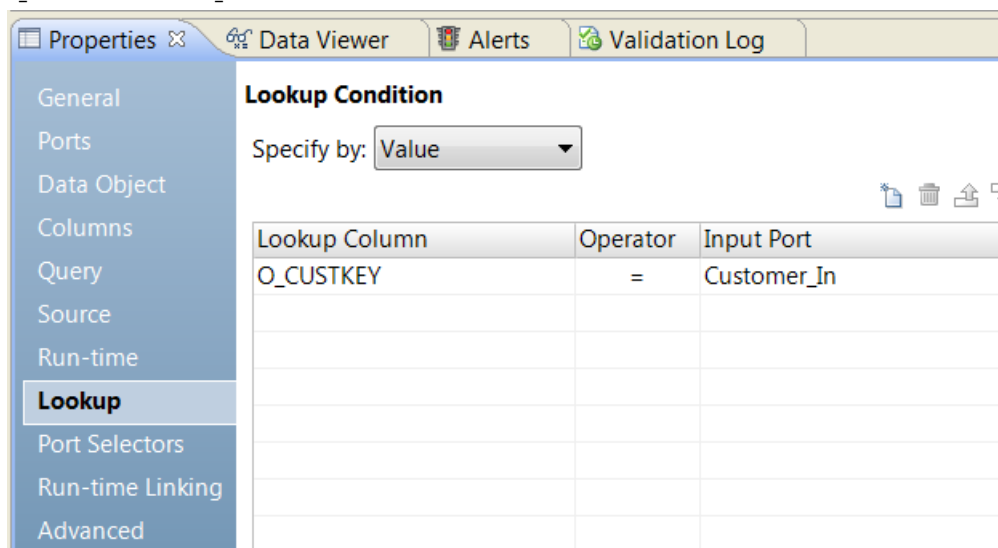
12. **[完了]** をクリックします。
リソースマッピングに [ルックアップトランスフォーメーション] が表示されます。
13. マッピングの [ルックアップトランスフォーメーション] をクリックして選択します。
14. ルックアップトランスフォーメーションの **[プロパティ]** ビューで、**[ポート]** タブを選択します。

15. [Customer_In] という名前のポートを追加します。このポートは、精度が 15 の文字列です。このポートを有効にしなくても、出力は可能です。



16. ルックアップトランスフォーメーションの【プロパティ】ビューで、【ルックアップ】タブをクリックします。
17. 次のルックアップ条件を入力します。

O_CUSTKEY = Customer_In



18. マッピングキャンバスで、[Read_Customer] の [C_CUSTKEY] ポートをルックアップトランスフォーメーションの [Customer_In] ポートにリンクします。
19. REST 出力トランスフォーメーションのポートに、ルックアップトランスフォーメーションの次の出力ポートをリンクします。

| ルックアップトランスフォーメーション | 出力トランスフォーメーション |
|--------------------|----------------|
| O_ORDERKEY | Order_Key |
| O_TOTALPRICE | Order_Price |
| O_ORDERDATE | Order_Date |

手順 3. 出力マッピングの設定

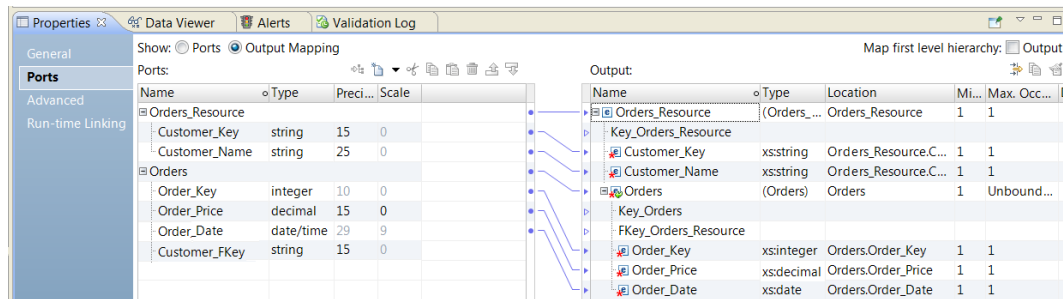
REST 出力トランスフォーメーションは Customer トランスフォーメーションおよび Orders トランスフォーメーションからデータを取得し、JSON ファイルを返します。

Orders_Resource マッピングで、REST 出力トランスフォーメーションには 2 つのグループの出力データが含まれています。親グループには顧客情報が含まれ、各顧客につき 1 回出現します。子グループには注文が含まれています。注文グループは複数出現です。

REST 出力トランスフォーメーションに複数のグループのポートが含まれている場合、Developer tool によってグループにリンクするキーが作成されます。Developer tool によって各グループにプライマリキーが作成されます。Developer tool によって各子グループの外部キーが作成されます。Orders グループの外部キーには、各注文の顧客番号が含まれます。

REST 出力トランスフォーメーションでポートを表示しても、キーは表示されません。キーにはポートがありません。ただし、キーにデータをリンクしてキー値を定義する必要があります。キーを表示するには、REST 出力トランスフォーメーションの【プロパティ】タブの【ポート】ビューで出力マッピングを表示します。【ポート】ビューには、表示するために選択したオプションによって、入力ポートまたは出力マッピングが表示されます。

次の図は、【出力マッピング】ビューを示しています。



Developer tool によって出力で以下のキーが作成されます。

| キー名 | 説明 |
|----------------------|--|
| Key_Orders_Resource | Orders_Resource グループのプライマリキー。 |
| Key_Orders | Orders グループのプライマリキー。 |
| FKey_Orders_Resource | Orders グループを Orders_Resource グループにリンクする外部キー。 |

入力ポートを出力キーにリンクする必要があり、リンクしないとマッピングは無効になります。

注: ポートを【出力】パネルの複数の要素にリンクできます。ただし、【ポート】パネルの 1 つのグループから【出力】パネルの複数のグループの要素にフィールドをリンクすることはできません。同じ【ポート】グループのすべてのフィールドが同じ【出力】グループの要素にリンクする必要があります。

出力マッピングの設定

顧客と注文の間にプライマリキーと外部キーのリレーションを作成するには、各グループに顧客番号を含める必要があります。

1. REST 出力トランスフォーメーションの [Orders] グループの [Order_Date] に [Customer_Fkey] ポートを追加します。

[Customer_Fkey] ポートは、顧客番号を受信します。[Orders] グループの顧客番号を受信して外部キーに設定するには、このポートを追加する必要があります。顧客番号は、出力では各注文と一緒に表示されません。

2. [ポート] パネルの Customer_Key を [出力] パネルの [Key_Orders_Resource] にリンクします。リンクを作成するには、[Customer_Key] をクリックして選択し、次に [Key_Orders_Resource] を選択します。[ポート] パネルの [マップ] アイコンをクリックすると、リンクが作成されます。

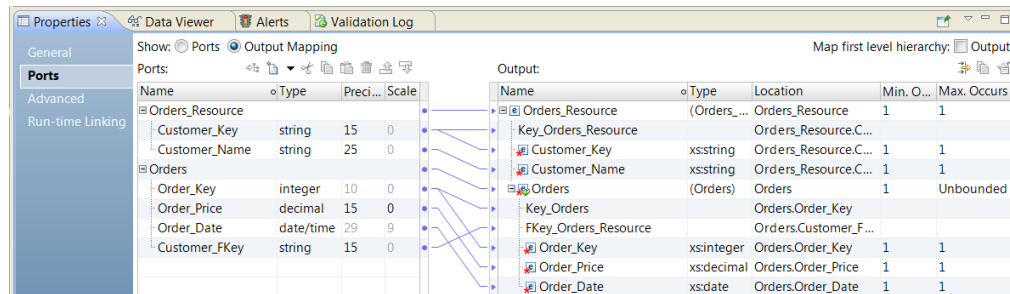
注: ポートをドラッグしてリンクを作成することはできません。

3. [Order_Key] ポートと [Key_Orders] 要素を選択します。[マップ] をクリックすると、リンクが作成されます。

[Order_Key] には、[Key_Orders] 要素と [Order_Key] 要素へのリンクが必要です。

4. [Customer_Fkey] ポートと [Fkey_Orders_Resource] 要素を選択します。[マップ] をクリックすると、リンクが作成されます。

次の図は、出力マッピングとキーへのリンクを示しています。



手順 4. [データビューア] ビューでのマッピングのテスト

REST リソースマッピングは [データビューア] ビューで実行できます。[データビューア] ビューではリソース ID またはフィルタ条件によって出力データをフィルタリングできます。

Orders_Resource グループの要素の値を入力して、データをフィルタリングできます。要素は Customer_Key または Customer_Name です。Orders グループの要素を使用して出力データをフィルタリングすることはできません。Orders グループは子グループです。

リソース識別子で出力データをフィルタリングすることも、フィルタ条件を入力することもできます。リソース識別子はキーであり、キーは Customer_Key 要素です。フィルタ条件は式エディタに入力する式です。キーで検索しない場合、フィルタ条件を使用して出力データをフィルタリングします。

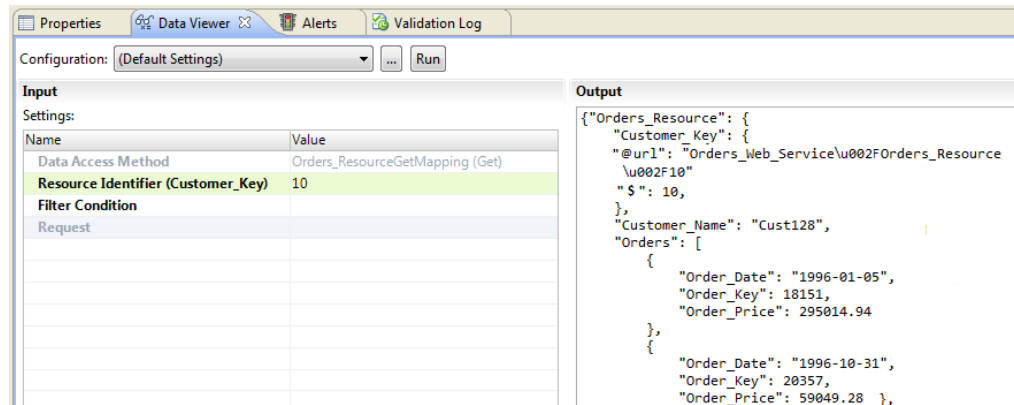
リソース識別子またはフィルタ条件を入力しないと、マッピングによって [データビューア] ビューのすべての行が返されます。

リソース ID による出力のフィルタリング

リソース ID を入力して、キー値で出力データを選択できます。リソース ID は、リソース作成時に選択したキーです。この例の場合、リソース ID は、Customer_Key です。

1. **【データビューア】** ビューで、リソース識別子の **【値】** カラムに有効な顧客番号を入力します。
2. **【実行】** をクリックします。

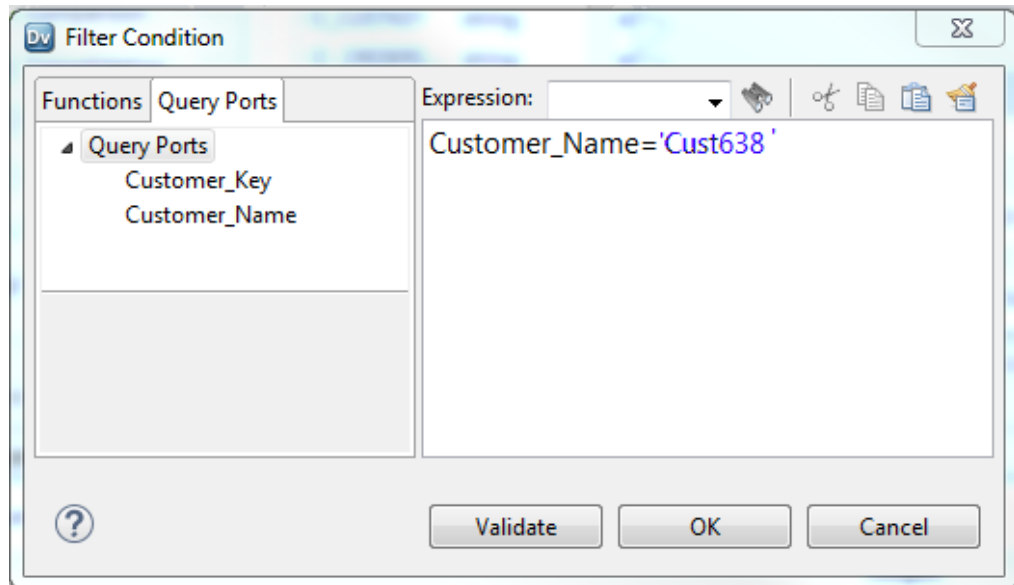
出力には、入力した顧客のデータが含まれます。
次の図は、顧客番号 10 の出力を示しています。



フィルタ条件による出力のフィルタリング

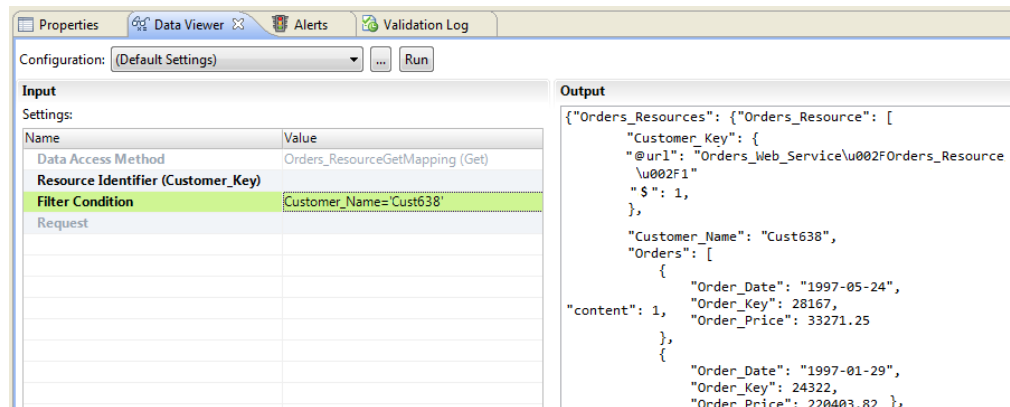
フィルタ条件を入力して、特定の値で出力データを選択できます。

1. **【データビューア】** ビューで、フィルタ条件の **【値】** カラムにある選択矢印をクリックします。
式エディタが表示されます。
2. **【クエリポート】** タブをクリックします。
ポートのリストが表示されます。



3. クエリポートを選択して、フィルタ式を作成します。
検索値は、単一引用符で囲む必要があります。

4. **【検証】** をクリックして式を検証します。
5. 式が有効の場合は、**【OK】** をクリックします。
式は、**【データビューア】** ビューにフィルタ条件として表示されます。
6. **【データビューア】** ビューで、**【実行】** をクリックします。
Cust638 という名前の顧客の注文が **【出力】** パネルに表示されます。
次の図は、**【出力】** パネルの結果を示しています。



手順 5.アプリケーションのデプロイ

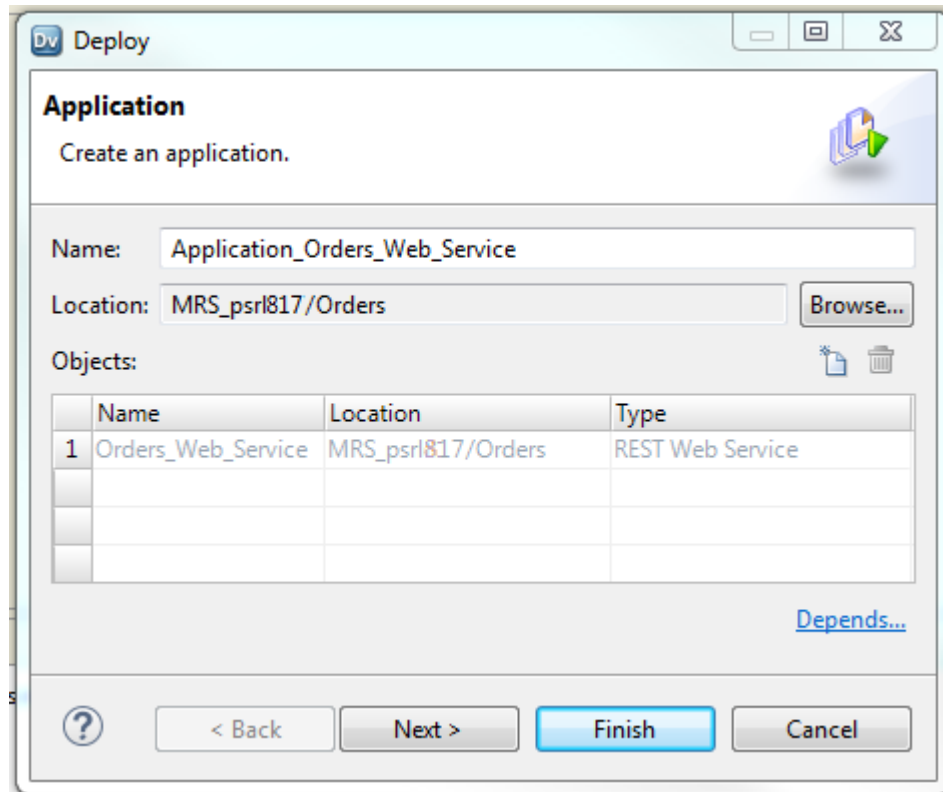
Web サービスをアプリケーションとしてデータ統合サービスにデプロイします。アプリケーションをデプロイして、Web サービスクライアントが Web サービスに接続できるようにする必要があります。

Web サービスをデプロイしたら、Administrator ツールでアプリケーションを表示できます。

アプリケーションのデプロイ

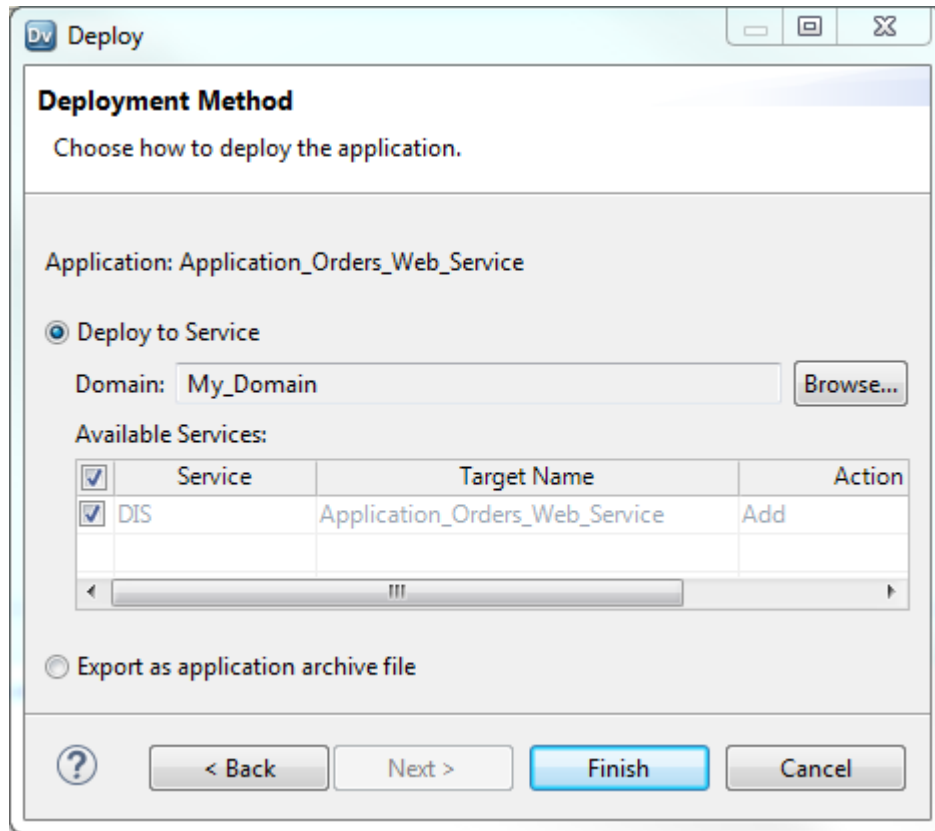
作成したアプリケーションをデータ統合サービスにデプロイします。

1. **【オブジェクトエクスプローラ】** ビューで **[Orders_Web_Service]** を右クリックして、**【デプロイ】** をクリックします。



2. デフォルトの名前と場所を受け入れます。**【次へ】** をクリックします。

3. ドメイン、およびアプリケーションのデプロイ先のデータ統合サービスを選択します。



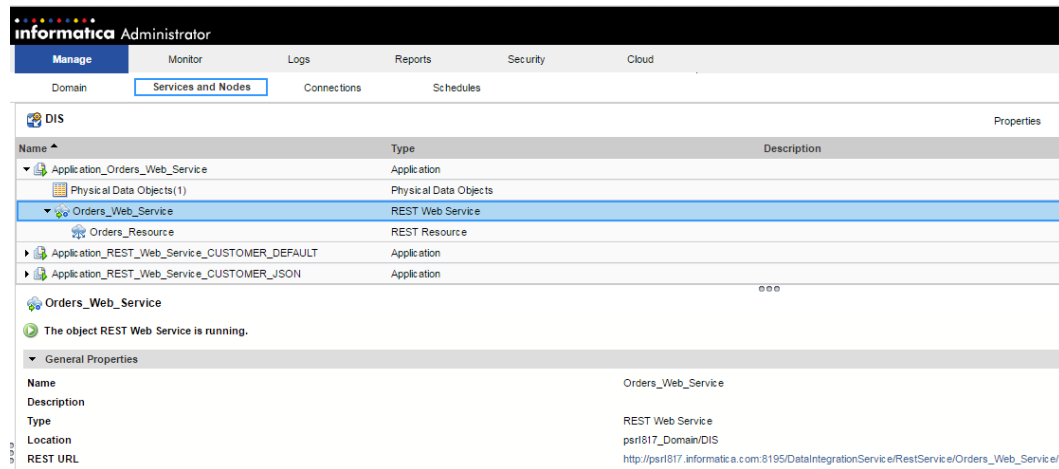
4. **【完了】** をクリックすると、アプリケーションがデプロイされます。

手順 6. ブラウザからの Web サービスのクエリ

ブラウザから REST Web サービスを問い合わせる。

Web サービスをデータ統合サービスにデプロイしたら、Web サービスがアプリケーションとして Administrator ツールに表示されます。アプリケーションの **【全般プロパティ】** パネルで Web サービスにアクセスするための URL を表示できます。

次の図は、Web サービスの【全般プロパティ】パネルの REST URL を示しています。



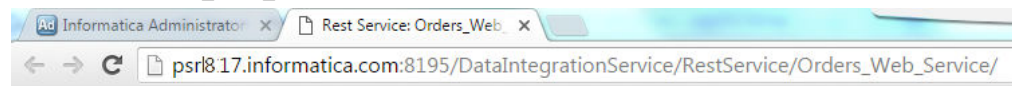
Web サービスのクエリ

URL にリソース ID か、フィルタ式を含めて Web サービスに対しクエリを実行できます。

1. Administrator ツールで Web サービスアプリケーションの【全般プロパティ】パネルの REST URL をクリックします。

データ統合サービスには Web サービスのリソースが表示されます。

次の図は、Orders_Web_Service のリソースです。



Resources:

- [Orders_Resource](#)

2. [Orders_Resource] リンクをクリックします。

Web サービスによって、リソースマッピングからすべての出力データが返されます。

3. 特定の顧客を検索するには、リソース名とリソース ID をブラウザの URL の末尾に追加します。このリンクは、http://psr1817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/10 です。

次の図は、Orders_Resource で顧客番号 10 を検索するクエリの結果です。



```
{
  "Orders_Resource": {
    "Orders_Resource": {
      "Customer_Key": {
        "@url": "http://psrl817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/10",
        "$": 10,
      },
      "Customer_Name": "Cust128",
      "Orders": [
        {
          "Order_Date": "1996-01-05",
          "Order_Key": 18151,
          "Order_Price": 295014.94
        },
        {
          "Order_Date": "1996-10-31",
          "Order_Key": 20357,
          "Order_Price": 59049.28
        },
        {
          "Order_Date": "1997-07-27",
          "Order_Key": 24678,
          "Order_Price": 112857.62
        },
        {
          "Order_Date": "1998-07-03",
          "Order_Key": 25122,
          "Order_Price": 92135.6
        }
      ]
    }
  }
}
```

4. キーの代わりに顧客名で検索するには、リソース ID の代わりに URL にフィルタ条件を追加します。

`http://psrl817.informatica.com:8195/DataIntegrationService/RestService/Orders_Web_Service/Orders_Resource/?filterCondition=Customer_Name='Cust628'`

データオブジェクトから REST Web サービスを作成する方法

モデルリポジトリのデータオブジェクトから REST Web サービスを作成できます。Developer tool を使用して、読み取りトランスフォーメーションと出力トランスフォーメーションを含むデフォルトマッピングを作成します。REST Web サービスリソースを含むデータオブジェクトカラムを選択できます。

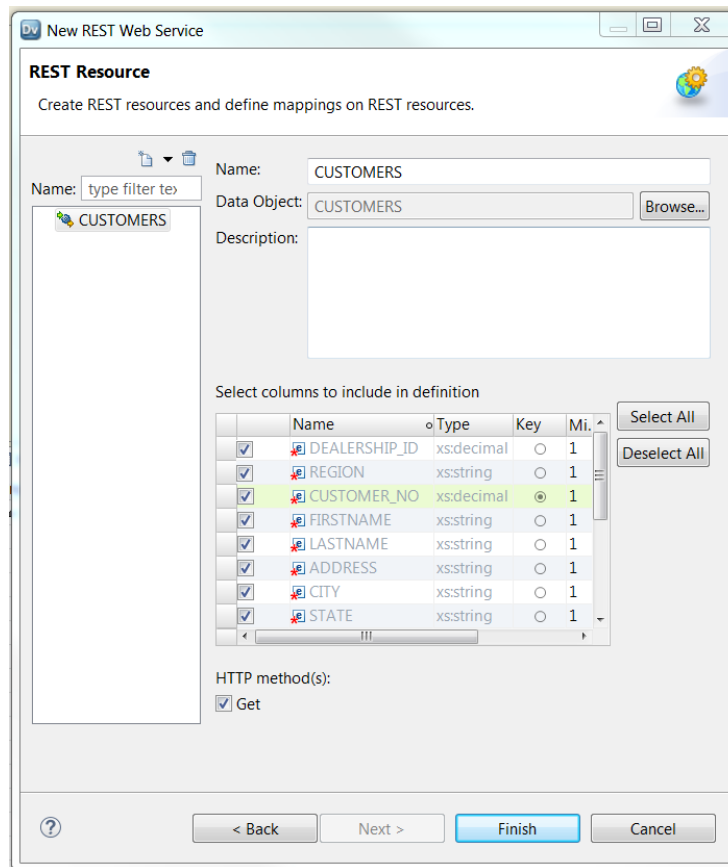
1. **[ファイル]** > **[新規]** > **[データサービス]** をクリックします。
2. **[ウィザードを選択]** ダイアログボックスで、**[REST Web サービス]** を選択します。**[次へ]** をクリックします。
3. **[REST Web サービス]** ダイアログボックスで、Web サービスの名前を入力します。デフォルトの名前は、REST_Web_Service です。**[次へ]** をクリックします。
4. **[REST リソース]** ダイアログボックスで、**[新規]** > **[データオブジェクトから作成]** をクリックします。
5. モデルリポジトリのデータオブジェクトのリストからデータオブジェクトを選択して、**[OK]** をクリックします。

[REST リソース] ダイアログボックスに、選択したデータオブジェクトからの REST リソースが表示されます。

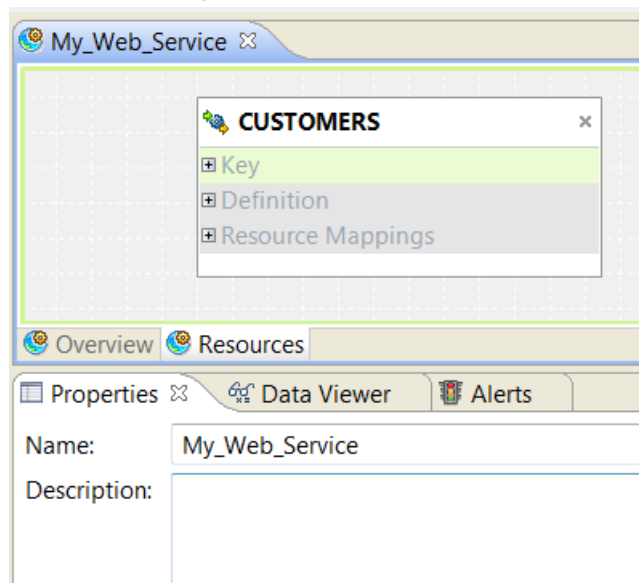
6. リソースに含めるカラムを選択します。リソースのプライマリキーを選択します。

デフォルトでは、すべてのカラムが選択されています。リソースに含めないカラムは、選択を解除できます。デフォルトでは、GET メソッドが選択されています。このオプションは、クリアしないでください。

次の図は、[REST リソース] ダイアログボックスを示しています。

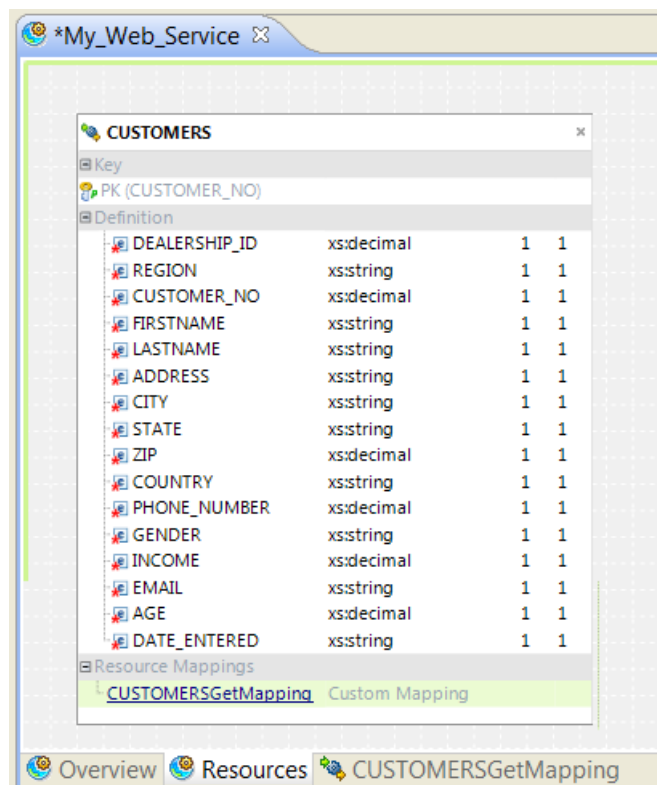


7. **【完了】** をクリックすると、Web サービスが作成されます。
REST Web サービスの **【リソース】** ビューが Developer tool に表示されます。
次の図は、Developer tool の **【リソース】** ビューを示しています。



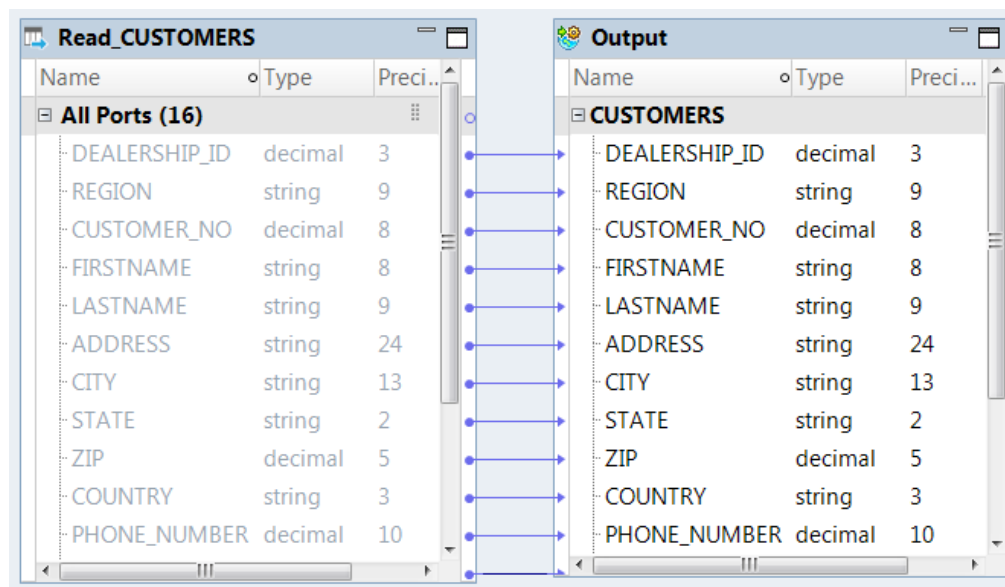
8. **【リソース】** ビューでコンポーネントを展開すると、キー、定義のカラム、およびリソースマッピングのリンクを表示できます。

次の図は、リソースキー、定義のカラム、およびリソースマッピングのリンクを示しています。



9. リソースマッピングのリンクをクリックすると、リソースマッピングが表示されます。

次の図は、デフォルトリソースマッピングを示しています。



10. 必要に応じて、マッピングにトランスフォーメーションを追加します。
11. Web サービスをテストするには、**【データビューア】** ビューに移動します。

12. **【入力 | 設定】** の **【データアクセス値】** フィールドで、テストするリソースマッピングを選択します。
13. 出力をキーでフィルタリングするには、検索するキーを **【リソース識別子】** フィールドに入力します。
14. 出力をその他の要素でフィルタリングするには、フィルタ条件を入力します。**【値】** フィールドをクリックして、式エディタを開きます。

式エディタでは、複数の式を入力できます。

次の図は、**【データビューア】** ビューの **【入力設定】** を示しています。

| Name | Value |
|-----------------------------------|---------------------------|
| Data Access Method | Customer_OrdersGetMapp... |
| Resource Identifier (CUSTOMER_NO) | 3 |
| Filter Condition | |
| Request | |

15. **【実行】** をクリックすると、出力が表示されます。
出力は、Web サービスの設定に基づいて、JSON または XML になります。
16. Web サービスをデータ統合サービスにデプロイします。

データオブジェクトを REST Web サービスとしてデプロイする方法

リレーショナルデータオブジェクトまたはフラットファイルデータオブジェクトを REST Web サービスとしてデプロイできます。

1. オブジェクトエクスプローラで、Web サービスでアクセスするデータオブジェクトを右クリックします。
2. **【デプロイ】** > **【REST Web サービスとしてデプロイします】** をクリックします。

[アプリケーションのデプロイ] ダイアログボックスが表示されます。

| Name | Location | Type |
|------|----------|------|
| | | |
| | | |
| | | |

3. アプリケーションの名前を入力して、[次へ] をクリックします。オブジェクトは追加しないでください。

[デプロイ方法] ダイアログボックスが表示されます。

4. アプリケーションをデプロイするデータ統合サービスを選択します。

次の図は、[デプロイ方法] ダイアログボックスを示しています。

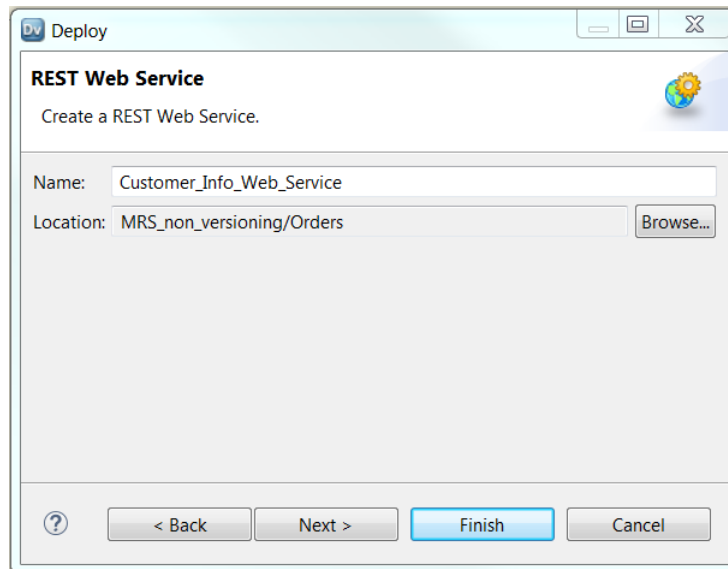
| Service | Target Name | Action |
|---|-------------------|--------|
| <input checked="" type="checkbox"/> E_DIS | Customer_Info_App | Add |

5. [次へ] をクリックします。

[REST Web サービス] ダイアログボックスが表示されます。

6. Web サービスの名前を入力します。

次の図は、**[REST Web サービス]** ダイアログボックスを示しています。



7. **[次へ]** をクリックし、リソースに含めるカラムを選択するか、またはリソースをさらに追加します。**[完了]** をクリックすると、データオブジェクトのすべてのカラムを含むリソースが作成されます。
[次へ] をクリックすると、**[REST リソース]** ダイアログボックスが表示されます。
8. **[REST リソース]** ダイアログボックスで、リソースの名前を入力し、定義に含めるカラムを選択します。
デフォルトでは、すべてのカラムが選択されています。フィールドはクリアしないでください。デプロイされた Web サービスには、デフォルトマッピングが必要です。

次の図は、[REST リソース] ダイアログボックスを示しています。

REST Resource
Create REST resources and define mappings on REST resources.

Name: type filter text
CUSTOMERS

Name: CUSTOMERS
Data Object: CUSTOMERS
Description: Return customer information by customer ID

Select columns to include in definition

| Name | Type | Key | Mi... | M... | Desc |
|---------------|------------|----------------------------------|-------|------|------|
| DEALERSHIP_ID | xs:decimal | <input type="radio"/> | 1 | 1 | |
| REGION | xs:string | <input type="radio"/> | 1 | 1 | |
| CUSTOMER_NO | xs:decimal | <input checked="" type="radio"/> | 1 | 1 | |
| FIRSTNAME | xs:string | <input type="radio"/> | 1 | 1 | |
| LASTNAME | xs:string | <input type="radio"/> | 1 | 1 | |
| ADDRESS | xs:string | <input type="radio"/> | 1 | 1 | |
| CITY | xs:string | <input type="radio"/> | 1 | 1 | |
| STATE | xs:string | <input type="radio"/> | 1 | 1 | |
| ZIP | xs:decimal | <input type="radio"/> | 1 | 1 | |
| COUNTRY | xs:string | <input type="radio"/> | 1 | 1 | |
| PHONE_NUMBER | xs:decimal | <input type="radio"/> | 1 | 1 | |
| GENDER | xs:string | <input type="radio"/> | 1 | 1 | |
| INCOME | xs:decimal | <input type="radio"/> | 1 | 1 | |

HTTP method(s):
☒ Get

< Back Next > Finish Cancel

9. **【完了】** をクリックすると、Web サービスがデプロイされます。
リソースを追加したり、リソース定義を変更したりしないでください。
10. Web サービスにリソースを追加するには、**【新規】** をクリックします。
データオブジェクトからリソースを作成することも、手動でリソースのカラムに入力することもできます。

第 12 章

REST Web サービスコンシューマ トランスフォーメーション

この章では、以下の項目について説明します。

- [REST Web サービスコンシューマトランスフォーメーションの概要, 144 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションの設定, 146 ページ](#)
- [HTTP メソッド, 148 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションのポート, 151 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションの入力マッピング, 153 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションの出力マッピング, 156 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションの詳細プロパティ, 158 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションの作成, 159 ページ](#)
- [配列を含む JSON 応答メッセージの解析, 160 ページ](#)

REST Web サービスコンシューマトランスフォーメーションの概要

REST Web サービスコンシューマトランスフォーメーションは、データアクセスまたはデータ変換する Web サービスクライアントとして REST Web サービスに接続するアクティブなトランスフォーメーションです。REST Web サービスコンシューマトランスフォーメーションを使用して、REST Web サービスに接続します。REST Web サービスコンシューマトランスフォーメーションは、リクエストを REST Web サービスに送ることと、REST Web サービスから応答を受け取ることができます。

REST Web サービスコンシューマトランスフォーメーションは、トランスフォーメーションまたは HTTP 接続で定義する URL を通じて、Web サービスに接続します。HTTPS 接続を使用することもできます。REST Web サービスコンシューマトランスフォーメーションでは、TLS 1.2、TLS 1.1、TLS 1.0 を使用できます。

REST Web サービスには、Web サービスがサポートするアクションごとに HTTP メソッドが 1 つあります。データ統合サービスが REST Web サービスに接続されている場合、データの Get、Post、Put、Delete を実行するためのリクエストが送信できます。このリクエストの対象は、個別のリソースでもリソースの集合でもかまいません。データ統合サービスがリクエストメッセージを送った後に、Web サービスから応答メッセージを受け取ります。

リクエストメッセージと応答メッセージには、階層を形成できる要素を使った XML または JSON のデータが入っています。リクエストメッセージまたは応答メッセージに複数回出現する要素が入っている場合、要素のグ

ループが XML または JSON の階層内にレベルを形成します。あるレベルが別のレベルの入れ子になっている場合、グループは関連しています。

REST Web サービスコンシューマトランスフォーメーションでは、メソッド入力とメソッド出力でリクエストメッセージと応答メッセージの構造を定義します。メソッド入力とメソッド出力にはマッピングが含まれ、メッセージ要素を入力ポートと出力ポートにマップする方法を定義しています。

REST Web サービスコンシューマトランスフォーメーションはプロキシサーバーをサポートします。REST Web サービスコンシューマトランスフォーメーションを使って Microsoft SharePoint アプリケーションに接続することもできます。

例

オンラインストアは、製品データベースに対しリソースを定義します。データベースは各製品を部品番号で識別します。

Web サービスクライアントは REST Web サービスを通じて製品の詳細にアクセスします。Web サービスは次の URL を使用します。

<http://www.HypoStores.com/products/ProductDetails>

具体的な製品についての詳細（説明や単価など）を取得し、詳細をマッピング内のダウンストリームのトランスフォーメーションに渡す必要があります。製品についての詳細を取得して別のトランスフォーメーションに渡すための REST Web サービスコンシューマトランスフォーメーションを作成します。

次のテーブルに、ユーザーが設定するトランスフォーメーションの詳細を示します。

| トランスフォーメーションの詳細 | 値 |
|-----------------|---|
| HTTP メソッド | 取得 |
| ベース URL | http://www.HypoStores.com/products/ProductDetails |
| 入力引数ポート | Part_No |
| 出力ポート | Description、Unit_Price |
| メソッド出力 | <応答メッセージの構造。> |

メソッド出力は、出力マッピングを含み、応答メッセージ内の要素を出力ポートにマップする方法を定義しています。

データ統合サービスがリクエストを Web サービスに送るときに、引数ポート内の値をベース URL に付加します。例えば、部品 0716 についての詳細を取得するために、データ統合サービスは次の URL を使用します。

http://www.HypoStores.com/products/ProductDetails?Part_No=0716

データ統合サービスは応答を受け取ると、応答メッセージ内の製品の説明と単価を出力ポートのデータに変換します。

Part_No をパラメータとして渡して、マッピングを実行している最中に値を置き換えることもできます。

REST Web サービスコンシューマトランスフォーメーションの処理

REST Web サービスコンシューマトランスフォーメーションは、メソッド入力と入力ポート内のデータに基づいてリクエストメッセージを作成します。応答メッセージ内の要素をメソッド出力に基づいて出力ポートのデータに変換します。

REST Web サービスコンシューマトランスフォーメーションの入力ポートには、マッピング内のアップストリームのトランスフォーメーションからのリレーショナルデータがあります。データ統合サービスは、メソッド入力を使用して、データを入力ポートからリクエストメッセージ内の要素に変換します。

Web サービスに接続するときに、データ統合サービスは、トランスフォーメーションプロパティまたは HTTP 接続で設定したベース URL を読み込みます。URL ポートまたは引数ポートからの値をベース URL に付加することによって、Get、Post、Put、Delete を実行する必要があるリソースを識別します。

データ統合サービスが応答を受け取ると、応答メッセージ内のデータをトランスフォーメーションの出力ポートに渡します。データ統合サービスは、メソッド出力を設定する方法に基づいてデータを渡します。出力ポートにはリレーショナルデータが入っています。データ統合サービスは、出力ポート内のデータをマッピング内のダウストリームのトランスフォーメーションに、またはターゲットに送ります。

REST Web サービスコンシューマトランスフォーメーションの設定

REST Web サービスコンシューマトランスフォーメーションを作成する場合は、HTTP メソッドを選択し、メソッド入力とメソッド出力を定義します。Get メソッドを選択する場合は、メソッド入力を定義しません。

HTTP リクエストメッセージ内の入力要素は、入力ポートにマップします。HTTP 応答メッセージ内の出力要素は、出力ポートにマップします。Developer ツールは、最初のレベルの要素にポートを作成します。

トランスフォーメーションを設定するときは、以下のタスクをすべて行ってください。

1. HTTP メソッドを選択します。
2. リクエストメッセージと応答メッセージのヘッダと本文の要素を代表するポートを設定します。
3. 入力マッピングを設定します。
4. 出力マッピングを設定します。
5. 詳細プロパティ（Web サービスのベース URL や接続など）を設定します。

REST Web サービスに認証が必要な場合、HTTP 接続オブジェクトを作成します。

メッセージの設定

データ統合サービスがリクエストメッセージを生成し、メソッド入力とメソッド出力に基づいて、さらに REST Web サービスコンシューマトランスフォーメーションで設定するポートに基づいて、応答メッセージを解釈します。

入力ポートは、リクエストメッセージのさまざまな部分を表します。取得または変更するリソースを特定する入力ポートを追加できます。リクエストメッセージ内の HTTP ヘッダ、クッキー情報、および要素を表す入力ポートを追加することもできます。

出力ポートは、ダウストリームのトランスフォーメーションやマッピング内のターゲットに送信する応答メッセージの要素を表します。応答メッセージ内の HTTP ヘッダ、クッキー情報、応答コード、および要素を表す出力ポートを追加できます。

リソースの識別

HTTP リクエストでリソースを識別できるように、データ統合サービスがベース URL に特定の入力ポートの値を付加します。ベース URL は、HTTP 接続またはトランスフォーメーションプロパティで定義します。URL または引数ポートを使用して特定のリソースを識別します。

Web サービスが一意的な文字列を使ってリソースを識別するときに、URL ポートを使用します。

例えば、HypoStores という REST Web サービスは、次の URL を通じて部品番号で部品を識別します。

```
http://www.HypoStores.com/products/ProductDetails/<Part_No>
```

部品を識別するには、次のトランスフォーメーションの詳細を定義します。

1. ベース URL を、次の URL に設定します。

```
http://www.HypoStores.com/products/ProductDetails
```

2. URL ポートを定義し、その URL ポートを介して部品番号をトランスフォーメーションに渡します。

マッピングが部品番号 500 を URL ポートに渡す場合、データ統合サービスはリクエストメッセージで次の URL を使用します。

```
http://www.HypoStores.com/products/ProductDetails/500
```

Web サービスがリソースの場所を引数で識別する場合に、引数ポートを使用します。

例えば、部品番号を HypoStores という REST Web サービスに「Part_No」という引数で渡す必要があるとします。

部品を識別するには、次のトランスフォーメーションの詳細を定義します。

1. ベース URL を、次の URL に設定します。

```
http://www.HypoStores.com/products/ProductDetails
```

2. 引数名「Part_No」を使って引数ポートを作成し、その引数ポートから部品番号をトランスフォーメーションに渡します。

マッピングが部品番号 600 を引数ポートに渡す場合、データ統合サービスはリクエストメッセージで次の URL を使用します。

```
http://www.HypoStores.com/products/ProductDetails?Part_No=600
```

複数の引数を定義するには複数の引数ポートを作成します。データ統合サービスでは、アンパーサンド文字 (&) で引数と引数を分割します。

例えば、従業員の詳細を REST Web サービスから取得し、従業員の名と姓をそれぞれ「First_Name」と「Last_Name」という引数を使って渡します。「First_Name」と「Last_Name」という引数名の引数ポートを作成します。マッピングが「John Smith」という名前をトランスフォーメーションに渡す場合、データ統合サービスはリクエストメッセージで次のように URL を使用します。

```
http://www.HypoStores.com/employees/EmpDetails?First_Name=John&Last_Name=Smith
```

URL または引数ポートを指定しない場合、データ統合サービスは、HTTP 接続またはトランスフォーメーションプロパティのベース URL を使用してリソースを識別します。HTTP 接続内のベース URL は、トランスフォーメーション内のベース URL より優先されます。

HTTP メソッド

REST Web サービスコンシューマトランスフォーメーションを作成するときは、データ統合サービスがリクエストメッセージで使用する HTTP メソッドを選択します。トランスフォーメーションを作成した後は、HTTP メソッドを変更することはできません。

以下の HTTP メソッドのうちいずれかを使用するようにトランスフォーメーションを設定します。

Get

Web サービスからリソースを個別にあるいはまとめて取得します。例えば、複数製品を表形式で取得することも、1 製品についての情報を取得することもできます。

Post

Web サービスにデータを送信します。Post メソッドを使用すると、リソースを個別に、またはまとめて作成できます。例えば、新しい店の取引の詳細を追加できます。

Put

リソースを個別に、あるいはまとめて置き換えます。データが存在しない場合、Put メソッドでデータを送信します。例えば、発送先となる顧客の住所を更新できます。

Delete

リソースを個別に、あるいはまとめて削除します。例えば、すでに退職している社員のレコードを削除できます。

HTTP Get メソッド

データ統合サービスは、HTTP Get メソッドを使用して REST Web サービスからデータを取得します。Get メソッドを使用してリソースを個別に、またはまとめて取得します。

Get メソッドを使用するように REST Web サービスコンシューマトランスフォーメーションを設定する場合、入力ポート、メソッド出力、および出力ポートを設定します。メソッド入力には設定しません。

例

HypoStores という製品のデータベースから部品番号 500 に関する説明と価格を取得する必要があるとします。この Web サービスでは、次の URL を使用して部品を特定しています。

`http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>`

次の URL を入力します。

`http://www.HypoStores.com/products/ProductDetails`

定義する可能性のある入力ポートを下表に示します。

| ポートタイプ | 引数名 | 入力値 |
|--------|---------|-----|
| 引数 | Part_No | 500 |

定義する可能性のある出力ポートを下表に示します。

| ポートタイプ | ポート名 | 戻り値 |
|--------|-----------|--|
| 出力 | Part_Desc | …<desc>ACME ボールペン、12-pk、黒、0.7 mm</desc>… |
| 出力 | Price_USD | …<price>9.89</price>… |

HTTP Post メソッド

データ統合サービスは、HTTP Post メソッドを使用して REST Web サービスにデータを送信します。Web サービスは、Post メソッドが実行する実際の機能を決定します。リソースを個別またはまとめて作成する際に、Post メソッドを使用する場合があります。

Post メソッドを使用するように REST Web サービスコンシューマトランスフォーメーションを設定する場合、入力ポート、メソッド入力、メソッド出力、および出力ポートを設定します。

例

HypoStores という製品のデータベースに新しい部品 501 を掲載する必要があるとします。Web サービスは部品 501 に、次の URL を使用しています。

<http://www.HypoStores.com/products/ProductDetails/501>

次の URL を入力します。

<http://www.HypoStores.com/products/ProductDetails>

定義する可能性のある入力ポートを下表に示します。

| ポートタイプ | ポート名 | 入力値 |
|--------|-------------|---------------------------|
| URL | URL_Part_No | 501 |
| 入力 | Part_Desc | ACME ボールペン、12-pk、黒、0.5 mm |
| 入力 | Price_USD | 9.89 |

定義する可能性のある出力ポートを下表に示します。

| ポートタイプ | ポート名 | 戻り値 |
|--------|------|-----------------|
| 出力 | 応答 | <Web サービスが戻す応答> |

HTTP Put メソッド

データ統合サービスは、HTTP Put メソッドを使用して REST Web サービスによるデータの更新を行います。Put メソッドを使用してリソースを個別に、またはまとめて更新します。データが存在しない場合、データ統合サービスがリソースを個別にあるいはまとめて作成します。

Put メソッドを使用するように REST Web サービスコンシューマトランスフォーメーションを設定する場合、入力ポート、メソッド入力、メソッド出力、および出力ポートを設定します。

例

HypoStores という製品のデータベースで、部品 501 の単価を更新する必要があるとします。Web サービスは部品 501 に、次の URL を使用しています。

`http://www.HypoStores.com/products/ProductDetails/501`

次の URL を入力します。

`http://www.HypoStores.com/products/ProductDetails`

定義する可能性のある入力ポートを下表に示します。

| ポートタイプ | ポート名 | 入力値 |
|--------|-------------|------|
| URL | URL_Part_No | 501 |
| 入力 | Price_USD | 9.99 |

定義する可能性のある出力ポートを下表に示します。

| ポートタイプ | ポート名 | 戻り値 |
|--------|------|-----------------|
| 出力 | 応答 | <Web サービスが戻す応答> |

HTTP Delete メソッド

データ統合サービスは、HTTP Delete メソッドを使用して REST Web サービスによるデータ削除を行います。Delete メソッドを使用してリソースを個別に、またはまとめて削除します。

Delete メソッドを使用するように REST Web サービスコンシューマトランスフォーメーションを設定する場合、入力ポート、メソッド入力、メソッド出力、および出力ポートを設定します。

例

HypoStores という製品のデータベースから部品番号 502 を削除する必要があるとします。この Web サービスでは、次の URL を使用して部品を特定しています。

`http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>`

次の URL を入力します。

`http://www.HypoStores.com/products/ProductDetails`

定義する可能性のある入力ポートを下表に示します。

| ポートタイプ | 引数名 | 入力値 |
|--------|---------|-----|
| 引数 | Part_No | 502 |

定義する可能性のある出力ポートを下表に示します。

| ポートタイプ | ポート名 | 戻り値 |
|--------|------|-----------------|
| 出力 | 応答 | <Web サービスが戻す応答> |

REST Web サービスコンシューマトランスフォーマーのポート

REST Web サービスコンシューマトランスフォーマーに、複数の入力ポートと複数の出力ポートを設定することができます。XML または JSON の階層の構造に基づいてポートをグループに分けて作成します。

トランスフォーマーのポートを表示するときに、XML または JSON 階層を確認する必要がない場合はポートを表示します。ポートを表示しているときは、グループの定義、ポートの定義、およびメソッド入力/出力から入力/出力ポートへの要素のマッピングを行うことができます。

REST Web サービスコンシューマトランスフォーマーには、複数の入力グループと複数の出力グループを設定できます。ポートを作成するときは、グループを作成してそのグループにポートを追加します。ポートは、XML または JSON 内の入力階層または出力階層の構造に基づいたグループ階層で定義します。キーを追加して、子グループと親グループを関連付けます。

プライマリキーは、最下位のグループを除く階層内のすべてのグループに必要です。外部キーは、ルートグループを除く階層内のすべてのグループに必要です。

このトランスフォーマーには、RequestInput という名前のルート入力グループがあります。プライマリキーをルートの入力グループに追加する必要があります。キーは、string 型、bigint 型、integer 型のいずれかである必要があります。ルートの入力グループに任意のポートをパススルーポートとして設定できます。

要素をポートにマップするには、**[場所]** カラム内のフィールドをクリックし、**[場所の選択]** ダイアログボックスで階層を展開します。その後で階層から要素を選択します。

入力ポート

入力ポートは、アップストリームのトランスフォーマーからのデータ、または Web サービスに渡す必要のあるソースを表します。複数の入力ポートを設定できます。各入力ポートは、リクエストメッセージ内の要素に割り付けられています。

入力ポートを追加するには、入力グループを選択してから **[新規]** ボタンの横の矢印をクリックし、**[フィールド]** を選択します。

出力ポート

出力ポートは、ダウンストリームのトランスフォーマーまたはターゲットに渡す予定の応答メッセージ内の要素を表します。複数の出力ポートを設定できます。各出力ポートが応答メッセージ内の要素にマップされます。

出力ポートを追加するには、出力グループを選択し、**[新規]** ボタンの横の矢印をクリックして、**[フィールド]** を選択します。

パススルーポート

パススルーポートは、データを変更せずにトランスフォーマーを介してデータを渡します。ルートの入力グループに任意のポートをパススルーポートとして設定できます。

パススルーポートを追加するには、ポートをルートの入力グループに追加します。ポートを右クリックして、**[マップ]** を選択します。

引数ポート

リソースの URL が引数を取るときに引数ポートでリソースを識別することができます。引数ポートは入力グループのルートに追加します。

引数ポートにはポート名と引数名があります。ポート名に許可されていない文字が引数名に含まれている場合は、ポート名と異なる引数名を入力してください。例えば、引数「Cust-ID」を Web サービスに渡そうとしても、ポート名の中のダッシュ (-) をデータ統合サービスが許可しません。この場合は「Cust-ID」を引数名として入力し、「CustID」をポート名として入力してください。

データ統合サービスは、各引数ポートの引数名と値を、「名前=値」の形でペアにしてベース URL に付加します。引数ポートは複数設定することができます。その場合はデータ統合サービスが、アンパーサンド (&) 文字を使ってリクエスト内の複数の引数を分割します。

以下に例を示します。

```
http://www.HypoStores.com/customers/CustDetails?Last_Name=Jones&First_Name=Mary
```

トランスフォーメーション内で引数ポートと URL ポートを定義する場合は、データ統合サービスは URL ポートの値をベース URL に付加し、その後に引数名と値を続けます。

引数ポートを追加するには、ルートの入力グループを右クリックし、**[新規]** > **[引数ポート]** を選択します。引数名とポート名を入力します。

URL ポート

URL ポートでは、静的 URL を通じてリソースを識別できます。リソースを識別するために、データ統合サービスが URL ポートの値をベース URL に付加します。

以下に例を示します。

```
http://www.HypoStores.com/products/ProductDetails/<URL_port_value>
```

ルートの入力グループに URL ポートを追加します。

複数の URL ポートを設定することができます。データ統合サービスは、各 URL ポート内の値をスラッシュ文字 (/) で分割します。トランスフォーメーションの URL ポートと引数ポートを定義する場合、データ統合サービスはベース URL の後で引数名と値の前に URL ポート値を付加します

URL ポートを追加するには、ルートの入力グループを右クリックして、**[新規]** > **[URL ポート]** を選択します。

HTTP ヘッダポート

HTTP ヘッダポートは、要求メッセージの中の HTTP のヘッダを表します。複数の HTTP ヘッダポートを設定できます。

リクエストでヘッダ情報を Web サービスに渡すには、ポートをルートの入力グループに追加します。ルートの入力グループに 1 つの HTTP ヘッダポートを設定できます。HTTP ヘッダをルート入力グループに追加する場合、それをパススルーポートとして設定できます。

HTTP ヘッダポートにはポート名と HTTP ヘッダ名があります。ポート名には許可されていない文字が HTTP ヘッダ名に含まれている場合は、ポート名と異なる HTTP ヘッダ名を入力してください。例えば、ヘッダ名「Content-Type」を Web サービスに渡そうとしても、ポート名の中のダッシュ (-) をデータ統合サービスが許可しません。「Content-Type」を HTTP ヘッダ名として入力し、「ContentType」をポート名に入力します。

HTTP ヘッダポートを追加するには、ルートの入力グループを右クリックし、**[新規]** > **[HTTP ヘッダ]** を設定します。ヘッダ名とポート名を入力します。

クッキーポート

クッキー認証を使用するよう REST Web サービスコンシューマトランスフォーメーションを設定することができます。リモートの Web サーバーが、クッキーに基づいて、Web サービスコンシューマのユーザーを追跡します。マッピングが Web サービスを繰り返し呼び出す場合のパフォーマンスが向上します。

リクエストでクッキー情報を Web サービスに渡すには、ポートをルートの入力グループに追加します。ルートの入力グループに 1 つのクッキーポートを設定できます。クッキーポートをルートの入力グループに追加する場合、それをパススルーポートとして設定できます。

クッキー情報を応答から抽出するには、クッキーポートを出力グループに追加します。各出力グループに 1 つのクッキーポートを設定できます。

クッキーポートを Web サービス要求メッセージに射影する場合、Web サービスプロバイダは応答メッセージでクッキー値を返します。クッキー値は、マッピング内のダウンストリームにある別のトランスフォーメーションに渡すことも、ファイル内に保存することもできます。クッキー値をファイルに保存する場合、そのクッキー値を REST Web サービスコンシューマトランスフォーメーションへの入力に設定できます。

クッキーポートを追加するには、ルートの入力グループを右クリックして、**[新規]** > **[その他のポート]** を選択します。その次に、**[クッキーポート]** **[OK]** をクリックします。

出力 XML ポート

出力 XML ポートは Web サービスからの応答を表します。出力 XML ポートは文字列ポートです。

出力 XML ポートを出力グループに追加します。出力ポートごとに出力 XML ポートを 1 つ設定することができます。

ルートの入力グループを右クリックし、**[新規]** > **[その他のポート]** を選択します。その次に **[出力 XML]** を選択し、**[OK]** をクリックします。

応答コードポート

応答コードのポートは、Web サービスからの HTTP 応答コードを代表しています。応答コードポートは整数のポートです。

応答コードポートを出力グループに追加します。出力グループごとに応答コードポートを 1 つ設定することができます。

応答コードのポートを追加するには、出力グループを選択して、ルートの入力グループを右クリックして、**[新規]** > **[その他のポート]** を選択します。それから **[応答コード]** を選択して **[OK]** をクリックします。

REST Web サービスコンシューマトランスフォーメーションの入力マッピング

トランスフォーメーションのポートを確認するときは、入力マッピングを表示してメソッド入力階層を表示します。入力マッピングを表示しているときは、入力グループの定義、入力ポートの定義、およびメソッド入力要素への入力ポートのマッピングを行うことができます。

入力マッピングは以下の領域で構成されています。

ポート

[ポート] 領域では、トランスフォーメーションの入力グループと入力ポートを作成します。

メソッド入力

【メソッド入力】領域には、REST Web サービスコンシューマトランスフォーメーションから Web サービスに送られるリクエストメッセージ内の要素が表示されます。スキーマオブジェクトを使用してトランスフォーメーションを作成する場合、スキーマオブジェクトがメソッド入力の階層を定義します。

入力ポートを作成したら、その入力ポートを【ポート】領域から【メソッド入力】領域の要素にマップします。入力ポートをメソッド入力内の要素にマップすると、【メソッド入力】領域の【場所】カラムにポートの場所が表示されます。

入力階層の第 1 レベルをマップするよう選択すると、Developer ツールによってメソッド入力の第 1 レベルの要素が入力ポートにマップされます。また、Developer ツールはマッピングを実行するためのポートも作成します。階層の第 1 レベルに複数回出現する親要素が含まれ、その子要素が 1 つまたは複数回出現する場合、Developer ツールは階層の第 1 レベルをマップしません。

入力ポートをメソッド入力内の要素に結ぶ線を表示するように選択できます。

入力ポートを要素にマップするためのルールとガイドライン

入力ポートをメソッド入力階層内の要素にマップする場合には、以下のルールを確認します。

- 階層内の 1 個の要素に対し 1 個の入力ポートをマップできます。同じポートを、階層内の任意の数のキーにマップすることができます。
- 入力ポートおよび要素にデータ型の互換性があることが必要です。
- 1 つの入力グループ内のポートを、メソッド入力内の複数の階層レベルにマップできます。
- メソッド入力のキーに入力ポートをマップする必要があります。キーにマップするポートのデータ型は、string、integer、または bigint であることが必要です。メソッド入力内の、リクエストメッセージに含める階層レベルより上位にある全レベルのキーに、データをマップします。マップするレベルと、その上位にある全レベルの外部キーを含めます。

注: メソッド入力階層の最下位レベルだけをマップする場合は、入力ポートをキーにマップする必要はありません。

- RequestInput のルート要素を、メソッド入力定義の Rest_Consumer_input グループの子要素にマップする必要があります。
- string 型、bigint 型、または integer 型の複数の入力ポートを【メソッド入力】領域のキーにマップして、複合キーを作成できます。複合キーの【場所】フィールドをクリックして、入力ポートの順序を変更したり、いずれかのポートを削除したりすることができます。
- Web サービスが JSON 文書を生成する場合、xmlRoot が応答階層の中の最初のノードであることを確認します。xmlRoot が Web サービスの JSON 応答を使った最初のノードではない場合、NULL 値が表示される可能性があります。

メソッド入力への入力ポートのマッピング

メソッド入力要素が表示されているときに、入力グループと入力ポートを定義でき、さらに入力ポートをメソッド入力要素にマップすることができます。

1. REST Web サービスコンシューマトランスフォーメーションを開きます。
2. 【ポート】ビューで、入力マッピングを表示します。
3. ルート入力グループのプライマリーキーを定義します。

4. 次のいずれかの方法を使用して、入力グループまたは入力ポートを【ポート】領域に追加します。

| 方法 | 説明 |
|---------------------------|--|
| 要素をドラッグ。 | 【メソッド入力】領域からグループまたは子要素を【ポート】領域の空のカラムにドラッグします。【ポート】領域にグループをドラッグする場合、Developer ツールはポートを持たないグループを追加します。 |
| グループまたはポートを手動で追加。 | グループを追加するには、【新規】ボタンの隣の矢印をクリックし、【グループ】をクリックします。ポートを追加するには、【新規】ボタンの隣の矢印をクリックし、【フィールド】をクリックします。 |
| 別のトランスフォーメーションからポートをドラッグ。 | エディタ内で、ポートを別のトランスフォーメーションから REST Web サービスコンシューマトランスフォーメーションにドラッグします。 |
| ポートをコピー。 | 別のトランスフォーメーションからポートを選択し、【ポート】領域へコピーします。ポートをコピーするときは、キーボードショートカットを使用するか、Developer ツールの【コピー】ボタンと【貼り付け】ボタンを使用します。 |
| 【階層の第 1 レベルのマッピング】を選択。 | Developer ツールが、メソッド入力の第 1 レベルの要素を入力ポートと入力グループにマッピングします。また、Developer ツールはマッピング実行のための入力ポートと入力グループも作成します。 |

5. ポートを手動で作成するか、別のトランスフォーメーションからポートをコピーするときは、【メソッド入力】領域の【場所】カラムをクリックし、リストからポートを選択します。
6. 入力ポートを複合キーにマッピングするには、以下のいずれかの方法を使用します。

| 方法 | 説明 |
|-------------------------------|--|
| 入力ポートをドラッグする。 | 2 つ以上の入力ポートを選択してから、メソッド入力階層内のキーにドラッグします。 |
| 【場所の選択】ダイアログボックスから入力ポートを選択する。 | メソッド入力階層でキーの【場所】カラムをクリックし、入力ポートを選択します。 |

7. ポートの場所をクリアするには、以下のいずれかの方法を使用します。

| 方法 | 説明 |
|----------------|--|
| 【クリア】をクリックする。 | 【メソッド入力】領域で要素を 1 つか複数選択し、【クリア】をクリックします。 |
| ポートと要素を結ぶ線を削除。 | メソッド入力領域で入力ポートと要素を結ぶ線を 1 つか複数選択してから、キーボードの Delete キーを押します。 |

REST Web サービスコンシューマトランスフォーメーションの出力マッピング

トランスフォーメーションのポートを確認するときは、出力マッピングを表示してメソッド出力階層を確認します。出力マッピングを表示している場合は、出力グループの定義、出力ポートの定義、出力ポートへのメソッド出力要素のマッピングを行うことができます。

出力マッピングは以下の領域で構成されています。

メソッド出力

【操作出力】 領域には、Web サービスから REST Web サービスコンシューマトランスフォーメーションに返される応答メッセージ内の要素が表示されます。スキーマオブジェクトを使用してトランスフォーメーションを作成する場合、スキーマオブジェクトがメソッド出力の階層を定義します。

ポート

【ポート】 領域では、トランスフォーメーションの出力グループと出力ポートを作成します。

出力ポートを作成したら、要素を **【メソッド出力】** 領域から **【ポート】** 領域のポートにマップします。要素をメソッド出力から出力ポートにマップすると、要素の場所が **【ポート】** 領域の **【場所】** カラムに表示されます。

出力階層の第 1 レベルをマップするよう選択すると、Developer ツールによってメソッド出力の第 1 レベルの要素が出力ポートにマップされます。また、Developer ツールはマッピングを実行するためのポートも作成します。階層の第 1 レベルに複数回出現する親要素が含まれ、その子要素が 1 つまたは複数回出現する場合、Developer ツールは階層の第 1 レベルをマップしません。

出力ポートを階層構造で表示するように選択することもできます。それぞれの子グループが親グループの下に表示されます。メソッド出力内の要素を出力ポートと結ぶ線を表示するように選択できます。

関連付けられたスキーマオブジェクトがリポジトリから削除された場合、Developer ツールはメソッド要素の場所を出力マッピングに保持します。出力マッピングを表示すると、**【ポート】** 領域で出力ポートの **【場所】** カラムにメソッド要素の場所が表示されます。別のスキーマをトランスフォーメーションに関連付けると、Developer ツールがそれぞれの場所が有効であるかどうかを調べます。場所が有効でなくなると、Developer ツールは出力マッピングの **【ポート】** 領域にあるメソッド要素の場所をクリアします。

要素を出力ポートにマップするためのルールとガイドライン

メソッド出力階層内の要素を出力ポートにマップする際は、以下のルールを確認してください。

- メソッド出力要素および出力ポートにはデータ型の互換性が必要です。
- 1 つの要素を同じグループ内の複数の出力ポートにマップすることはできません。
- パススルーポートを除く各出力ポートに、有効な場所を指定する必要があります。
- 複数出現の子要素を空の出力ポートにドラッグした場合、グループを他の出力グループに関連付ける必要があります。グループを選択すると、Developer ツールがグループ間に関連付けるキーを作成します。
- 複数回出現する要素を親要素が含まれているグループにドラッグする場合、含める子要素の出現回数を設定することができます。または、親グループを、トランスフォーメーション出力内の複数出現子グループで置き替えることもできます。
- Web サービスが JSON 文書を生成する場合、xmlRoot が応答階層の中の最初のノードであることを確認します。xmlRoot が Web サービスの JSON 応答を使った最初のノードではない場合、NULL 値が出力ポートに現れる可能性があります。

ビューのカスタマイズのオプション

メソッド出力階層を、クッキーポート、パススルーポート、およびキーが【メソッド出力】領域内に表示されるように変更することができます。要素の順序を定義するグループ化構造を表示することもできます。

【メソッド出力】領域で、【ビューのカスタマイズ】をクリックします。以下のいずれかのオプションを有効にします。

Sequence、Choice、および All

要素定義が sequence か choice か all かを、線で表示します。

sequence グループ内の要素は、階層内に定義された順序で並んでいる必要があります。

choice グループ内の少なくとも 1 つの要素が、応答メッセージに表示される必要があります。

全グループのすべての要素を応答メッセージ内に含める必要があります。

キー

【メソッド出力】領域にキーを表示します。【メソッド出力】領域には、各グループのキーが含まれています。【ポート】領域内の出力ポートに、キーを追加できます。

パススルーポート

【メソッド出力】領域には、パススルーポートが表示されます。パススルーポートとは、トランスフォーメーションを通過させるだけで変更せずにデータを渡すポートのことです。パススルーポートをメソッド出力から REST Web サービスコンシューマトランスフォーメーションの出力グループのいずれかに射影することができます。パススルーポートは、データを 1 回のみ受け取るため、応答メッセージ内のルートレベルにあります。

出力ポートへのメソッド出力のマッピング

トランスフォーメーションの出力マッピングが表示されているときに、出力グループと出力ポートが定義でき、出力ポートにメソッド出力要素をマップすることができます。

1. REST Web サービスコンシューマトランスフォーメーションを開きます。
2. 【ポート】ビューで、出力マッピングを表示します。
3. 以下のいずれかの方法を使用して、出力グループまたは出力ポートを【ポート】領域に追加します。

| 方法 | 説明 |
|---------------------------|--|
| 要素をドラッグ。 | 【メソッド出力】領域のグループまたは子要素を、【ポート】領域の空のカラムにドラッグします。【ポート】領域にグループをドラッグする場合、Developer ツールはポートを持たないグループを追加します。 |
| グループまたはポートを手動で追加。 | グループを追加するには、【新規】ボタンの隣の矢印をクリックし、【グループ】をクリックします。ポートを追加するには、【新規】ボタンの隣の矢印をクリックし、【フィールド】をクリックします。 |
| 別のトランスフォーメーションからポートをドラッグ。 | エディタ内で、ポートを別のトランスフォーメーションから REST Web サービスコンシューマトランスフォーメーションにドラッグします。 |
| ポートをコピー。 | 別のトランスフォーメーションからポートを選択し、【ポート】領域へコピーします。ポートをコピーするときは、キーボードショートカットを使用するか、Developer ツールの【コピー】ボタンと【貼り付け】ボタンを使用します。 |

4. ポートを手動で作成するか、別のトランスフォーメーションからポートをコピーするときは、**【ポート】** 領域の **【場所】** カラムをクリックし、リストから要素を選択します。
5. ポートの場所をクリアするには、以下のいずれかの方法を使用します。

| 方法 | 説明 |
|---------------------|--|
| 【クリア】 をクリック。 | 【ポート】 領域でポートを 1 つまたは複数選択し、 【クリア】 をクリックします。 |
| 要素とポートを結ぶ線を削除。 | メソッド出力の要素と出力ポートを結ぶ線を 1 つまたは複数選択し、キーボードの Delete キーを押します。 |

REST Web サービスコンシューマトランスフォーメーションの詳細プロパティ

REST Web サービスコンシューマトランスフォーメーションのデータがデータ統合サービスでどのように処理されるかを決定するためのプロパティを設定します。

【詳細】 タブで、以下のプロパティを設定します。

トレースレベル

このトランスフォーメーションのログに表示される情報の詳細度。Terse、Normal、Verbose Initialization、Verbose data から選択できます。デフォルトは **【Normal】** です。

接続

Web サービスに接続する HTTP 接続オブジェクトを特定します。Developer ツールで HTTP 接続を作成および編集します。HTTP 接続を設定するときは、ベース URL、Web サービスに必要なセキュリティのタイプ、および接続タイムアウトの期間を設定します。

REST Web サービスコンシューマトランスフォーメーションは、URL を使用して Web サービスに接続します。URL はトランスフォーメーションプロパティまたは HTTP 接続で定義できます。

以下のような状況で HTTP 接続を設定してください。

- URL の入力ポートを使用しない。
- Web サービスには HTTP 認証または SSL 証明書が必要です。
- デフォルトの接続タイムアウト期間を変更したほうがよい。

XML スキーマ検証

実行時間に応答メッセージを検証します。**【無効な XML でのエラー】** または **【検証なし】** が選択できます。

ソート済み入力

入力データを必ずしもすべて処理しなくても、データ統合サービスが出力を生成できるようにします。入力データが XML 入力階層内のキーでソートされる場合に、ソート済み入力を有効にします。

URL

REST Web サービスのベース URL。HTTP 接続内のベース URL は、この値より優先されます。

フォーマット

Web サービスの応答の形式。Web サービスの応答によって、**XML** または **JSON** を選択します。

REST Web サービスコンシューマトランスフォーメーションの作成

REST Web サービスコンシューマトランスフォーメーションは、再利用可能なものと再利用不可能なものが作成できます。再利用可能なトランスフォーメーションは、複数のマッピングで使用できます。再利用不可能なトランスフォーメーションは、単一のマッピングで使用されます。

REST Web サービスコンシューマトランスフォーメーションを作成する場合、要素と XML 階層を手動で定義するか、スキーマオブジェクトから要素と階層をインポートすることができます。このスキーマオブジェクトは、XML ファイルの場合もあれば、テキストファイルの場合もあります。

REST Web サービスコンシューマトランスフォーメーションの作成

REST Web サービスコンシューマトランスフォーメーションを作成するときは、メソッドを選択し、選択したメソッドに基づいてメソッドの入力と出力を定義します。

1. REST Web サービスコンシューマトランスフォーメーションを作成するには、以下のメソッドのいずれかを使用します。

| メソッド | 説明 |
|-------|---|
| 再利用可能 | [Object Explorer] ビューで、プロジェクトまたはフォルダを選択します。【ファイル】 > 【新規】 > 【トランスフォーメーション】 をクリックします。REST Web サービスコンシューマトランスフォーメーションを選択し、【次へ】 をクリックします。 |
| 再利用不可 | マッピングまたはマップレットで、REST Web サービスコンシューマフォーメーションをトランスフォーメーションパレットからマッピングまたはマップレットエディタにドラッグします。 |

2. トランスフォーメーション名を入力し、場所と HTTP メソッドを選択します。
3. 【次へ】 をクリックします。
4. メソッド入力を定義するために、以下のメソッドのいずれかを使用します。

| メソッド | 説明 |
|--------------------|----------------------------------|
| 空として作成 | XML の要素と階層を手動で定義します。 |
| スキーマオブジェクト内の要素から作成 | XML の要素と階層をスキーマオブジェクトからインポートします。 |

【メソッド入力の定義】 領域には、トランスフォーメーションの入力グループと入力ポートが表示されます。【マッピング入力】 領域には、要求メッセージ階層が表示されます。

5. 入力グループと入力ポートを定義し、入力要素に入力ポートをマップします。
6. 【次へ】 をクリックします。
7. メソッド出力を定義するには、【空として作成】 または 【スキーマオブジェクト内の要素から作成】 を選択します。

【メソッド出力の定義】 領域には、トランスフォーメーションの入力グループと入力ポートが表示されます。【マッピング出力】 領域には、要求メッセージ階層が表示されます。

8. 出力グループと出力ポートを定義し、要素を出力ポートにマップします。
9. 【完了】 をクリックします。

配列を含む JSON 応答メッセージの解析

要素が複合型の子要素であり、この要素の最大オカレンスが無制限である場合、スキーマは無効です。JSON パーサーは、要素の複数のインスタンスを抽出することを制限します。

複合型の子要素の最大オカレンスは、スキーマ内の複合型に対して 0 または 1 にし、順序インジケータを choice にする必要があります。最大オカレンスを 1 に変更してスキーマを検証する場合、一度に抽出できる要素のインスタンスは 1 つです。

スキーマ内の複合型の choice 順序インジケータでは、最大オカレンスを無制限として使用できます。

JSON 応答メッセージの例

次のスキーマは、複合型要素 xmlRoot に最大オカレンスが無制限である要素名 Likes が含まれています。

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="xmlRoot">
    <xs:complexType>
      <xs:all>
        <xs:element type="xs:byte" name="Age"/>
        <xs:element type="xs:string" name="FirstName"/>
        <xs:element type="xs:string" name="Likes" maxOccurs="unbounded" minOccurs="0"/>
        <xs:element type="xs:string" name="FamilyName"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

JSON 応答は、次の形式で変更できます。

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="xmlRoot">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element type="xs:byte" name="Age"/>
        <xs:element type="xs:string" name="FirstName"/>
        <xs:element type="xs:string" name="Likes"/>
        <xs:element type="xs:string" name="FamilyName"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

<xs:choice maxOccurs="unbounded">を使用すると、コンテンツを任意の順序で 1 回以上繰り返すことができます。

応答メッセージ内の名前なし配列

REST Web サービスコンシューマ変換では、要求メッセージではなく応答メッセージのみで名前なし配列をサポートします。メソッド出力定義で定義された名前なし配列スキーマを解析するには、複合型または単純型の配列要素の親要素が xmlRoot という名前である必要があります。








REST Web サービスコンシューマトランスフォーメーションでは、最大発生数が「バインドなし」に設定された xmlRoot 要素の子要素として xmlRoot を定義し、名前なし配列の要素を xmlRoot 要素の子要素として定義する必要があります。

次の図は、名前なし配列の定義済みメソッド出力を示しています。

☐ Ports ☐ Method input ☒ Method output

Show: ☒ Method output definition ☐ Output mapping

Method output definition

| | Name | Type | Min... | Ma... | Description | >> |
|--|---|---------------|--------|-------|-------------|----|
| |  Rest_Consume... | (Rest_Cons... | | | | |
| |   xmlRoot | (xmlRoot) | 1 | 1 | | |
| |   xmlRoot | (xmlRoot) | 1 | Un... | | |
| |  emp | xs:string | 1 | 1 | | |
| |  empid | xs:string | 1 | 1 | | |

第 13 章

REST Web サービスコンシューマ トランスフォーメーションの使用 例

この章では、以下の項目について説明します。

- [データを受信できるよう REST Web サービスコンシューマトランスフォーメーションを設定する方法, 162 ページ](#)
- [スキーマを使用して REST Web サービスコンシューマトランスフォーメーションを作成する方法, 169 ページ](#)
- [REST Web サービスコンシューマトランスフォーメーションでカスタム URL を生成する方法, 180 ページ](#)

データを受信できるよう REST Web サービスコンシューマトランスフォーメーションを設定する方法

REST Web サービスコンシューマトランスフォーメーションを使用して Web サービスから未解析または解析済みのデータを受信します。応答メッセージを指定するとき、元のメッセージを受信することを選択することも、

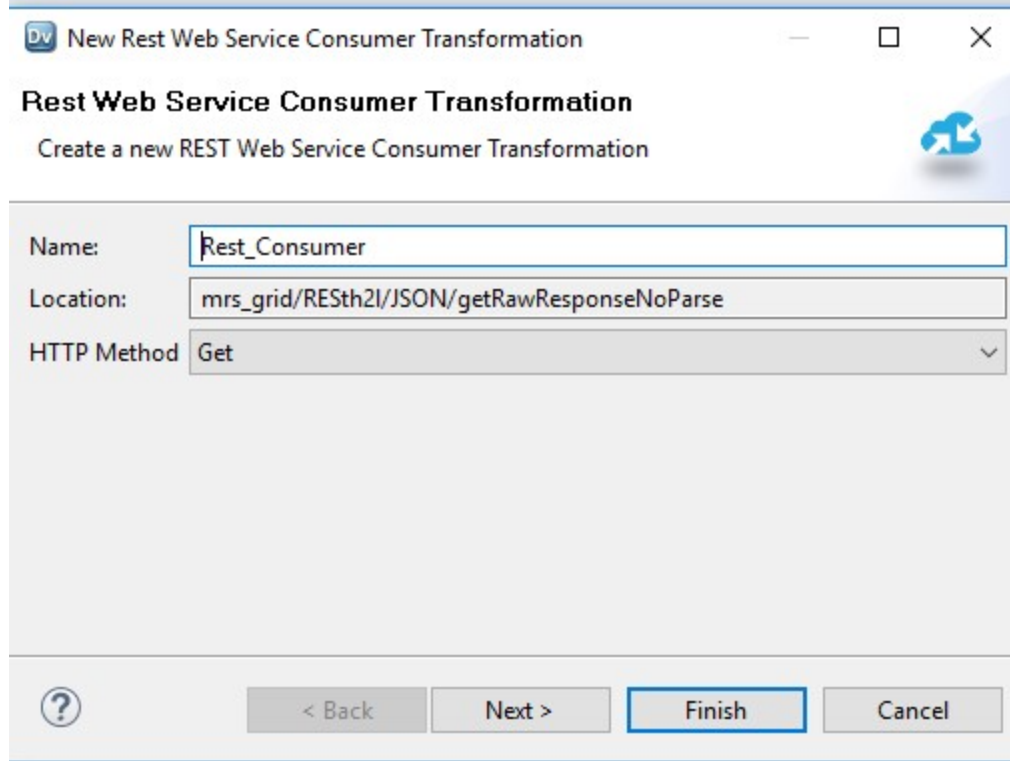
メッセージの要素を表す事前定義されたフィールドを指定することによってメッセージを解析することを選択することもできます。

Web サービスから未解析のデータを受信する方法

トランスフォーメーションを作成するには、次の手順を実行します：

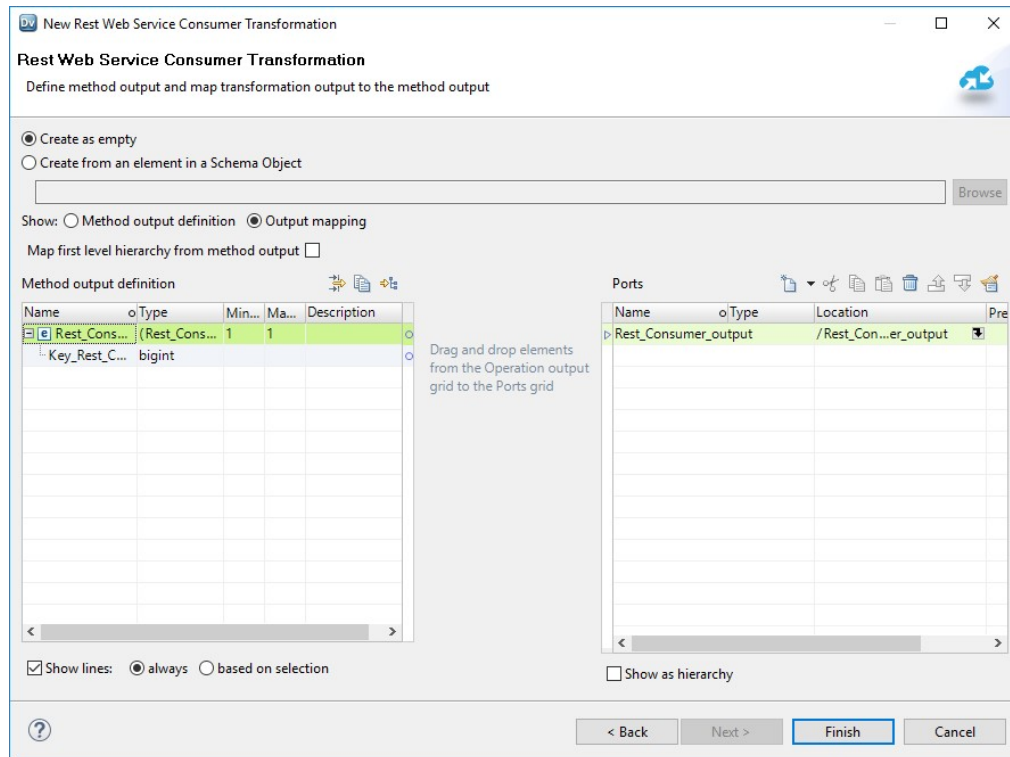
1. Developer tool マッピングエディタで、マッピングパレットを下にスクロールして、REST Web サービス コンシューマトランスフォーメーションを見つけ、マッピングにドラッグします。

[新しい REST Web サービスコンシューマトランスフォーメーション] ウィザードが表示されます。

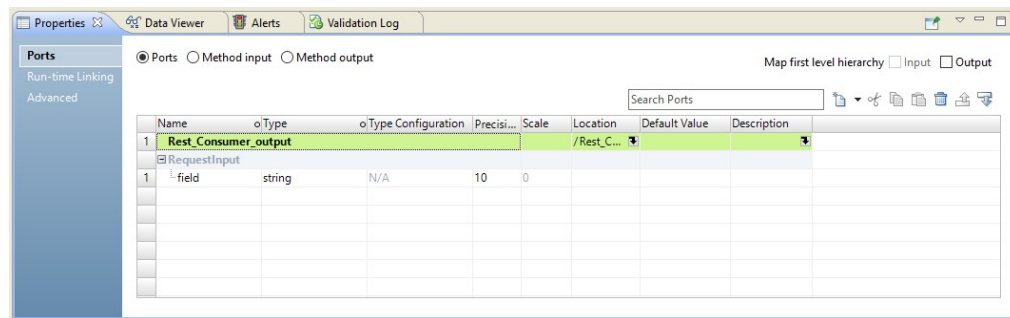


2. トランスフォーメーションの名前を入力します。
3. **[HTTP メソッド]** の隣にある **[Get]** を選択します。
Get メソッドを使用するときは、入力マッピングを設定しません。
4. **[次へ]** をクリックします。
Get メソッドを使用しているときは入力は適用されないというメッセージがウィザードに表示されます。
5. **[次へ]** をクリックします。
6. **[空として作成]** を選択します。
7. **[表示]** の隣にある **[出力マッピング]** を選択します。

次の図はウィザードのオプションを示しています:

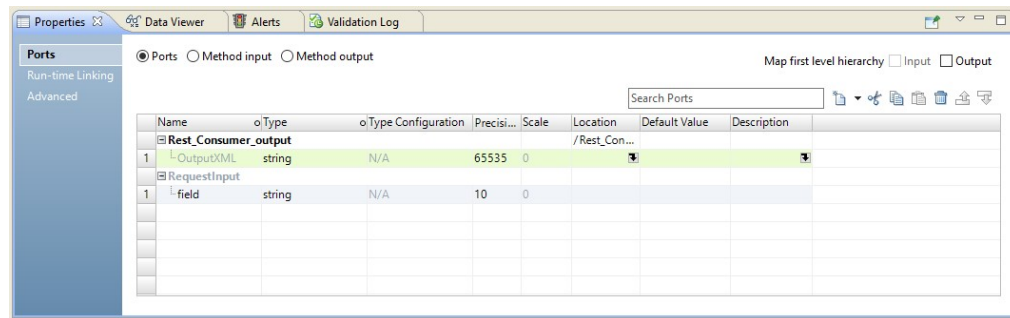


8. 出力グループを左の【メソッド出力定義】エリアから右の【ポート】エリアへドラッグします。
9. 【完了】をクリックします。
10. REST Web サービスコンシューマトランスフォーメーションの【ポート】タブに移動します。
11. 出力グループをハイライトします。



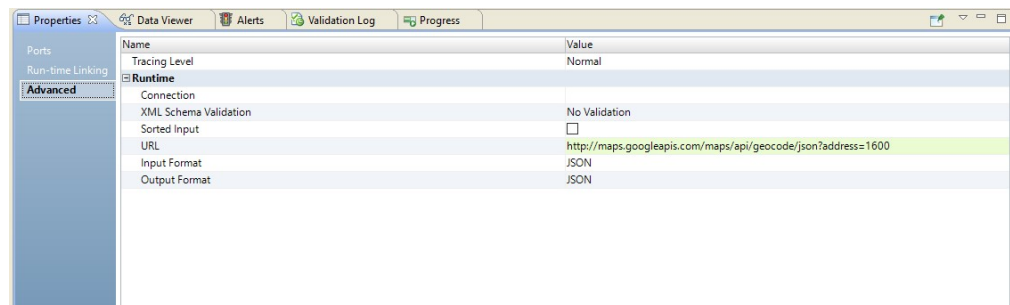
12. 【ポート】タブの右上にある【新規】を選択し、【その他のポート】を選択します。【出力 XML】の隣にあるボックスをチェックし、【OK】をクリックします。

次の図のように、[出力 XML] ポートは出力グループの下にネストされています：



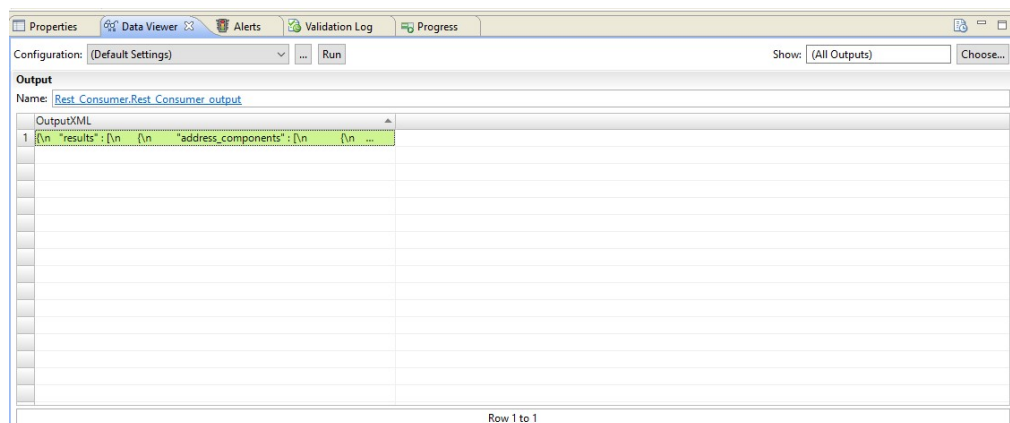
13. [詳細] タブに移動します。
14. Web サービスに接続するには、URL プロパティにベース URL を設定します。
15. HTTP 接続を使用する場合は、接続名をプロパティ **[接続]** に入力します。HTTP 接続は、プロパティ「URL」に設定されたベース URL を上書きします。
16. 要求メッセージと応答メッセージの形式を入力します。**[入力形式]** と **[出力形式]** を XML または JSON で設定します。

次の図は詳細プロパティを設定する場所を示しています：



17. データビューアを実行するか、マッピングを実行します。

次の図は出力 XML ポートの応答の一例を示しています：



Web サービスから解析されたデータを受信する方法

Web サービスから受信したメッセージを解析することができます。メッセージを解析するとき、REST Web サービスコンシューマトランスフォーメーションは、メソッド出力定義で応答メッセージに指定した構造に従ってデータを受信します。

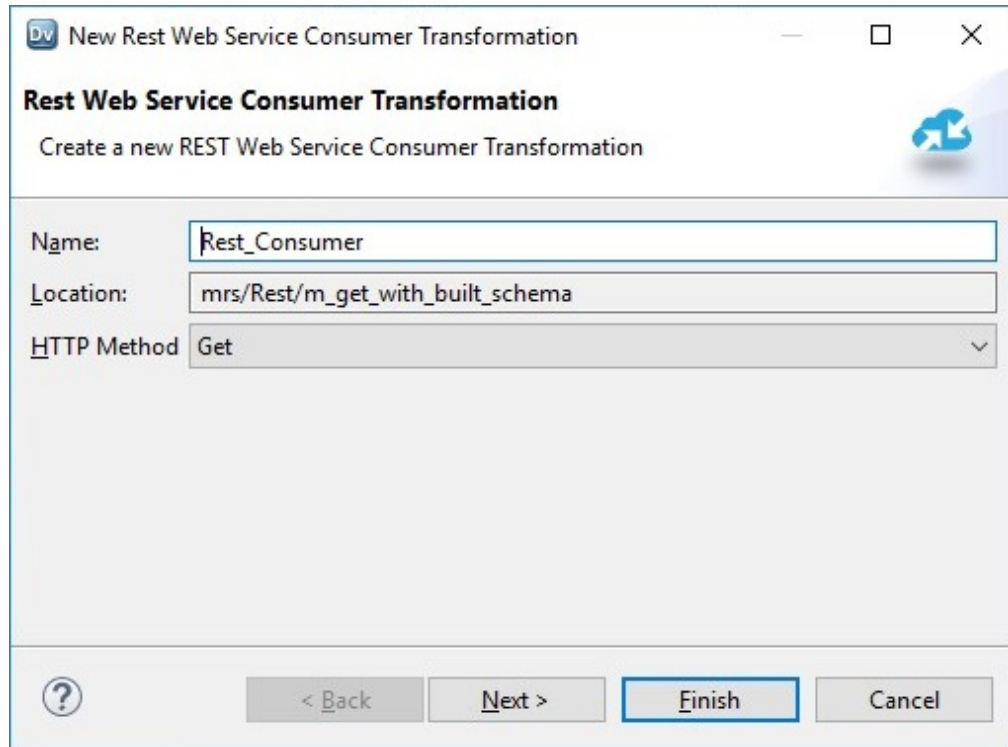
例えば、次の JSON 応答メッセージを解析するとします:

```
{"name": "infa", "message": "hello"}
```

この応答メッセージを解析するには、トランスフォーメーションを設定するときに次の手順を実行します:

1. Developer tool マッピングエディタで、マッピングパレットを下にスクロールして、REST Web サービスコンシューマトランスフォーメーションを見つけ、マッピングにドラッグします。

[新しい REST Web サービスコンシューマトランスフォーメーション] ウィザードが表示されます。



2. トランスフォーメーションの名前を入力します。
3. **[HTTP メソッド]** の隣にある **[Get]** を選択します。
Get メソッドを使用するときは、入力マッピングを設定しません。
4. **[次へ]** をクリックします。
Get メソッドを使用しているときは入力に適用されないというメッセージがウィザードに表示されます。
5. **[次へ]** をクリックします。
6. **[空として作成]** を選択します。
7. **[表示]** の隣にある **[出力マッピング]** を選択します。
8. 受信することが予測される応答メッセージに基づいてメソッド出力定義を設計します。

☐ Ports
 ☐ Method input
 ☒ Method output

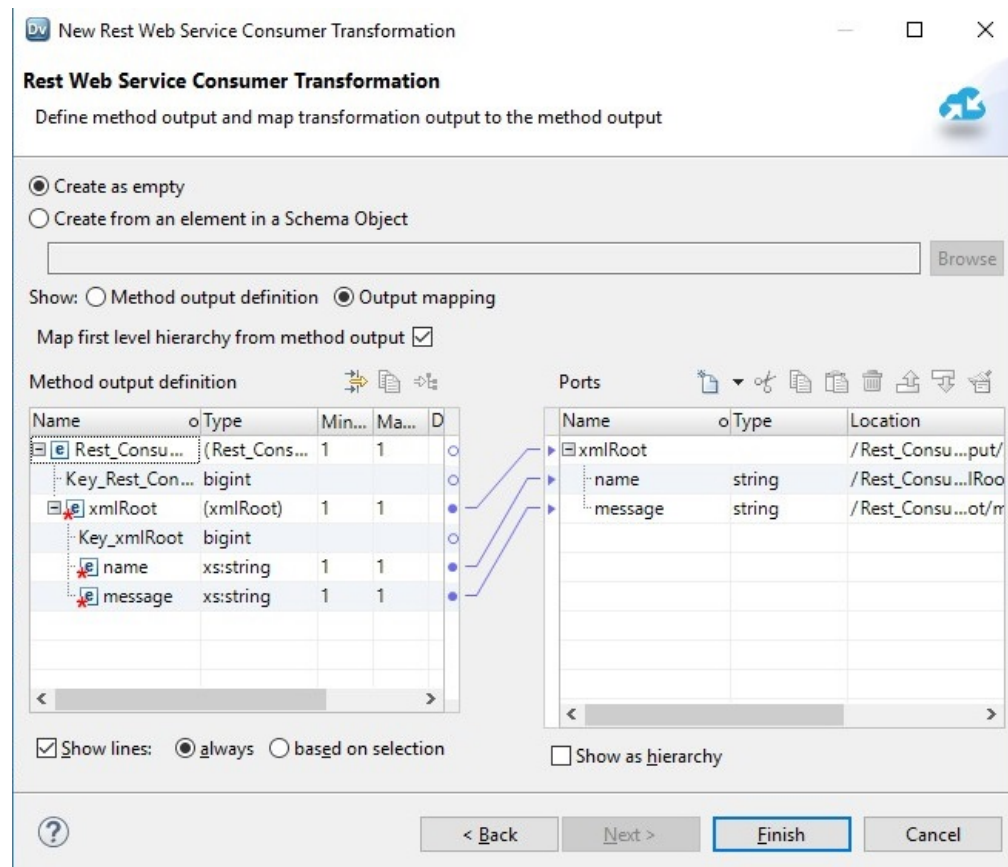
Show: ☒ Method output definition
 ☐ Output mapping

Method output definition

| Name | Type | Min... | Ma... | Description | >> |
|-----------------------------|---------------|--------|-------|-------------|----|
| Rest_Consumer_output | (Rest_Cons... | | | | |
| xmlRoot | (xmlRoot) | 1 | 1 | | |
| name | xs:string | 1 | 1 | | |
| message | xs:string | 1 | 1 | | |

9. **【出力マッピング】**を選択します。
10. ポートを**【メソッド出力定義】**エリアから**【ポート】**エリアにドラッグして出力マッピングを設定します。

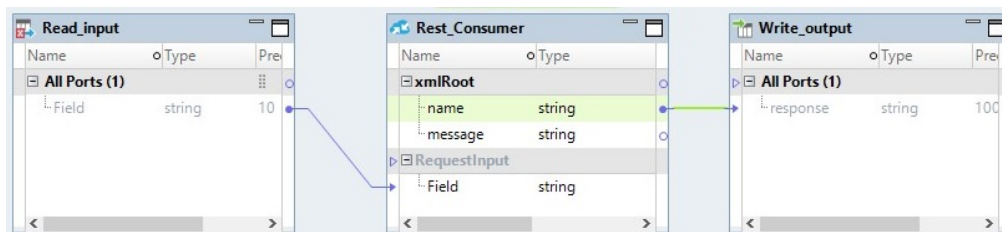
次の図は、出力マッピングの例を示しています:



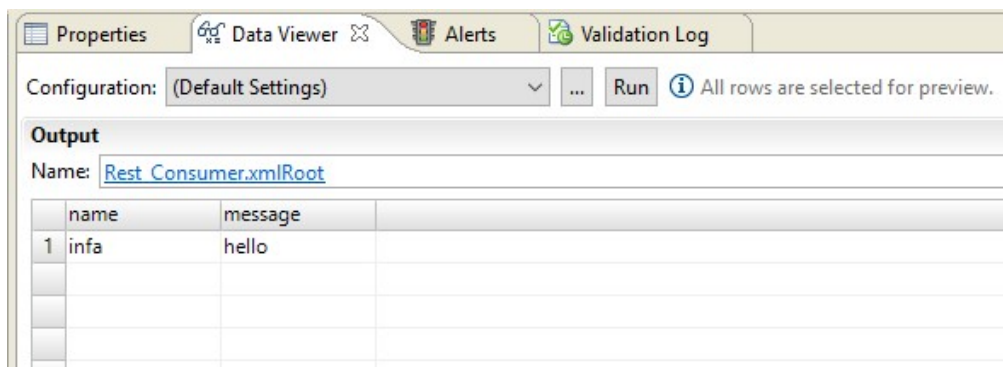
11. **完了** をクリックします。
12. トランスフォーメーションを選択して **詳細** タブに移動します。

13. 【出力形式】を【JSON】に設定します。

次の図はトランスフォーメーションがマッピングに統合されたときにどのように表示されるかを示しています：



次の図はマッピングを実行したときに応答がどのように表示されるかを示しています：



スキーマを使用して REST Web サービスコンシューマトランスフォーメーションを作成する方法

トランスフォーメーションに多くの要素が含まれており手動で設定するのが困難な場合はスキーマを使用して REST Web サービスコンシューマトランスフォーメーションを作成します。トランスフォーメーションの要求メッセージおよび応答メッセージに含めるすべての要素を含むスキーマをインポートします。

XML フォーマットで要求メッセージおよび応答メッセージを処理する方法

REST Web サービスコンシューマトランスフォーメーションが Web サービスに送信する要求メッセージを作成するのに使用する要素階層が含まれた XML ファイルがあります。その XML ファイルには組織内の人物を特定するのに使用できるデータ (ID、名前、メール、国など) が含まれています。これらの要素は XML ファイルに次のように表示されます：

```
<Person><ID>sample</ID><Name>sample</Name><Email>sample</Email><Country>sample</Country></Person>
```

応答メッセージを作成するのに使用する要素階層が含まれた別の XML ファイルがあります。その XML ファイルには各人物に応答するとき使用する挨拶の種類に関するデータが含まれています。その挨拶には各個人の ID および挨拶内容に関するデータが含まれています。これらの要素は XML ファイルに次のように表示されます：

```
<Greeting><ID>sample</ID><Content>sample</Content></Person>
```

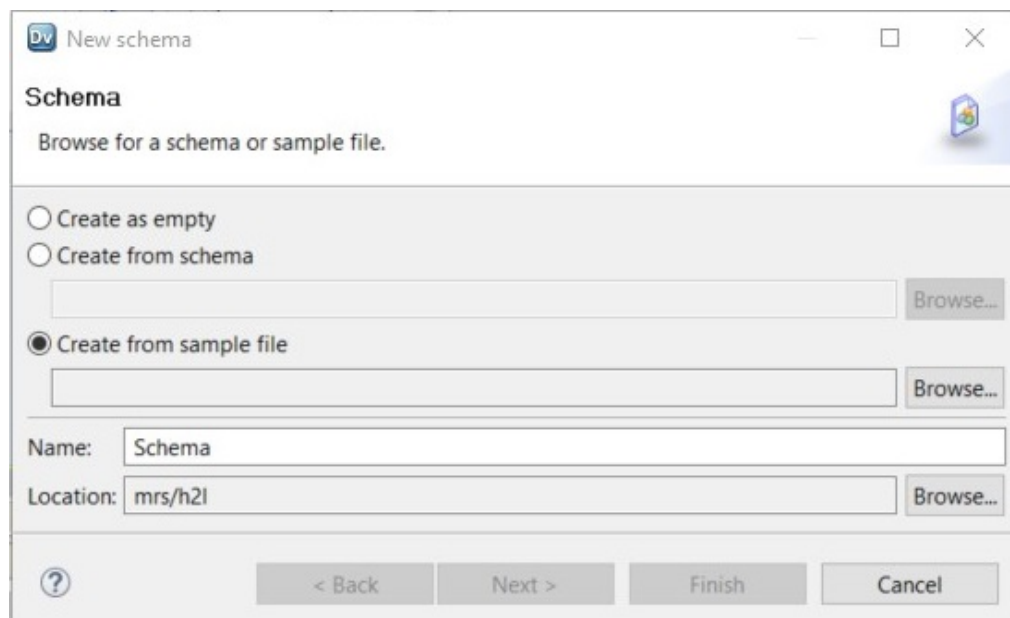
REST Web サービスコンシューマトランスフォーメーションでこれらの XML ファイルの要素階層を使用するには、まず Developer tool で XML ファイルをスキーマオブジェクトとしてインポートします。次にそのスキーマ

マオブジェクトを使用して、REST Web サービスコンシューマトランスフォーメーションを作成するときのメソッド入力およびメソッド出力を定義します。

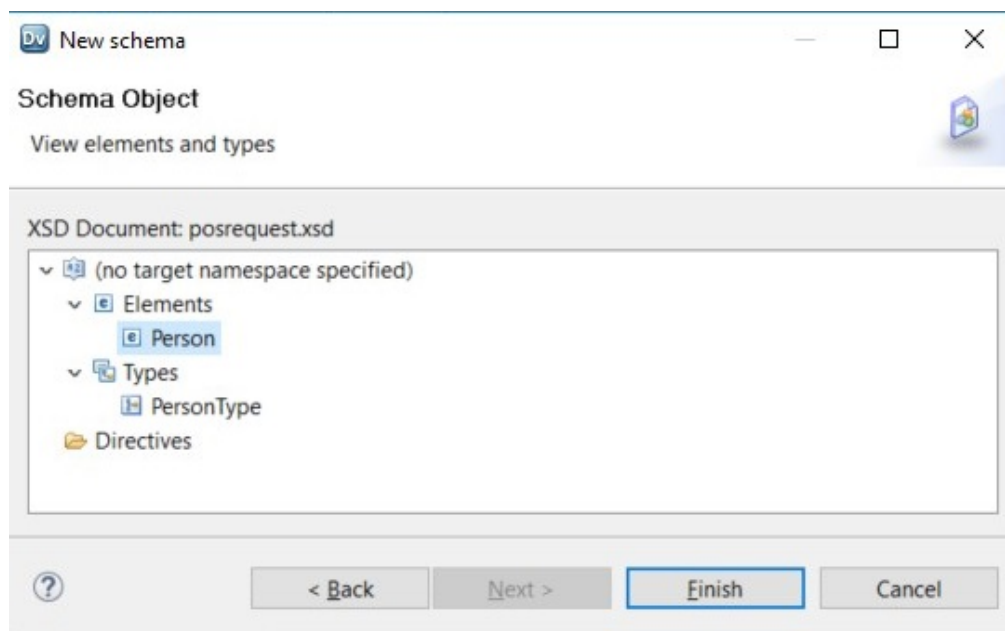
手順 1.XML ファイルからスキーマを作成する。

次の手順を実行してメソッド入力用のスキーマオブジェクトを作成します。その手順を繰り返してメソッド出力用のスキーマオブジェクトを作成します。

1. Developer tool で、[ファイル] - [新規] - [スキーマ] とクリックしてスキーマウィザードを開きます。
2. [サンプルファイルから作成する] を選択します。
3. [参照] をクリックします。
4. XML ファイルに移動し、[開く] をクリックします。
5. スキーマの名前を入力し、プロジェクトの場所を確認します。必要に応じて、新しい場所を参照します。



6. [次へ] をクリックしてスキーマ要素を表示します。



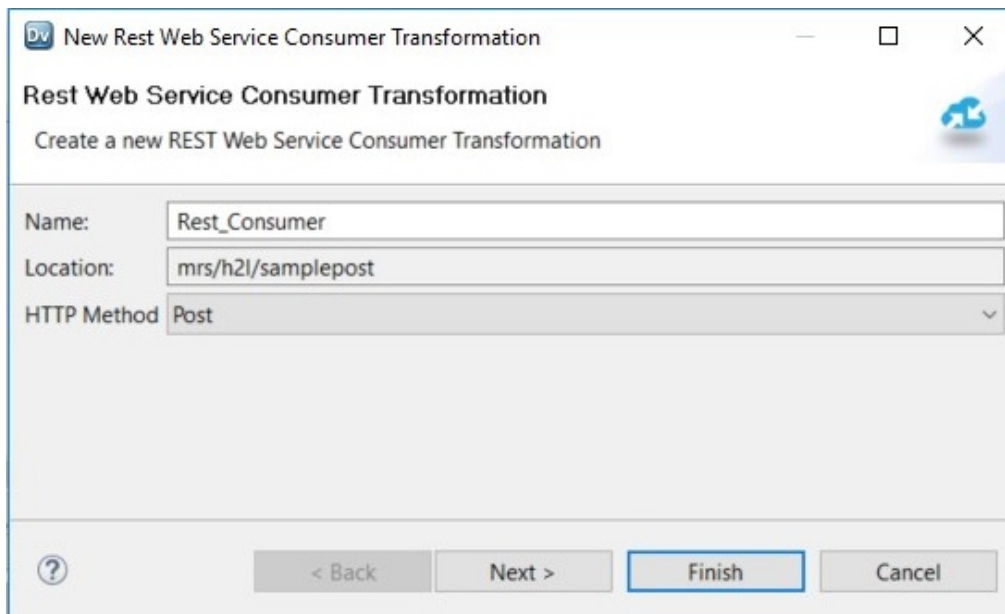
7. [完了] をクリックします。
作成したスキーマオブジェクトを使用してトランスフォーメーションにメソッド入力を定義できます。
8. 手順 1 から 7 を繰り返してメソッド出力用のスキーマオブジェクトを作成します。

手順 2.REST Web サービスコンシューマトランスフォーメーションを作成します。

トランスフォーメーションを作成します。スキーマを使用してメソッド入力とメソッド出力を定義します。

1. Developer tool マッピングエディタで、マッピングパレットを下にスクロールして、REST Web サービスコンシューマトランスフォーメーションを見つけ、マッピングにドラッグします。
2. トランスフォーメーションの名前を入力します。

3. HTTP メソッドを選択します。この例では、HTTP メソッド「Post」を選択します。



New Rest Web Service Consumer Transformation

Rest Web Service Consumer Transformation

Create a new REST Web Service Consumer Transformation

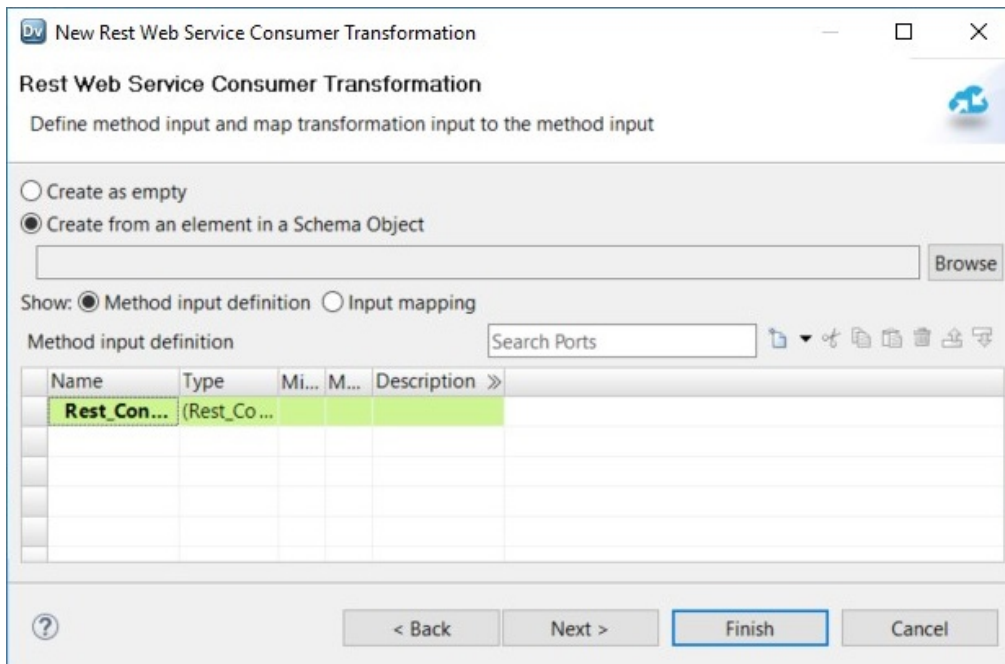
Name: Rest_Consumer

Location: mrs/h2l/samplepost

HTTP Method: Post

< Back Next > Finish Cancel

4. [次へ] をクリックしてメソッド入力を定義します。
5. [スキーマオブジェクトで要素から作成] を選択します。



New Rest Web Service Consumer Transformation

Rest Web Service Consumer Transformation

Define method input and map transformation input to the method input

☐ Create as empty

☒ Create from an element in a Schema Object

Browse

Show: ☒ Method input definition ☐ Input mapping

Method input definition

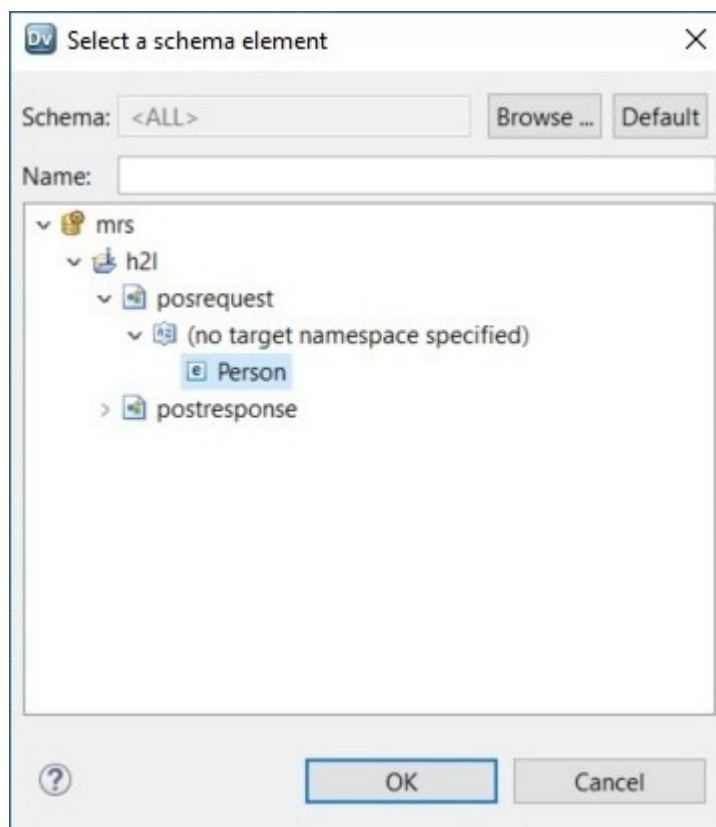
| Name | Type | Mi... | M... | Description >> |
|-------------|--------------|-------|------|----------------|
| Rest_Con... | {Rest_Co ... | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Search Ports

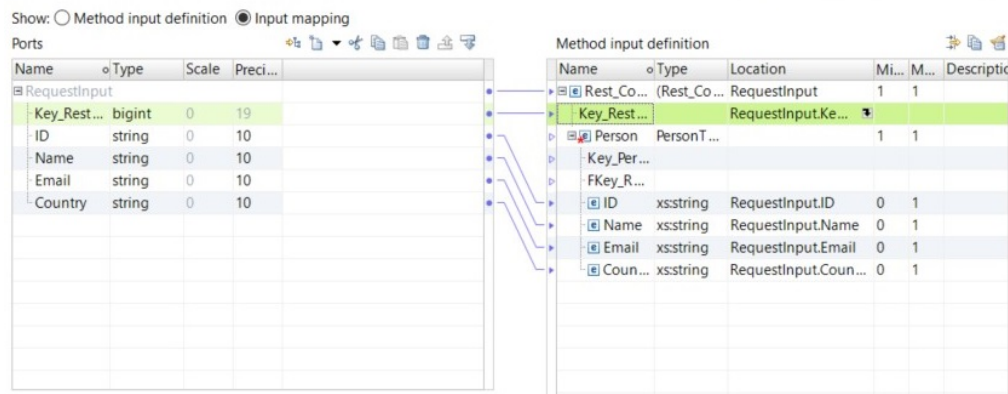
< Back Next > Finish Cancel

6. [参照] をクリックします。

7. スキーマ内の親要素を選択します。この例では、メソッド入力に使用されているスキーマには親要素 Person が含まれています。

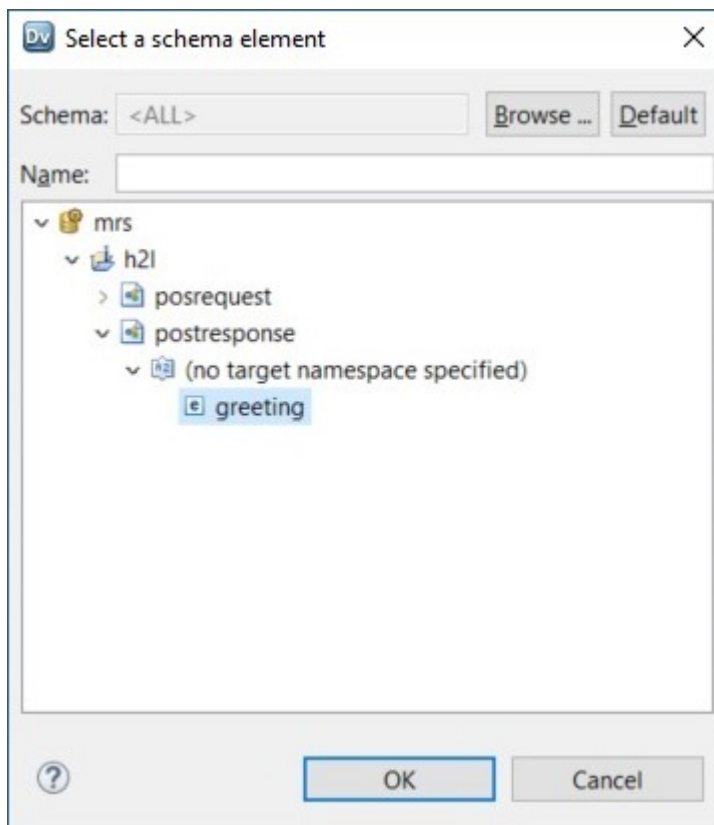


8. [OK] をクリックします。
9. [入力マッピング] を選択します。
10. [ポート] エリアの [RequestInput] にある要素を [メソッド入力定義] エリアの [Rest_Consumer_Input] にマッピングします。必ずキーを [ポート] エリアから [メソッド入力定義] エリアにマッピングしてください。

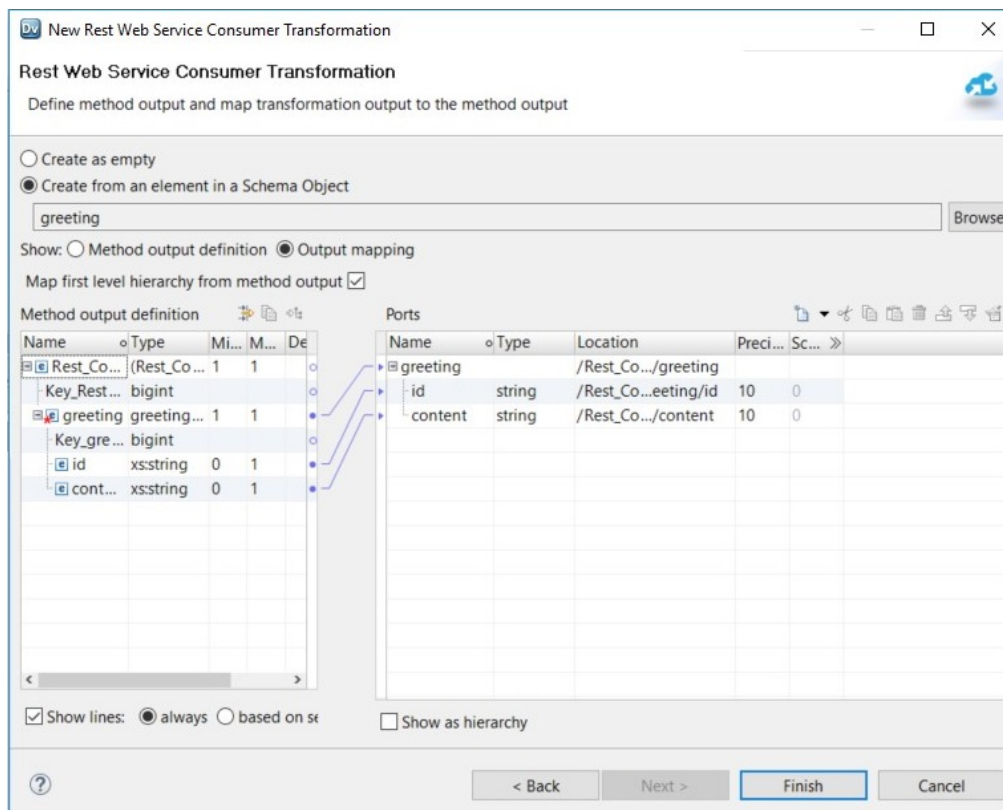


11. メソッド出力を定義するには、トランスフォーメーションウィザードで [次へ] をクリックします。
12. [スキーマオブジェクトで要素から作成] を選択します。

13. スキーマ内の親要素を選択します。この例では、メソッド出力に使用するスキーマに親要素 greeting が含まれています。

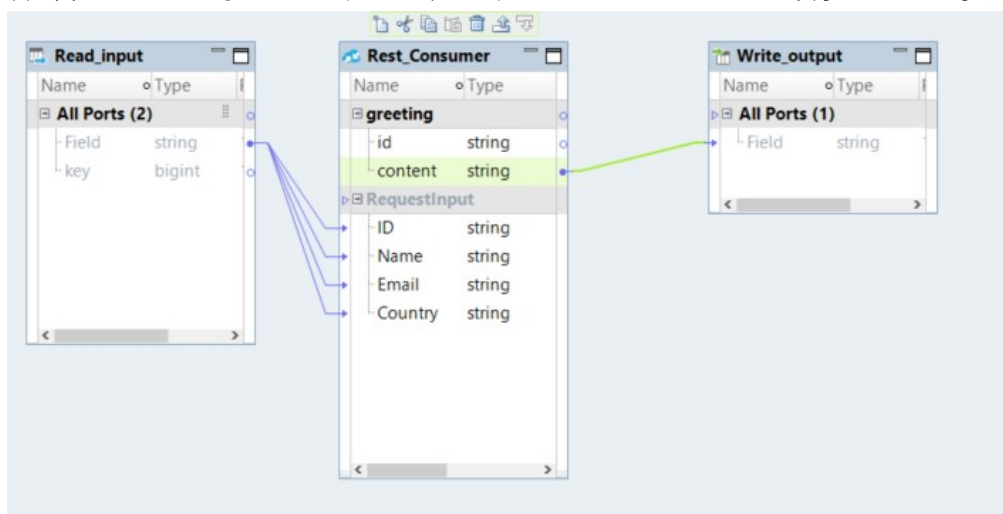


14. [メソッド出力定義] エリアの [Rest_Consumer_Output] 下にある要素を [ポート] エリアにマッピングします。キーをマッピングする必要はありません。



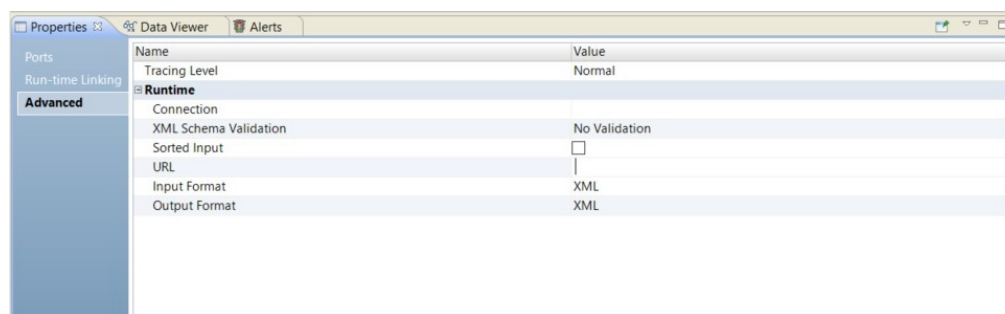
15. [完了] をクリックします。

次の図はトランスフォーマーがマッピングワークフローでどのように表示されるかを示しています:



16. マッピングエディタで REST Web サービスコンシューマトランスフォーマーを選択します。
17. [プロパティ] ビューに移動し、[詳細] タブを選択します。

18. 詳細プロパティを設定します。ベース URL を入力するには、プロパティ **【URL】** を設定するか、プロパティ **【接続】** で HTTP 接続を入力します。



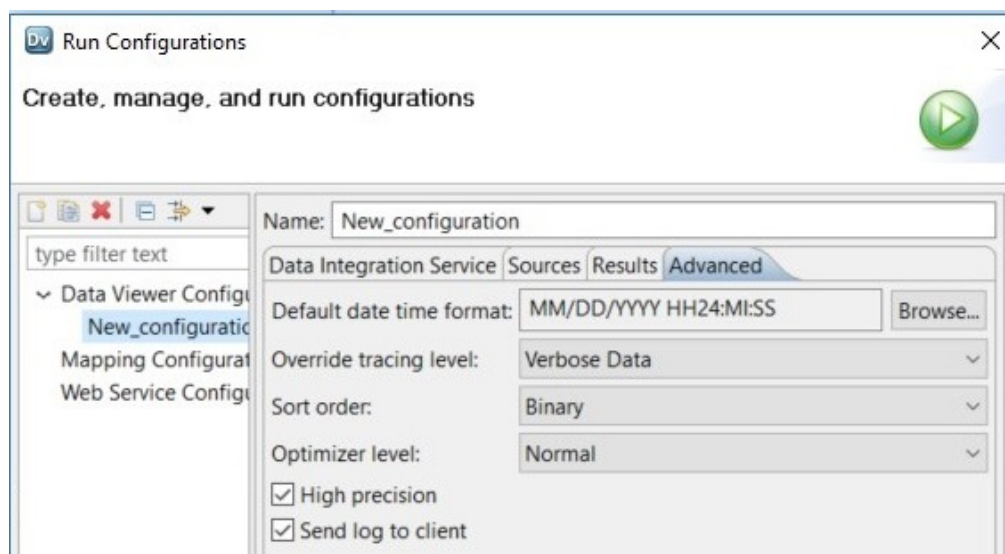
JSON フォーマットで解析メッセージおよび応答メッセージを扱う方法

JSON 応答を解析するように REST Web サービスコンシューマトランスフォーメーションを設定するには、Developer tool の設定でログレベルを **【詳細 - データ】** に設定します。詳細データログを開き、応答を XML ファイルにコピーします。XML ファイルをスキーマオブジェクトとしてインポートし、そのスキーマを使用してトランスフォーメーションを作成します。トランスフォーメーションを作成した後、JSON 応答を受信できるようにそのトランスフォーメーションを設定します。

手順 1. ログレベルを **【詳細 - データ】** に設定する。

ログレベルを **【詳細 - データ】** に設定するには、次の手順を実行します：

1. Developer tool で、**【ウィンドウ】 - 【設定】** をクリックします。
2. **【設定】** ウィンドウの左側で、**【実行設定】 - 【マッピング】** をクリックします。
3. 右側で、**【詳細】** タブを選択します。
4. プロパティ **【オーバーライドトレースレベル】** については、**【詳細 - データ】** を選択します。



5. 詳細データログを開きます。
6. **HTTP Response BODY: <?xml** を検索します

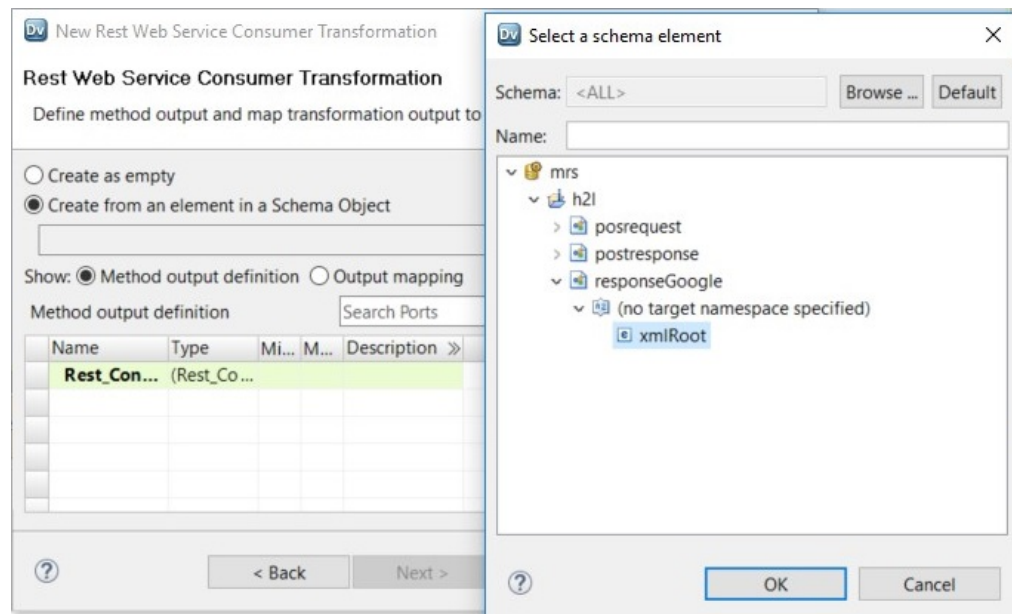
7. <xmlRoot>と</xmlRoot>との間のデータをコピーし、データを XML ファイルとして保存します。
8. Developer tool で XML ファイルをスキーマオブジェクトとしてインポートします。

手順 2.このスキーマを使用して REST Web サービスコンシューマトランスフォーメーションを作成します。

スキーマを使用して REST Web サービスコンシューマトランスフォーメーションが受信するデータを解析するには、次の手順を実行します：

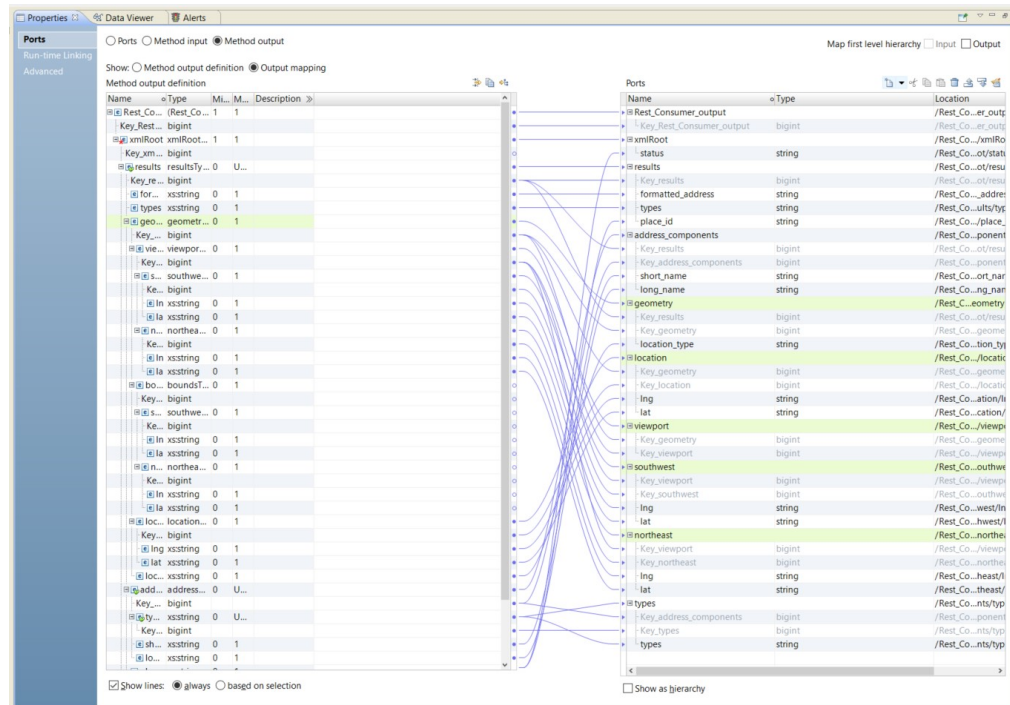
1. Developer tool マッピングエディタで、マッピングパレットを下にスクロールして、REST Web サービスコンシューマトランスフォーメーションを見つけ、マッピングにドラッグします。
2. トランスフォーメーションの名前を入力します。
3. HTTP メソッドを選択します。
4. [次へ] をクリックします。
5. メソッド入力を定義するには、[空として作成] または [スキーマオブジェクト内の要素から作成] を選択します。メソッド入力を定義するときにメソッド入力定義と入力マッピングを定義します。
6. [次へ] をクリックします。
7. メソッド出力を定義するには、[スキーマオブジェクト内の要素から作成] を選択します。
8. [参照] をクリックします。
9. 要素 [xmlRoot] を選択します。

次の図は要素 xmlRoot がどのように表示されるかを示しています：



10. 出力マッピングを設定します。親要素に子要素が含まれている場合は、子要素を手動でマッピングする必要があります。[メソッド出力定義] エリアから [ポート] エリアに親要素および子要素のグループをドラッグアンドドロップします。

次の図のようなマッピングになります:



- REST Web サービスコンシューマトランスフォーメーションプロパティを設定します。プロパティ [出力形式] については [JSON] を指定します。

12. トランスフォーメーションを実行します。

次の図は受信する応答の一例を示しています：

Configuration: (Default Settings) Run Show: (All Outputs) Choose...

Output

Name: [Rest Consumer1.Rest Consumer output](#)

| Key_Rest_... |
|--------------|
| 1 1 |

Row 1 to 1

Name: [Rest Consumer1.xmlRoot](#)

| status |
|--------|
| 1 OK |

Row 1 to 1

Name: [Rest Consumer1.results](#)

| Key_results | formatted... | types | place_id |
|-------------|----------------|---------------|---------------|
| 1 1 | 4000 SE 8... | street_add... | ChIJ_ZYPcj... |
| 2 2 | La Jota Hal... | premise | ChIJyQVpn... |
| 3 3 | KY-1600, K... | route | ChIJKabb6... |
| 4 4 | NC-1600,... | route | ChIJ7Z6kO... |
| 5 5 | NC-1600 | route | ChIJX2Av |

Row 1 to 5

Name: [Rest Consumer1.address.components](#)

| Key_results | Key_addre... | short_name | long_name |
|-------------|--------------|-------------|--------------|
| 1 1 | 1 | 1600 | 1600 |
| 2 1 | 2 | 4000 | 4000 |
| 3 1 | 3 | SE 82nd Ave | Southeast... |
| 4 1 | 4 | Lents | Lents |
| 5 1 | 5 | Portland | Portland |

Row 1 to 30

Name: [Rest Consumer1.types](#)

| Key_addre... | Key_types | types |
|--------------|-----------|--------------|
| 1 1 | 1 | subpremise |
| 2 2 | 2 | street_nu... |
| 3 3 | 3 | route |
| 4 4 | 4 | neighborh... |
| 5 4 | 5 | politica... |

Row 1 to 48

Name: [Rest Consumer1.geometry](#)

| Key_results | Key_geom... | location_ty... |
|-------------|-------------|----------------|
| 1 1 | 1 | ROOFTOP |
| 2 2 | 2 | ROOFTOP |

応答に含まれるデータが親要素とネストされた子要素で構成されるグループを形成するように解析されていることに注意してください。

REST Web サービスコンシューマトランスフォーメーションでカスタム URL を生成する方法

REST Web サービスコンシューマトランスフォーメーションは、ベース URL を使用して Web サービスに接続します。ベース URL は URL ポートまたは引数ポートを定義することでカスタマイズできます。

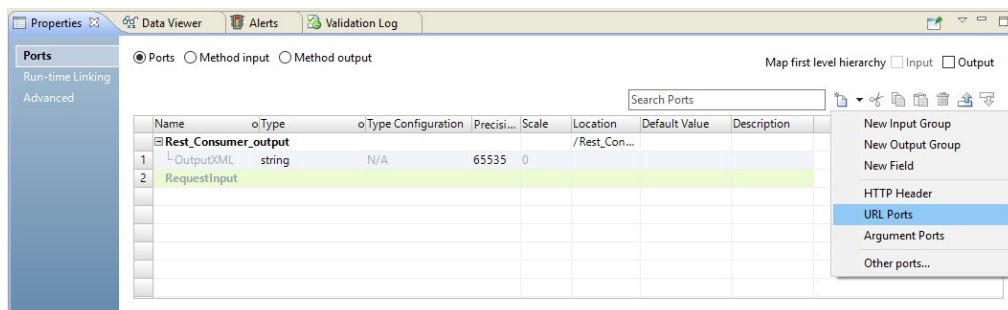
動的 URL を作成するように URL ポートを設定するか、Web サービスでアクセスするリソースを URL を使用してフィルタするように引数ポートを設定します。データ統合サービスでは、URL へ値を付加するか、ベース URL に引数ポートを付加します。

URL ポートを使用して動的 URL を構築する方法

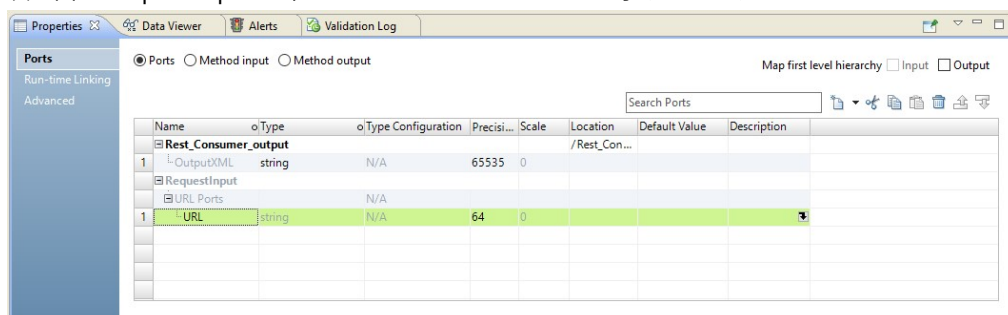
REST Web サービスコンシューマトランスフォーメーションにおいて URL ポートを使用して動的 URL を構築します。トランスフォーメーションに URL ポートを定義することによって `https://sample.com/1/posts` や `https://sample.com/2/posts` のような動的 URL を構築できます。

REST Web サービスコンシューマトランスフォーメーションを作成した後で動的 URL を設定するには、次の手順を実行します：

1. REST Web サービスコンシューマトランスフォーメーションを選択します。
2. **【プロパティ】** ビューに移動し、**【詳細】** タブを選択します。
3. ベース URL を設定するには、次の手順のいずれかを実行します：
 - **【URL】** プロパティを設定するには、ベース URL を入力します。
 - **【接続】** プロパティを設定するには、HTTP 接続を選択します。
4. **【ポート】** タブに移動します。
5. **【RequestInput】** グループをハイライトし、**【新規】** を選択します。ドロップダウンメニューから **【URL ポート】** を選択します。



次の図は RequestInput の下にネストされた URL ポートを示しています：



6. URL ポートの名前を編集します。

この例では、URL ポートの名前を number に変更します。

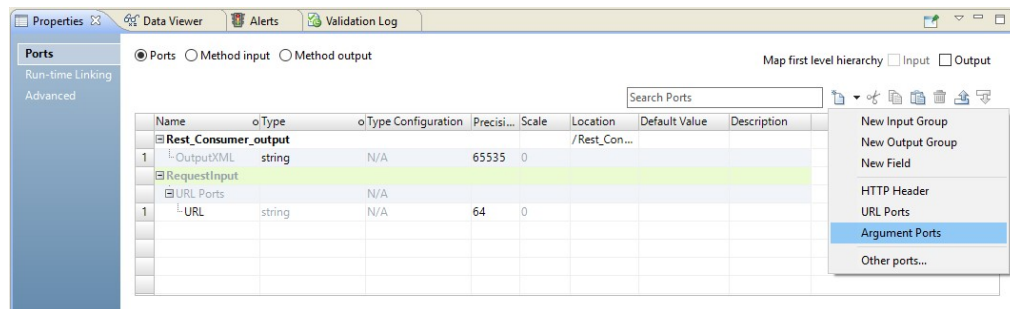
- Post 用にもう一つ URL ポートを作成します。ポートの名前を posts に変更します。
- `https://sample.com/1/posts` や `https://sample.com/2/posts` のような動的 URL を構築するには、アップストリームのマッピングオブジェクトから REST Web サービスコンシューマトランスフォーメーションの URL ポートにポートをリンクします。数値ポートを URL ポート number に、文字列ポートを URL ポート posts にリンクします。

引数ポートを使用して URL パラメータを生成する方法

引数ポートを使用して URL 内のパラメータを生成すると、Web サービスから要求するデータをフィルタすることができます。例えば、Get メソッドを使用する場合、メソッド入力定義や入力マッピングを定義することはできませんが、Get 要求のフィルタを使用することはできます。引数ポートを使用して要求メッセージ内にフィルタを作成することができます。引数ポートを使用して URL パラメータを生成し、`https://sample.com?arg1=val1&arg2=val2` のような URL を構築します。

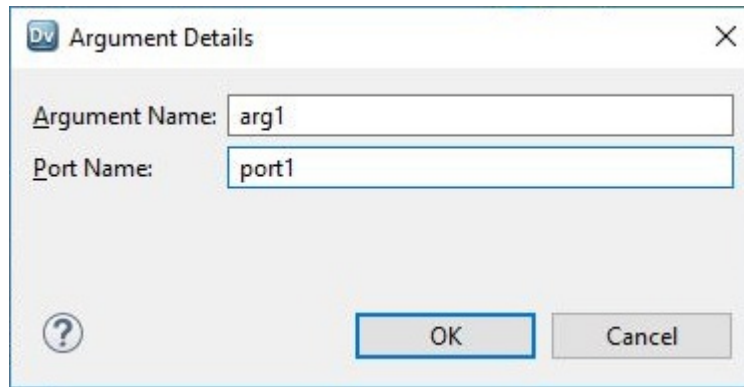
REST Web サービスコンシューマトランスフォーメーションを作成した後で引数ポートを設定するには、次の手順を実行します:

- REST Web サービスコンシューマトランスフォーメーションを選択します。
- 【プロパティ】** ビューに移動し、**【詳細】** タブを選択します。
- ベース URL を設定するには、次の手順のいずれかを実行します:
 - 【URL】** プロパティを設定するには、ベース URL を入力します。
 - 【接続】** プロパティを設定するには、HTTP 接続を選択します。
- 【ポート】** タブに移動します。
- 【RequestInput】** グループをハイライトし、**【新規】** を選択します。ドロップダウンメニューから **【引数ポート】** を選択します。



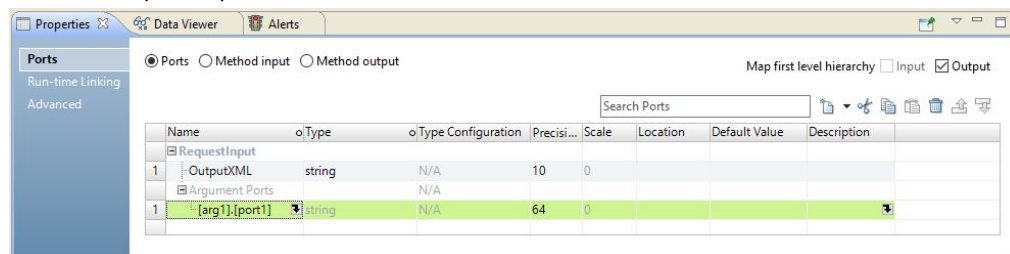
- 【引数名】** および **【ポート名】** の値を入力します。

ポート名にデータ統合サービスが処理できない文字が含まれている場合は、ポート名を変更します。



The 'Argument Details' dialog box has a title bar with a close button. It contains two text input fields: 'Argument Name' with the value 'arg1' and 'Port Name' with the value 'port1'. At the bottom, there is a help icon (question mark), an 'OK' button, and a 'Cancel' button.

次の図は RequestInput の下にネストされた引数ポートを示しています：

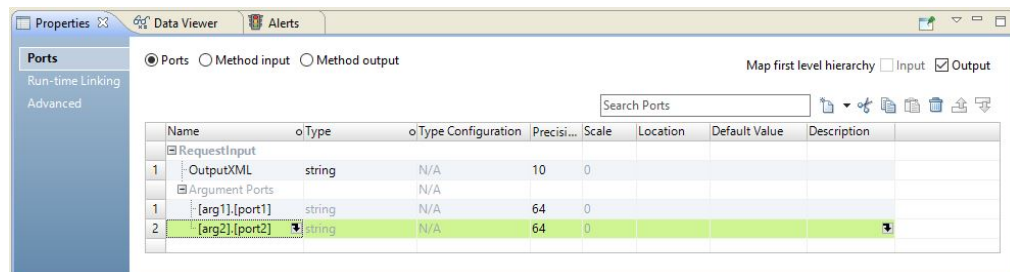


The screenshot shows the 'Data Viewer' tab in a software interface. The 'Ports' section is active, showing a tree view with 'RequestInput' expanded. Below it, a table lists the ports. The table has columns: Name, oType, oType Configuration, Precisi..., Scale, Location, Default Value, and Description.

| Name | oType | oType Configuration | Precisi... | Scale | Location | Default Value | Description |
|------------------|--------|---------------------|------------|-------|----------|---------------|-------------|
| RequestInput | | | | | | | |
| 1 OutputXML | string | N/A | 10 | 0 | | | |
| Argument Ports | | | | | | | |
| 1 [arg1].[port1] | string | N/A | 64 | 0 | | | |

7. 手順 3 と 4 を繰り返して 2 つ目の引数ポートを追加します。

2 つ目の引数ポートが 1 つ目の引数ポートの下に表示されます：



This screenshot is similar to the previous one, but it shows two ports under 'Argument Ports'. The second port, '[arg2].[port2]', is highlighted in green.

| Name | oType | oType Configuration | Precisi... | Scale | Location | Default Value | Description |
|------------------|--------|---------------------|------------|-------|----------|---------------|-------------|
| RequestInput | | | | | | | |
| 1 OutputXML | string | N/A | 10 | 0 | | | |
| Argument Ports | | | | | | | |
| 1 [arg1].[port1] | string | N/A | 64 | 0 | | | |
| 2 [arg2].[port2] | string | N/A | 64 | 0 | | | |

引数ポートの値は `http://sample.com?arg1=val1&arg2=val2` の形式で URL に付加されます。ポート名は URL では使用されません。データ統合サービスがポート名を使用して引数ポートを識別します。

8. 引数ポート `arg1` および `arg2` をベース URL のパラメータとして使用するには、アップストリームのマッピングオブジェクトから Web サービスコンシューマトランスフォーメーションの引数ポートにポートをリンクします。

アップストリームオブジェクトからリンクされたポートは URL `https://sample.com?arg1=val1&arg2=val2` の `val1` と `val2` を定義します。

第 14 章

REST および SOAP Web サービスの管理

この章では、以下の項目について説明します。

- [Web サービスの管理の概要, 183 ページ](#)
- [Web サービスのプロパティの構成, 184 ページ](#)
- [Web サービスのセキュリティ管理, 188 ページ](#)
- [グリッド上の Web サービス, 192 ページ](#)
- [Web サービスのログ, 193 ページ](#)
- [Web サービスの監視, 193 ページ](#)

Web サービスの管理の概要

REST または SOAP Web サービスは、Administrator ツールで管理できます。Web サービスセキュリティを設定し、Web サービスを設定して、Web サービスログを表示し、Web サービス要求を監視できます。これらのタスクを実行するには、適切な特権が必要です。

Web サービスをデータ統合サービスにデプロイした後に実行できるタスクは、以下のとおりです。

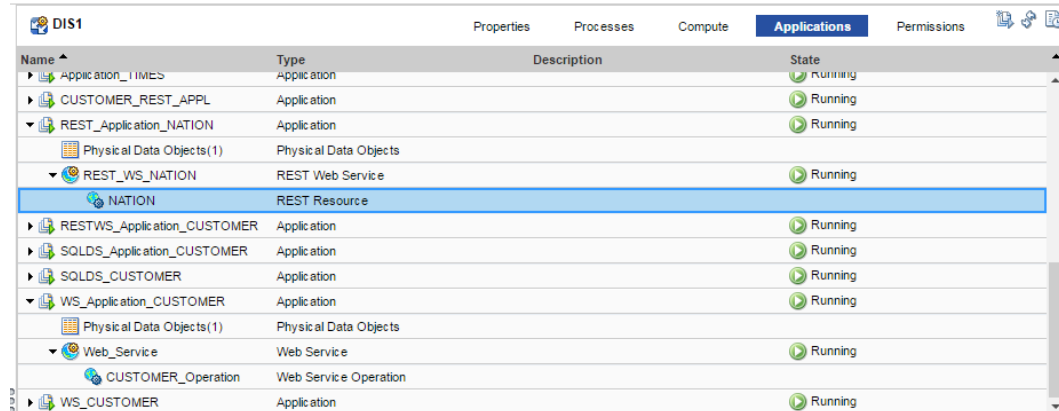
- Web サービスのセキュリティの設定。Web サービスのセキュリティを有効化し、Web サービス上での権限を割り当てます。
- Web サービスのプロパティと Web サービス操作のプロパティを設定します。
- グリッド上で Web サービスジョブを実行するように、データ統合サービスを設定します。
- Web サービスのログの確認。Web サービスに関するデータ統合サービスのログを確認します。Web サービスのランタイムログディレクトリにある Web サービスランタイムログを確認します。
- Web サービスの監視。Administrator ツールまたは Monitoring ツールを使用して、Web サービス要求を監視します。

Web サービスのプロパティの構成

Administrator ツールで、データ統合サービスにデプロイする各 Web サービスの Web サービスプロパティを設定できます。

Administrator ツールのデータ統合サービスの【アプリケーション】ビューから Web サービスのプロパティを編集します。アプリケーション名を展開し、Web サービスまたは REST Web サービスを選択します。プロパティビューにプロパティが表示されます。

次の図は、データ統合サービスの【アプリケーション】ビューを示しています。



| Name | Type | Description | State |
|-----------------------------|-----------------------|-------------|---------|
| Application_TIMES | Application | | Running |
| CUSTOMER_REST_APPL | Application | | Running |
| REST_Application_NATION | Application | | Running |
| Physical Data Objects(1) | Physical Data Objects | | |
| REST_WS_NATION | REST Web Service | | Running |
| NATION | REST Resource | | |
| RESTWS_Application_CUSTOMER | Application | | Running |
| SQLDS_Application_CUSTOMER | Application | | Running |
| SQLDS_CUSTOMER | Application | | Running |
| WS_Application_CUSTOMER | Application | | Running |
| Physical Data Objects(1) | Physical Data Objects | | |
| Web_Service | Web Service | | Running |
| CUSTOMER_Operation | Web Service Operation | | |
| WS_CUSTOMER | Application | | Running |

SOAP Web サービス操作または REST Web サービスリソースのプロパティを編集することもできます。プロパティを表示する操作名またはリソース名を選択します。

Web サービスのプロパティ

REST Web サービスおよび SOAP Web サービスのプロパティには、読み取り専用の全般プロパティと、データ統合サービスが Web サービスを実行する際に使用するプロパティがあります。

アプリケーションビューの上部パネルで Web サービスまたは REST Web サービスを展開すると、Web サービス操作または Web サービスに含まれるリソースにアクセスできます。

アプリケーションビューには、Web サービス、Web サービス操作、または Web サービスリソースの読み取り専用の全般プロパティが表示されます。ビューに表示されるプロパティはオブジェクトのタイプによって異なります。

以下の表に、Web サービス、Web サービス操作、またはリソースのタイプごとに読み取り専用の全般プロパティを示します。

| プロパティ | 説明 |
|-------|--|
| 名前 | 選択したオブジェクトの名前。すべてのオブジェクトについて表示されます。 |
| 説明 | 選択したオブジェクトの簡単な説明。すべてのオブジェクトについて表示されます。 |
| タイプ | 選択したオブジェクトのタイプ。すべてのタイプのオブジェクトについて表示されます。 |
| 場所 | 選択したオブジェクトの場所。これには、ドメインとデータ統合サービスの名前が含まれます。すべてのオブジェクトについて表示されます。 |
| URL | Web サービスに接続するために使用される URL。Web サービスについて表示されます。 |

以下の表に、Web サービスの設定可能な Web サービスプロパティを示します。

| プロパティ | 説明 |
|---------------------------------------|---|
| スタートアップのタイプ | アプリケーションの起動時や Web サービスの開始時に Web サービスが実行可能になるかどうかを示します。 |
| トレースレベル | Web サービスのランタイムログに書き込まれるエラーメッセージのレベル。以下のいずれかのメッセージレベルを選択します。 <ul style="list-style-type: none"> - OFF。DTM プロセスによって、Web サービスのランタイムログにメッセージは書き込まれません。 - SEVERE。SEVERE メッセージには、Web サービスの実行が停止する可能性があるエラーが含まれます。 - WARNING。WARNING メッセージには、修復可能な障害や警告が含まれます。DTM プロセスによって、Web サービスのランタイムログに WARNING および SEVERE メッセージが書き込まれます。 - INFO。INFO メッセージには、Web サービスのステータスメッセージが含まれます。DTM プロセスによって、Web サービスのランタイムログに INFO、WARNING、および SEVERE メッセージが書き込まれます。 - FINE。FINE メッセージには、Web サービス要求のデータ処理エラーが含まれます。DTM プロセスによって、Web サービスのランタイムログに FINE、INFO、WARNING、および SEVERE メッセージが書き込まれます。 - FINEST。FINEST メッセージはデバッグに使用されます。DTM プロセスによって、Web サービスのランタイムログに FINEST、FINE、INFO、WARNING、および SEVERE メッセージが書き込まれます。 - ALL。DTM プロセスによって、Web サービスのランタイムログに FINEST、FINE、INFO、WARNING、および SEVERE メッセージが書き込まれます。 デフォルトは [INFO] です。 |
| 要求タイムアウト | データ統合サービスでの操作マッピングの実行時に Web サービス要求がタイムアウトするまでの最大時間（ミリ秒）。デフォルトは 3,600,000 です。 |
| 最大同時要求数 | Web サービスで同時に処理できる要求の最大数。デフォルトは 10 です。 |
| ソート順序 | データ統合サービスを Unicode モードで実行している場合にデータのソートと比較に使用するソート順序。 |
| Transport Layer Security (TLS) を有効にする | Web サービスで HTTPS を使用する必要があることを示します。HTTPS を使用するようにデータ統合サービスが設定されていない場合、Web サービスは開始されません。 |

以下の表に、REST Web サービス固有のプロパティを示します。

| プロパティ | 説明 |
|-------|--|
| 認証が必要 | REST Web サービスに対する基本認証を有効にします。基本認証では、Web サービス要求のユーザー名とパスワードが必要です。デフォルトでは無効になっています。 |
| 入力精度 | 要求メッセージにおいてデータ統合サービスが解析する最大文字数。要求メッセージが入力精度を超えると、Web サービス要求は失敗します。デフォルトは 10,000 です。 |
| 出力精度 | 応答メッセージに対してデータ統合サービスが生成する最大文字数。応答メッセージが出力精度を超えると、データ統合サービスは応答メッセージを切り詰めます。デフォルトは 3,000 です。 |

以下の表に、SOAP Web サービス固有のプロパティを示します。

| プロパティ | 説明 |
|-----------------------|---|
| WS-Security を有効にする | データ統合サービスでユーザーの資格情報を検証し、ユーザーに各 Web サービス操作を実行する権限があることを確認できるようにします。SOAP Web サービスのみ。 |
| 最適化レベル | Data Integration Service がオブジェクトに適用する最適化レベルです。設定する最適化レベルに関連する数値を入力します。以下のいずれかの数値を入力できます。 <ul style="list-style-type: none">- 0. データ統合サービスは最適化を適用しません。- 1. データ統合サービスは初期プロジェクション最適化方式を適用します。- 2. データ統合サービスは、初期プロジェクション、初期選択、プッシュイン、および述部の各最適化方式を適用します。- 3. データ統合サービスは、コストベース、初期プロジェクション、初期選択、プッシュイン、述部、準結合の各最適化方式を適用します。 |
| DTM キープア ライブ時間 | DTM インスタンスが最後の要求の完了後にオープン状態を維持する期間（ミリ秒）。同じ操作に対して送信された Web サービス要求は、オープンインスタンスを再利用できます。要求の処理に必要な時間が、DTM インスタンスの初期化時間よりも短い場合は、キープアライブ時間を使用してパフォーマンスを向上させます。要求が失敗すると、DTM インスタンスは終了します。 整数でなければなりません。負の整数値は、データ統合サービスに DTM キープアライブ時間が使用されることを示します。0 を指定した場合、データ統合サービスでは DTM インスタンスがメモリに残りません。デフォルトは-1 です。 |
| SOAP の出力 精度 | 応答メッセージに対してデータ統合サービスが生成する最大文字数。応答メッセージが SOAP の出力精度を超えると、データ統合サービスは応答メッセージを切り詰めます。デフォルトは 200,000 です。 |
| SOAP の入力 精度 | 要求メッセージにおいてデータ統合サービスが解析する最大文字数。要求メッセージが SOAP の入力精度を超えると、Web サービス要求は失敗します。デフォルトは 200,000 です。 |

Web サービスの操作とリソースのプロパティ

データ統合サービスで Web サービスの操作または Web サービスリソースを実行する際に使用する設定を設定します。

以下の表に、SOAP Web サービス操作または REST Web サービスリソースの設定可能なプロパティを示します。

| プロパティ | 説明 |
|-----------------|--|
| 結果セットキャッシュの有効期限 | 結果セットキャッシュを使用できる時間（ミリ秒）。-1 に設定した場合、キャッシュは無期限です。0 に設定した場合、結果セットキャッシュは無効になります。有効期限の変更は、既存のキャッシュには適用されません。すべてのキャッシュで同じ有効期限を使用する場合は、有効期限を変更した後、結果セットキャッシュをページします。デフォルト値は 0 です。 |

Web サービスの結果セットキャッシュ処理

結果セットキャッシュ処理によって、データ統合サービスで Web サービス要求のキャッシュされた結果を使用できるようになります。短期間で同じクエリを複数のユーザーが実行する場合、結果セットキャッシュ処理を使用して同じクエリのランタイムを減らすことができます。

結果セットキャッシュ処理を設定すると、各 Web サービス要求に関連付けられている DTM プロセスの結果が、データ統合サービスによってキャッシュされます。また、設定した有効期間の結果がキャッシュされます。キャッシュが期限切れになる前に外部クライアントが同じクエリまたは要求を行うと、キャッシュされた結果がデータ統合サービスから返されます。キャッシュが存在しないか期限を過ぎている場合、データ統合サービスは DTM インスタンスを開始して要求を処理します。

キャッシュ内のデータ量がキャッシュメモリの最大サイズを超えると、データ統合サービスによって、結果セットが暗号化キャッシュファイル（<Informatica_install_dir>/tomcat/bin/disTemp/<Service_Name>/<Node_Name>/）に格納されます。

Web サービスで WS-Security が使用される場合は、データ統合サービスでユーザーごとに Web サービスの結果セットキャッシュが格納されます。データ統合サービスでは、Web サービス要求のユーザー名トークンに指定されているユーザー名ごとにキャッシュが格納されます。データ統合サービスでユーザーごとに結果がキャッシュされる場合は、Web サービス要求を送信したユーザーだけにキャッシュされた結果が返されます。

Administrator ツールで結果セットキャッシュ処理を設定するには、以下の手順を実行します。

1. データ統合サービスプロセスのプロパティで結果セットキャッシュのプロパティを設定します。

以下の表に、結果セットキャッシュのプロパティを示します。

| プロパティ | 説明 |
|------------------|---|
| 最大合計ディスクサイズ | 結果セットキャッシュのストレージ全体で利用できる最大サイズ（バイト）。デフォルトはゼロです。 |
| キャッシュメモリごとの最大サイズ | メモリ内の単一の結果セットキャッシュインスタンスに割り当てる最大サイズ（バイト）。デフォルトはゼロです。 |
| 最大合計メモリサイズ | メモリ内の結果セットキャッシュのストレージ全体に割り当てられた最大サイズ（バイト）。デフォルトはゼロです。 |
| キャッシュの最大数 | データ統合サービスに利用できる結果セットキャッシュインスタンスの最大数。デフォルトはゼロです。 |

2. SOAP Web サービス操作のプロパティまたは REST Web サービスリソースのプロパティでキャッシュの有効期間を設定します。

結果セットキャッシュの有効期限は、結果セットを使用できる時間（ミリ秒）です。-1 に設定した場合、キャッシュは無期限です。ゼロに設定した場合、結果セットキャッシュは無効になります。有効期限の変更は、既存のキャッシュには適用されません。すべてのキャッシュで同じ有効期限を使用するには、有効期限を変更した後に結果セットキャッシュをページします。デフォルトはゼロです。

3. データ統合サービスでユーザーごとに結果をキャッシュするには、Web サービスのプロパティで WS-Security を有効にします。

Web サービス操作が結果セットをキャッシュするように設定されている場合に、Web サービス要求に対して結果セットキャッシュ処理を無効にするには、SOAP 要求の HTTP ヘッダーに以下の構文を追加します。

WebServiceOptions.disableResultSetCache=true

Web サービスのセキュリティ管理

HTTP クライアントフィルタ、トランスポートレイヤセキュリティ、およびメッセージレイヤセキュリティを通じて、Web サービスでのデータ転送の保護、およびデータアクセスの認証を行うことができます。メッセージレイヤセキュリティを設定すると、Data Integration Service で接続に資格情報を渡せるようになります。

REST Web サービスには、次のセキュリティオプションを設定できます。

認証が必要

REST Web サービスに対する基本認証を有効にします。基本認証では、各 Web サービス要求にドメインのユーザー名とパスワードが含まれている必要があります。Administrator ツールで、データ統合サービスのプロパティを有効にします。[アプリケーション] > [ApplicationName] [REST Web サービス] > [isAuthenticationRequired] をクリックします。認証が必要な場合、REST Web サービスが応答を返すには、GET 要求ごとにユーザー名とパスワードが必要です。デフォルトでは無効になっています。

SOAP Web サービスには、次のセキュリティオプションを設定できます。

HTTP クライアントフィルタ

Data Integration Service が Web サービスクライアントのホスト名または IP アドレスに基づいて要求を受け入れるようにする場合は、Administrator ツールを使用して HTTP クライアントフィルタを設定します。デフォルトでは、任意のマシンで実行されている Web サービスクライアントから要求を送信できません。

メッセージレイヤセキュリティ

Data Integration Service で SOAP リクエスト内のユーザー資格情報を認証する場合は、Administrator ツールを使用して WS-Security を有効にすると共に、Web サービスの権限を設定します。Data Integration Service では、SOAP リクエストでユーザー名トークンとして提供されるユーザー資格情報を検証できます。ユーザー名トークンが有効でない場合は、Data Integration Service で要求が拒否され、Web サービスクライアントにシステム定義フォールトが送信されます。ユーザーに Web サービス操作を実行する権限がない場合は、Data Integration Service で要求が拒否され、Web サービスクライアントにシステム定義フォールトが送信されます。

トランスポートレイヤセキュリティ (TLS)

Web サービスと Web サービスクライアントの通信に HTTPS URL を使用する場合は、Administrator ツールを使用して Web サービスに対して TLS を有効にします。Web サービスが実行されている Data Integration Service でも HTTPS プロトコルを使用する必要があります。HTTPS URL では、Web サービスと Web サービスクライアント間のデータ転送に使用する接続が SSL で保護されます。

パススルーセキュリティ

操作マッピングに接続資格情報が必要である場合は、Data Integration Service で接続に対する SOAP リクエスト内のユーザー名トークンから資格情報を渡すことができます。接続に資格情報を渡すように Data Integration Service を設定するには、Administrator ツールを使用して、Data Integration Service が接続にパススルーセキュリティを使用するように構成すると共に、Web サービスに対して WS-Security を有効にします。

注: ユーザー名トークンにハッシュパスワードまたはダイジェストパスワードが含まれている場合は、パススルーセキュリティを使用できません。

Web サービスの権限

Web サービスがユーザー認証を必要とする場合は、権限によって REST Web サービスまたは SOAP Web サービスに対するユーザーのアクセスレベルが制御されます。REST Web サービスまたは SOAP Web サービスに対

する権限を設定するには、Administrator ツールを使用します。リソースまたは操作に対する権限も設定できます。

権限を割り当てるには、Web サービス、リソース、または操作をデータ統合サービスの **【アプリケーション】** ビューから選択します。**【ユーザー権限】** または **【グループ権限】** をクリックします。

管理者は Web サービス権限を次のタイプのユーザーおよびグループに割り当てます。

- Web サービスコンシューマ。Web サービスに要求を送信して Web サービスから応答を受信するネイティブドメインユーザー。ユーザーは Web サービスに対する実行権限を持っている必要があります。
- Web サービス管理者。Administrator ツールにログインし、Web サービスのプロパティを編集し、他のユーザーに権限を付与できるユーザー。
- Web サービスオペレータ。Administrator ツールにログインし、Web サービスを監視し、Web サービスの開始または停止を行うことができるユーザー。

以下の権限をユーザーおよびグループに割り当てることができます。

- 付与権限: Administrator ツールまたは *infacmd* コマンドラインプログラムを使用して、Web サービスオブジェクトに対する権限を管理できます。
- 実行権限: Web サービス要求を送信したり Web サービス応答を受信したりできます。

以下の表に、各 SOAP Web サービスオブジェクトの権限を示します。

| オブジェクト | 権限の付与 | 実行権限 |
|-----------------|---|--|
| SOAP Web サービス | Web サービスおよび Web サービス内のすべての Web サービス操作に対する権限を付与および取り消すことができます。 | Web サービス内のすべての Web サービス操作について、Web サービス要求を送信したり Web サービス応答を受信したりできます。 |
| SOAP Web サービス操作 | Web サービス操作に対する権限を付与、取り消し、および拒否することができます。 | Web サービス操作について、Web サービス要求を送信したり Web サービス応答を受信したりできます。 |

以下の表に、各 REST Web サービスオブジェクトの権限を示します。

| オブジェクト | 権限の付与 | 実行権限 |
|---------------|--|---|
| REST Web サービス | REST Web サービスおよび Web サービス内のすべての Web サービスリソースに対する権限を付与および取り消すことができます。 | REST Web サービス内のすべての Web サービスリソースについて、Web サービス要求を送信したり Web サービス応答を受信したりできます。 |
| REST リソース | REST Web サービスリソースの付与、取り消し、および拒否を行うことができます。 | REST Web サービスリソースについて、Web サービス要求を送信したり Web サービス応答を受信したりできます。 |

SOAP リクエスト内のユーザー名トークン

Web サービスがユーザー認証を必要とする場合、Web サービスクライアントでは SOAP リクエストにユーザー名トークンヘッダーを含める必要があります。Web サービスがユーザー認証を必要としない場合は、Data Integration Service で SOAP リクエスト内のユーザー名トークンヘッダーが無視されます。

SOAP リクエスト内のユーザー名トークン要素には、以下のパスワードタイプのいずれかを指定することができます。

- プレーンテキスト
- ハッシュ
- ダイジェスト

注: ユーザー名トークンにハッシュパスワードまたはダイジェストパスワードが含まれている場合は、LDAP 認証を使用できません。

ユーザーパスワードは、UsernameToken 要素の Password 要素に組み込みます。Password 要素には、使用されるパスワードセキュリティのタイプを示す Type 属性が指定されます。

プレーンテキストパスワード

ユーザーパスワードを暗号化する必要がない場合は、SOAP リクエストのユーザー名トークンヘッダーにプレーンテキストパスワードを組み込みます。Data Integration Service では、UsernameToken 要素内のプレーンテキストパスワードを処理できます。

パスワードがプレーンテキストである場合は、UsernameToken 要素に以下の子要素が含まれます。

Username 要素

ネイティブセキュリティドメインまたは任意の LDAP セキュリティドメインのユーザー名を含んでいます。デフォルトのセキュリティドメインは、ネイティブセキュリティドメインです。ユーザー名がネイティブセキュリティドメインに属する場合、Username 要素ではセキュリティドメインの名前を必要としません。ユーザー名が LDAP セキュリティドメインに属する場合は、ユーザー名の前にセキュリティドメインの名前とスラッシュ (/) を付ける必要があります。

Password 要素

プレーンテキストのパスワードを含んでいます。Password 要素の Type 属性は「PasswordText」に設定します。

以下のサンプル SOAP ヘッダーは、プレーンテキストパスワードを含んだ UsernameToken 要素の例です。

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">Administrator</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

ハッシュパスワード

ユーザーパスワードを暗号化する場合がある場合は、SOAP リクエストのユーザー名トークンヘッダーにハッシュパスワードを組み込みます。Data Integration Service では、UsernameToken 要素内のハッシュパスワードを処理できます。

ハッシュパスワードを使用する場合は、UsernameToken 要素に以下の子要素が含まれます。

Username 要素

ネイティブセキュリティドメインのユーザー名を含んでいます。

Password 要素

ハッシュパスワードを含んでいます。このパスワードは、MD5 ハッシュ関数または SHA-1 ハッシュ関数でハッシュ化し、Base64 にエンコードする必要があります。Password 要素の Type 属性は「PasswordText」に設定します。

以下のサンプル SOAP ヘッダーは、ハッシュパスワードを含んだ UsernameToken 要素の例です。

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">Ntm58Cxf7SB0QAz30lsTq1nv-D7</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

ダイジェストパスワード

ユーザーパスワードが、ナンス値とタイムスタンプでハッシュ化される暗号化パスワードである場合は、SOAP リクエストのユーザー名トークンヘッダーにダイジェストパスワードを組み込みます。Data Integration Service では、UsernameToken 要素内のダイジェストパスワードを処理できます。

ダイジェストパスワードを使用する場合は、UsernameToken 要素に以下の子要素が含まれます。

Username 要素

ネイティブセキュリティドメインのユーザー名を含んでいます。

Password 要素

ダイジェストパスワードを含んでいます。このパスワードは、Nonce 要素のナンス値および Created 要素のタイムスタンプと連結されているパスワードをハッシュ化して生成された値です。このパスワードは、SHA-1 ハッシュ関数でハッシュ化し、Base64 にエンコードする必要があります。ダイジェストパスワードのセキュリティを得るために、Password 要素の Type 属性を「PasswordDigest」に設定します。

Nonce 要素

1 回しか使用できないランダムな値であるナンス値を含んでいます。デフォルトでは、この値は、要求が作成された時刻（Created 要素の値で示されます）から 300 秒間有効です。クライアントアプリケーションは、ナンス値が有効である時間内に要求を送信する必要があります。例えば、要求が午前 10:00 に作成されたことを Created 値が示している場合、その要求は午前 10:00 から午前 10:05 まで有効です。クライアントアプリケーションが午前 10:00 より前または午前 10:05 より後に Web サービスに要求を送信した場合、要求とナンス値は無効であり、要求は失敗します。

Created 要素

要求が作成された時刻を示すタイムスタンプ値を含んでいます。タイムスタンプでは UTC 形式、yyyy-MM-dd'T'HH:mm:ss.SSS'Z'を使用します。例: 2008-08-11T18:06:32.425Z

ダイジェストパスワードでは、標準の OASIS パスワードダイジェストアルゴリズムが使用されます。

Password_Digest = Base64 (SHA-1 (nonce + created + password))

ナンス値、タイムスタンプ、およびダイジェストパスワードは、任意のツールを使用して生成できます。

以下のサンプル SOAP ヘッダーは、ダイジェストパスワードを含んだ UsernameToken 要素の例です。

```
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

```

    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordDigest">Ntm58Cxf7SB0QAz30lsTq1nv-D7</wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary">zWELHdoAzNjQQ9xzLIwFZA==</wsse:Nonce>
      <wsu:Created>2010-10-15T20:56:18.633Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>

```

グリッド上の Web サービス

データ統合サービスは、グリッド上で Web サービスを実行できます。

データ統合サービスグリッドで Web サービス要求を実行すると、このサービスにより、データ統合サービスプロセスでジョブが実行されます。グリッドのすべてのノードに、サービスロールと計算ロールの両方が必要です。

グリッド上で実行するデータ統合サービスを有効にする場合、サービスロールのあるグリッド上の各ノードで 1 つのサービスプロセスが開始されます。データ統合サービスは、1 つのサービスプロセスをマスタサービスプロセスとして指定し、残りのサービスプロセスをワーカーサービスプロセスとして指定します。ワーカーサービスプロセスが開始されると、自身をマスタサービスプロセスとして登録し、マスタがワーカーを認識できるようにします。

ワーカーサービスプロセスにより、Web サービスジョブが実行されます。マスタサービスプロセスは、ワーカーサービスプロセスとしても機能し、ジョブを完了することができます。

1 つの Web サービス要求をサブミットすると、データ統合サービスは外部 HTTP ロードバランサを使用して、サービス要求をワーカーサービスプロセスに分配します。複数の Web サービスに複数の要求をサブミットすると、データ統合サービスによって、各クエリがラウンドロビン方式でワーカーサービスプロセスにディスパッチされます。

サービスプロセスでジョブを実行するグリッドの設定

データ統合サービスグリッドで Web サービス要求を実行するとき、管理者はサービスプロセスでジョブを実行するようにデータ統合サービスを設定します。管理者は、データ統合サービスを Administrator ツールで設定します。

管理者は、Web サービスジョブのグリッドを作成し、そのグリッドにデータ統合サービスを割り当て、サービスプロセスでジョブを実行するようにデータ統合サービスを設定して、負荷分散を設定します。管理者はまた、各ノードのデータ統合プロセスのプロパティと、実行 Data Transformation マネージャ (DTM) の計算プロパティを設定することができます。

Web サービスのログ

Web サービスのログは、Data Integration Service のログ、および Web サービスのランタイムログで確認できます。

Data Integration Service のログには、Web サービスの実行を阻止する Web サービスのステータス変更やエラーなどの Web サービスメッセージが記録されている場合があります。Data Integration Service のログは、Administrator ツールの **【ログ】** タブで確認できます。

Data Integration Service の DTM プロセスで Web サービス操作マッピングが実行されると、Web サービス操作マッピングに関するメッセージが Web サービスのランタイムログに記録されます。Data Integration Service の DTM プロセスでは、Web サービス要求ごとにログファイルが作成されます。Web サービスのランタイムログは、Data Integration Service プロセスログの場所にある ws ディレクトリで確認できます。デフォルトでは、次のディレクトリで Web サービスのランタイムログを確認できます。

```
<InformaticaInstallationDir>/tomcat/bin/disLogs/ws
```

Web サービスのトレースレベル

Administrator ツールで、DTM プロセスが Web サービスのランタイムログに書き込むメッセージのトレースレベルを示すように、**【トレースレベル】** プロパティを設定します。Web サービス要求の HTTP ヘッダーを設定して、Web サービスに対して設定されているトレースレベルをオーバーライドすることができます。

各 Web サービス要求では、Web サービストレースレベルの設定に基づいて、ランタイムログが生成されます。Web サービス要求に対して Web サービストレースレベルをオーバーライドするには、Web サービス SOAP リクエストの HTTP ヘッダーに項目 `WebServiceOptions.traceLevel= <trace_level>` を追加します。例えば、HTTP ヘッダーを使用してトレースレベルを FINE に設定するには、次のテキストを使用します。

```
WebServiceOptions.traceLevel= fine
```

Web サービスの監視

REST Web サービスまたは SOAP Web サービスを監視するには、そのプロパティ、ランタイム統計、ランタイムレポートおよび各 Web サービス要求に関する情報を表示します。

Web サービスは、以下の場所で監視することができます。

- Monitoring ツール。Developer tool で、**進行状況ビュー**の **【メニュー】** ボタンをクリックし、**【ジョブの監視】** を選択します。Web サービスを実行しているデータ統合サービスを選択し、**【OK】** をクリックします。Monitoring ツールが開きます。
- Administrator ツール。Administrator ツールで Web サービスを監視するには、**【モニタ】** タブをクリックします。

特定の Web サービスを監視しているときは、その Web サービスのサマリ統計と実行統計を表示できます。

【サマリ統計】 ビューには、Web サービスの配布と状態に関するグラフィカルな情報が表示されます。時間範囲を選択し、**【要求と接続】** パネルを展開すると、**【サマリ統計】** にその時間範囲の Web サービス要求が表示されます。要求の詳細または要求のサマリを表示できます。配布状況の図またはサマリの表を表示します。**【サマリ統計】** ビューには、モデルリポジトリに保存されているデータを使用して統計が表示されます。**【サマリ統計】** を表示する前に、監視設定でモデルリポジトリを設定する必要があります。

【実行統計】 ビューには、アプリケーションにデプロイされている特定の Web サービスに関する情報が表示されます。SOAP Web サービスまたは REST Web サービスに関する以下の統計を表示できます。

- 要求の総数。

- 完了した要求。
- 強制終了された要求。データ統合サービスの再起動時、または強制終了モードで無効化されたときに、強制終了された要求。
- 失敗した要求。

Web サービスのプロパティを表示するには、ナビゲータでアプリケーションを展開し、**【Web サービス】** フォルダまたは **【Rest サービス】** フォルダを選択します。コンテンツパネルに、Web サービスのリストが表示されます。コンテンツパネルには、各 Web サービスについて、名前、説明、状態などのプロパティが表示されます。

特定の Web サービスを選択すると、詳細パネルに、次のビューが表示されます。

- プロパティビュー。プロパティビューには、Web サービスの全般的なプロパティとランタイム統計が表示されます。
- レポートビュー。レポートビューには、選択した Web サービスに関する監視レポートが表示されます。
- 操作ビュー。操作ビューには、Web サービスに含まれている各操作の名前と説明が表示されます。また、各操作のプロパティ、要求、およびレポートも表示されます。
- [要求] ビュー。要求ビューには、それぞれの Web サービス要求について、要求 ID、ユーザー名、状態、開始時刻、経過時間、終了時刻などのプロパティが表示されます。要求のリストはフィルタリングすることができます。

Web サービスの [プロパティ] ビュー

【プロパティ】 ビューには、Web サービスの全般的なプロパティとランタイム統計が表示されます。

【プロパティ】 ビューのコンテンツパネルで Web サービスを選択すると、全般的なプロパティおよび監視統計が表示されます。

Web サービスの全般的なプロパティ

名前やオブジェクトのタイプなどの、Web サービスの全般的なプロパティを表示できます。

Web サービスの統計

特定の期間における Web サービス要求のランタイム統計を表示できます。**【統計】** セクションには、完了した Web サービス要求の数、失敗した Web サービス要求の数、および Web サービス要求の総数が表示されます。

Web サービスの [レポート] ビュー

【レポート】 ビューには、選択した Web サービスに関する監視レポートが表示されます。

Web サービスを監視しているときは、**【レポート】** ビューに、その Web サービスに関するレポートが表示されます。例えば、**【最もアクティブな WebService クライアント IP】** レポートを表示すると、特定の期間に最も多くの Web サービス要求を受け取った IP アドレスを確認することができます。

REST または SOAP Web サービスの [操作] ビュー

【操作】 ビューには、Web サービスに含まれる各操作またはリソースの名前と説明が表示されます。また、各操作のプロパティ、要求、およびレポートも表示されます。

【コンテンツ】 パネルで Web サービスを選択すると、**【詳細】** パネルに **【プロパティ】** ビュー、**【要求】** ビュー、および **【レポート】** ビューが表示されます。

【プロパティ】ビュー

【プロパティ】ビューには、選択した Web サービス操作またはリソースの全般プロパティと統計が表示されます。全般プロパティには、操作またはリソースの名前とオブジェクトのタイプが含まれます。また、このビューには、特定の期間の Web サービス操作に関する統計も表示されます。統計には、Web サービス要求が完了した数、失敗した数、およびその合計が含まれます。

【要求】ビュー

【要求】ビューには、それぞれの Web サービス操作について、要求 ID、ユーザー名、状態、開始時刻、経過時間、終了時刻などのプロパティが表示されます。要求のリストはフィルタリングすることができます。また、選択した Web サービス要求のログを表示することもできます。

SOAP Web サービスの【レポート】ビュー

【レポート】ビューには、SOAP Web サービス操作に関するレポートが表示されます。

Web サービスの【要求】ビュー

【要求】ビューには、それぞれの Web サービス要求について、要求 ID、ユーザー名、状態、開始時刻、経過時間、終了時刻などのプロパティが表示されます。要求のリストはフィルタリングすることができます。

【コンテンツ】パネルで Web サービス要求を選択すると、その要求に関するログが【詳細】パネルに表示されます。【詳細】パネルには、選択した Web サービス要求の全般プロパティと統計が表示されます。統計には、Web サービス要求が完了した数、失敗した数、およびその合計が含まれます。

【要求】ビューから Web サービス要求を強制終了することもできます。Web サービス要求を強制終了するには、ワークフロー要求を選択し、【コンテンツ】パネルで【アクション】 > 【選択した要求の強制終了】をクリックします。

付録 A

データ型の互換性

この付録では、以下の項目について説明します。

- [データ型リファレンスの概要, 196 ページ](#)
- [XML データ型とトランスフォーメーションデータ型, 197 ページ](#)
- [Decimal, 199 ページ](#)

データ型リファレンスの概要

マッピングを作成する場合は、ソースからのデータ読み込み、変換、ターゲットへの書き込みを Data Integration Service に実行させるための指示セットを作成します。データ統合サービスは、マッピング内の最初のトランスフォーメーションから始まるマッピング内のデータフロー、およびマッピング内の各ポートに割り当てられているデータ型に基づいて、データを変換します。

Developer tool には、次の 2 種類のデータ型が表示されます。

ネイティブデータ型

ネイティブデータ型は、物理データオブジェクトとして使用されるリレーショナルテーブルまたはフラットファイルに固有です。ネイティブデータ型は、物理データオブジェクトカラムのプロパティに表示されます。

トランスフォーメーションデータ型

トランスフォーメーションデータ型は、トランスフォーメーションに表示されるデータ型のセットです。ANSI SQL-92 汎用データ型に基づく内部データ型で、プラットフォーム間でデータを移動するときデータ統合サービスによって使用されます。トランスフォーメーションデータ型は、マッピング内のすべてのトランスフォーメーションに表示されます。

トランスフォーメーションデータ型には、次のデータ型があります。

- プリミティブデータ型。単一のカラム位置にある 1 つのデータ値を表します。
- 複合データ型。単一のカラム位置にある複数のデータ値を表します。Spark エンジンで実行されるマッピングで複合データ型を使用して、複合ファイルの階層データを処理します。

データ統合サービスは、ソースデータを読み取るときに、ネイティブデータ型に対応するトランスフォーメーションデータ型に変換してから、データのトランスフォームを実行します。データ統合サービスは、ターゲットに書き込むときに、トランスフォーメーションデータ型に対応するネイティブデータ型に変換します。

マルチバイト文字セットを指定した場合、データ型は 3 バイトまで文字を格納する追加スペースをデータベース内に割り当てます。

XML データ型とトランスフォーメーションデータ型

XML のデータ型は、データ統合サービスがデータのプラットフォーム間の移動に使用するトランスフォーメーションデータ型にマッピングされます。

データ統合サービスは、W3C が 2001 年 5 月 2 日の勧告で指定した XML データ型をすべてサポートしています。しかし、データ統合サービスは XML の値の範囲全体には対応していない場合があります。XML データタイプの詳細については、次のサイトで XML データタイプの W3C 仕様を参照してください:

<http://www.w3.org/TR/xmlschema-2>

以下の表に、XML データ型とトランスフォーメーションデータ型の比較を示します。

| データ型 | トランスフォーメーション | 範囲 |
|--------------|--------------|--|
| anyURI | String | 1~104,857,600 文字 |
| base64Binary | バイナリ | 1~104,857,600 バイト |
| boolean | String | 1~104,857,600 文字 |
| byte | Integer | -2,147,483,648~2,147,483,647 |
| 日付 | 日付/時刻 | 西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで) |
| dateTime | 日付/時刻 | 西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで) |
| decimal | Decimal | 精度 1~28、位取り 0~28 |
| double | Double | 15 桁精度 |
| duration | String | 1~104,857,600 文字 |
| ENTITIES | String | 1~104,857,600 文字 |
| ENTITY | String | 1~104,857,600 文字 |
| float | Double | 15 桁精度 |
| gDay | String | 1~104,857,600 文字 |
| gMonth | String | 1~104,857,600 文字 |
| gMonthDay | String | 1~104,857,600 文字 |
| gYear | String | 1~104,857,600 文字 |
| gYearMonth | String | 1~104,857,600 文字 |
| hexBinary | バイナリ | 1~104,857,600 バイト |
| ID | String | 1~104,857,600 文字 |
| IDREF | String | 1~104,857,600 文字 |

| データ型 | トランスフォーメーション | 範囲 |
|--------------------|--------------|---|
| IDREFS | String | 1～104,857,600 文字 |
| 整数型 | Integer | -2,147,483,648～2,147,483,647 |
| Integer | Integer | -2,147,483,648～2,147,483,647 |
| language | String | 1～104,857,600 文字 |
| 長整数型 | Bigint | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| 名前 | String | 1～104,857,600 文字 |
| NCName | String | 1～104,857,600 文字 |
| negativeInteger | Integer | -2,147,483,648～2,147,483,647 |
| NMTOKEN | String | 1～104,857,600 文字 |
| NMTOKENS | String | 1～104,857,600 文字 |
| nonNegativeInteger | Integer | -2,147,483,648～2,147,483,647 |
| nonPositiveInteger | Integer | -2,147,483,648～2,147,483,647 |
| normalizedString | String | 1～104,857,600 文字 |
| NOTATION | String | 1～104,857,600 文字 |
| positiveInteger | Integer | -2,147,483,648～2,147,483,647 |
| QName | String | 1～104,857,600 文字 |
| short | Integer | -2,147,483,648～2,147,483,647 |
| string | String | 1～104,857,600 文字 |
| 時間 | 日付/時刻 | 西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで) |
| トークン | String | 1～104,857,600 文字 |
| unsignedByte | Integer | -2,147,483,648～2,147,483,647 |
| unsignedInt | Integer | -2,147,483,648～2,147,483,647 |
| unsignedLong | Bigint | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| unsignedShort | Integer | -2,147,483,648～2,147,483,647 |

Decimal

Web サービスの操作マッピングに、28 桁を上回る精度を持つ Decimal データ型の入力トランスフォーメーションまたは出力トランスフォーメーションが含まれる場合、データ統合サービスは Decimal データ型を Double データ型に変換します。

索引

A

all グループ
REST Web サービスコンシューマトランスフォーメーションの表示 [157](#)
anyAttribute 属性
Web サービスコンシューマトランスフォーメーション [91](#), [94](#)
anyType
ポートのマッピング [81](#)
anyType 要素
Web サービスコンシューマトランスフォーメーション [91](#), [94](#)
解析 [68](#)
any 要素
Web サービスコンシューマトランスフォーメーション [91](#), [94](#)
attributeFormDefault
スキーマオブジェクト [29](#)

C

choice 要素
REST Web サービスコンシューマトランスフォーメーションの表示 [157](#)
SOAP メッセージの解析 [71](#)
Web サービスコンシューマトランスフォーメーションの表示 [92](#), [95](#)
説明 [82](#)

E

elementFormDefault
スキーマオブジェクト [29](#)

G

gzip
SOAP メッセージの圧縮 [100](#)

H

HTTP POST
ヘッダー [47](#)
HTTP エラー出力
Web サービスコンシューマトランスフォーメーションの有効化 [97](#)
HTTP 応答コード
REST Web サービスコンシューマトランスフォーメーション [153](#)
HTTP 接続
REST Web サービス [158](#)
HTTP ヘッダ
REST Web サービスコンシューマトランスフォーメーションへの追加 [152](#)
HTTP ヘッダー
Web サービスコンシューマトランスフォーメーションへの追加 [90](#)

I

isAuthenticationRequired
REST Web サービス [188](#)

N

ノード
Web サービスフォールト処理 [59](#)
NULL 可能
スキーマオブジェクト [25](#)

Q

Qname 要素
SOAP メッセージの解析 [70](#)

R

理由
Web サービスフォールト処理 [59](#)
REST Web サービス
Web サービスの作成 [122](#)
応答メッセージ形式 [120](#)
概要 [109](#), [118](#)
カスタムマッピング [115](#)
出力トランスフォーメーション [116](#)
スキーマビュー [113](#)
データビューアビュー [121](#)
デフォルトマッピング [114](#)
複数出現データ [117](#)
プロセス [110](#)
要求メッセージ形式 [118](#)
リソースキー [113](#)
リソースキーで問い合わせ [119](#)
リソースの同期 [113](#)
リソースマッピング [114](#)
REST Web サービスコンシューマトランスフォーメーション
Delete メソッド [150](#)
GET メソッド [148](#)
HTTP ヘッダポート [152](#)
HTTP メソッド [148](#)
JSON スキーマ [176](#)
Post メソッド [149](#)
Put メソッド [149](#)
RequestInput ポート [151](#)
URL ポート [152](#), [180](#)
XML スキーマ [169](#)
XML スキーマ検証 [158](#)
インターネット媒体のタイプ [158](#)
応答コードポート [153](#)
概要 [144](#)
カスタム URL [180](#)

REST Web サービスコンシューマトランスフォーメーション (続く)

- クッキーポート [153](#)
- 再利用可能 [159](#)
- 再利用不可 [159](#)
- 作成 [159](#)
- 出力 XML ポート [153](#)
- 出力ポート [151](#)
- 出力ポートの割付 [157](#)
- 出力マッピング [156](#)
- 出力マッピングのルール [156](#)
- 出力マッピングビューのカスタマイズ [157](#)
- 詳細プロパティ [158](#)
- スキーマ [169](#), [176](#)
- セキュリティ [87](#)
- 接続プロパティ [158](#)
- 設定 [146](#)
- ソート済み入力 [158](#)
- トランスポートレイヤセキュリティ [87](#)
- トレースレベル [158](#)
- 入力ポート [151](#)
- 入力マッピング [153](#)
- 入力マッピングのルール [154](#)
- パススルーポート [151](#)
- 引数ポート [152](#), [180](#)
- プロキシサーバーのサポート [144](#)
- プロセス [146](#)
- ベース URL の設定 [158](#)
- ポート [151](#)
- ポートへの要素のマッピング [151](#)
- マッピング出力 [144](#)
- マッピング入力 [144](#)
- マッピング入力ポート [154](#)
- メッセージの設定 [146](#)
- リソースの識別 [147](#)
- 解析されたデータ [165](#)
- 解析済 [163](#)
- 未解析 [163](#)
- 未解析のデータ [163](#)

REST Web サービスの作成

- データオブジェクトから作成 [137](#)
- データオブジェクトのデプロイ [140](#)

REST および SOAP

- 比較 [13](#)

S

sequence グループ

- REST Web サービスコンシューマトランスフォーメーションの表示 [157](#)

SOAP 1.1

- フォールト要素 [59](#)

SOAP 1.2

- フォールト要素 [59](#)

SOAP Web サービス

- 開発 [17](#)
- 作成の概要 [36](#)

SOAP アクション

- Web サービスコンシューマトランスフォーメーションのオーバーライド [97](#)

SOAP アクションのオーバーライド

- Web サービスコンシューマトランスフォーメーション [97](#)

SOAP 圧縮

- Web サービスコンシューマトランスフォーメーション [100](#)

SOAP 階層

- 入力ポートへのリレーション [74](#)

SOAP メッセージ

- anyType 要素の解析 [68](#)

SOAP メッセージ (続く)

- choice 要素の解析 [71](#)
- choice 要素のマッピング [82](#)
- union 要素へのポートのマッピング [84](#)
- 概要 [86](#)
- keys [75](#)
- 代替グループの解析 [70](#)
- ピボット化するデータ [78](#)
- 複数出現ノードのマッピング [66](#)
- 複数の入力ポートをマッピング [78](#)
- ポートのマッピング [76](#)
- リストの要素の解析 [71](#)
- リストの要素のマッピング [83](#)

SOAP メッセージ解析

- Qname 要素 [70](#)
- union 要素 [71](#)
- 正規化した出力 [66](#)
- 説明 [64](#)
- 派生型 [69](#)
- 非正規化した出力 [67](#)
- ピボット化した出力 [68](#)

U

union 要素

- SOAP メッセージの解析 [71](#)
- 説明 [84](#)

UsernameToken

- パスワードダイジェスト [191](#)
- パスワードタイプ [190](#)
- ハッシュパスワード [190](#)
- プレーンテキストパスワード [190](#)

W

Web サービスコンシューマトランスフォーメーション

- HTTP エラー出力の有効化 [97](#)
- HTTP ヘッダーの追加 [90](#)
- SOAP 圧縮 [100](#)
- SOAP メッセージ [86](#)
- エラー処理 [99](#)
- エンドポイント URL [90](#)
- 概要 [85](#)
- キーの表示 [92](#), [95](#)
- クッキー認証 [90](#)
- 作成 [103](#)
- 出力マッピング [94](#)
- 詳細プロパティ [97](#)
- セキュリティ [87](#)
- 操作 [87](#)
- 同時 Web サービス要求のメッセージ [97](#)
- 動的 Web サービス URL [90](#)
- 動的な WS-Security 名 [90](#)
- トランスポートレイヤセキュリティ [87](#)
- 入力マッピング [91](#)
- 汎用フォールト出力の有効化 [97](#)
- フィルタの最適化 [102](#)
- プッシュイン最適化 [102](#)
- 汎用 SOAP フォールト [99](#)
- マッピング出力ノード [94](#)
- マッピング入力ポート [91](#)
- 最適化にプッシュインの有効化 [103](#)
- 初期選択の最適化 [101](#)

Web サービス

- anyType へのポートのマッピング [81](#)
- Output トランスフォーメーションの設定 [53](#)

Web サービス (続く)

- SOAP の例 [17](#)
- WSDL [16](#)
- WSDL URL [16](#)
- WSDL からの作成 [39](#)
- WSDL なしでの作成 [41](#)
- WSDL の関連付け [39](#)
- WSDL ビュー [39](#)
- 演算 [15](#)
- 概要 [12](#)
- 概要ビュー [37](#)
- 監視 [193](#)
- 結果セットキャッシュの設定 [186](#)
- 権限 [189](#)
- 権限のタイプ [189](#)
- コンポーネント [15](#)
- システム定義フォールト [60](#)
- 手動による作成 [41](#)
- セキュリティ [188](#)
- 操作プロパティ [186](#)
- 代替グループ [82](#)
- 定義済みフォールト [61](#)
- 定義済みフォールトの作成 [47](#)
- データ統合サービスグリッド [192](#)
- デプロイメント [17](#)
- トレースレベル [193](#)
- 入力トランスフォーメーション [50](#)
- 入力トランスフォーメーションの設定 [51](#)
- 派生型 [81](#)
- 汎用フォールト [62](#)
- フォールト処理 [59](#)
- フォールトトランスフォーメーションの設定 [57](#)
- プロセス [13](#)
- プロパティ [184](#)
- プロパティの設定 [184](#)
- ヘッダーの作成 [47](#)
- 要素の作成 [46](#)
- リソースのプロパティ [186](#)
- ログ [193](#)

Web サービスコンシューマ

- プロセス [14](#), [110](#)

Web サービス接続

- 概要 [97](#)

Web サービストランスフォーメーション

- [場所] カラム [74](#)

Web サービスのセキュリティ

- HTTPS [188](#)
- HTTP クライアントフィルタ [188](#)
- isAuthenticationRequired [188](#)
- 権限 [188](#)
- 承認 [188](#)
- トランスポートレイヤセキュリティ [188](#)
- 認証 [188](#)
- パススルーセキュリティ [188](#)
- メッセージレイヤセキュリティ [188](#)

WS-Security ユーザー名

- 動的ポート [90](#)

WSDL

- Web サービスとの関連付け [39](#)
- WSDL からの Web サービスの作成 [39](#)
- 説明 [16](#)

WSDL URL

- 説明 [16](#)

WSDL データオブジェクト

- インポート [19](#)
- 概要ビュー [20](#)
- 作成 [20](#)
- 詳細ビュー [20](#)

WSDL データオブジェクト (続く)

- スキーマビュー [19](#)
- 同期 [21](#)

WSDL ファイル

- サービス要素 [86](#)
- 操作の要素 [86](#)
- バインディング要素 [86](#)
- ポート要素 [86](#)

X

XML として出力するためにマップ

- 出力トランスフォーメーション [52](#)
- フォールトトランスフォーメーション [55](#)

XML として入力するためにマップ

- 入力トランスフォーメーション [50](#)

え

演算

- 説明 [15](#)

エンドポイント URL

- REST Web サービスコンシューマトランスフォーメーション [152](#)
- Web サービスコンシューマトランスフォーメーション [90](#)

お

応答コード

- REST Web サービスコンシューマトランスフォーメーション [153](#)

応答メッセージ

- REST Web サービス [120](#)

か

階層の第 1 レベルのマップ

- 出力トランスフォーメーション [52](#)
- 入力トランスフォーメーション [50](#)
- フォールトトランスフォーメーション [56](#)

カスタムリソースマッピング

- 説明 [115](#)

き

keys

- SOAP メッセージ階層 [75](#)

基本型のプロパティ

- スキーマオブジェクト [27](#)

基本認証

- REST Web サービス [188](#)

く

空白を省略プロパティ

- スキーマオブジェクト [27](#)

クッキー認証

- REST Web サービスコンシューマトランスフォーメーション [153](#)
- Web サービスコンシューマトランスフォーメーション [90](#)

け

結果セットキャッシュ処理
Web サービス操作のプロパティ [186](#)
設定 [186](#)

こ

コード
Web サービスフォールト処理 [59](#)
固定値プロパティ
スキーマオブジェクト [25](#)

さ

サービス
WSDL ファイル要素 [86](#)
最小出現回数
スキーマオブジェクト [25](#)
最小長
スキーマオブジェクト [25](#)
最大出現回数
スキーマオブジェクト [25](#)
最大長
スキーマオブジェクト [25](#)
作成
フォールトトランスフォーメーション [57](#)

し

システム定義フォールト
Web サービスフォールト処理 [60](#)
出力トランスフォーメーション
Web サービスでの設定 [53](#)
XML として出力するためにマップ [52](#)
階層の第 1 レベルのマップ [52](#)
[詳細設定] タブ [53](#)
説明 [52](#)
ビューのカスタマイズオプション [63](#)
[ポート] タブ [52](#)
マッピングのルールとガイドライン [53](#)
出力マッピング
REST Web サービス [116](#)
REST Web サービスコンシューマトランスフォーメーション [156](#)
Web サービスコンシューマトランスフォーメーション [94](#)
複数グループ [117](#)
種類プロパティ
スキーマオブジェクト [27](#)
[詳細設定] タブ
出力トランスフォーメーション [53](#)
詳細プロパティ
REST Web サービスコンシューマトランスフォーメーション [158](#)
Web サービスコンシューマトランスフォーメーション [97](#)
詳細要素
Web サービスフォールト処理 [59](#)
認証
UsernameToken [190](#)
証明書
証明書の管理 [22, 34](#)
証明書のプロパティ [22, 35](#)
信頼されていない証明書 [22, 34](#)
信頼されていない証明書の追加 [22, 35](#)

す

スキーマオブジェクト
attributeFormDefault [29](#)
elementFormDefault [29](#)
インポート [30](#)
概要 [23](#)
概要ビュー [23](#)
スキーマビュー [24](#)
スキーマファイル [24](#)
スキーマファイルの編集 [31](#)
属性プロパティ [29](#)
代替グループ [26](#)
単純型 [27](#)
抽象プロパティ [25](#)
デフォルトエディタとして設定 [32](#)
同期 [31](#)
名前空間 [25](#)
ファイルの場所 [29](#)
複合要素 [28](#)
ブロックプロパティ [26](#)
プロパティからの継承 [28](#)
プロパティによる継承 [28](#)
要素の詳細プロパティ [26, 28](#)
要素の詳細プロパティ [26, 28](#)
要素のプロパティ [25](#)
スキーマビュー
スキーマオブジェクト [24](#)
単純型の詳細プロパティ [27](#)
スキーマビュー
REST Web サービス [113](#)
スキーマファイル
スキーマオブジェクトからの削除 [24](#)
スキーマオブジェクトへの追加 [24](#)
デフォルトエディタとして設定 [32](#)
編集 [33](#)
すべてのグループ
Web サービスコンシューマトランスフォーメーションの表示 [92, 95](#)

せ

生成キー
Web サービス出力グループ [66](#)
生成されたプレフィックス
名前空間の変更 [24](#)
セキュリティ
UsernameToken [190](#)
Web サービスのセキュリティ [188](#)
接続
REST Web サービス [158](#)
Web サービス [97](#)

そ

操作
WSDL ファイル要素 [86](#)
操作出力
説明 [15](#)
[操作出力] 領域
Web サービスコンシューマトランスフォーメーションのカスタマイズ [95](#)
操作入力
説明 [15](#)
[操作入力] 領域
Web サービスコンシューマトランスフォーメーションのカスタマイズ [92](#)

- 操作フォール
- 説明 [15](#)
- 操作マッピング
- 概要 [48](#)
- 出力トランスフォーメーション [52](#)
- テスト [62](#)
- 入力トランスフォーメーション [50](#)
- フォールトトランスフォーメーション [55](#)
- 操作領域
- Web サービストランスフォーメーション [74](#)
- 属性プロパティ
- スキーマオブジェクト [29](#)

た

- 第 1 レベル階層のマッピング
- 説明 [116](#)
- 代替グループ
- SOAP メッセージの解析 [70](#)
- Web サービス [82](#)
- Web サービスコンシューマトランスフォーメーション [91, 94](#)
- スキーマオブジェクト [26](#)
- 単純型
- スキーマオブジェクト [27](#)
- 単純型ビュー
- スキーマオブジェクト [27](#)

ち

- 抽象プロパティ
- スキーマオブジェクト [25](#)

て

- 定義済みフォールト
- Web サービスフォールト処理 [61](#)
- 作成 [47](#)
- データアクセス方式
- REST Web サービスプレビュー [121](#)
- データ型
- XML [197](#)
- 概要 [196](#)
- データ統合サービスグリッド
- Web サービス [192](#)

と

- 同時 Web サービス要求のメッセージ
- Web サービスコンシューマトランスフォーメーションの有効化 [97](#)
- 動的 URL
- Web サービスコンシューマトランスフォーメーション [90](#)
- トランスポートレイヤセキュリティ
- REST Web サービスコンシューマトランスフォーメーション [87](#)
- Web サービスコンシューマトランスフォーメーション [87](#)
- トレースレベル
- Web サービス [193](#)

な

- 名前空間
- スキーマオブジェクト [25](#)
- 生成されたプレフィックスの変更 [24](#)

に

- 入力トランスフォーメーション
- Web サービスでの設定 [51](#)
- XML として入力するためにマップ [50](#)
- 階層の第 1 レベルのマップ [50](#)
- 説明 [50](#)
- ビューのカスタマイズオプション [63](#)
- [ポート] タブ [50](#)
- マッピングのルールとガイドライン [50](#)
- [入力ポート] 領域
- SOAP メッセージの生成 [73](#)
- 入力マッピング
- REST Web サービスコンシューマトランスフォーメーション [153](#)
- Web サービスコンシューマトランスフォーメーション [91](#)

は

- バインディング
- WSDL ファイル要素 [86](#)
- [場所] カラム
- Web サービストランスフォーメーション [74](#)
- パスワード
- ハッシュ [190](#)
- プレーンテキスト [190](#)
- 派生型
- SOAP メッセージの解析 [69](#)
- Web サービス [81](#)
- 派生型要素
- Web サービスコンシューマトランスフォーメーション [91, 94](#)
- パターンプロパティ
- スキーマオブジェクト [25](#)
- 汎用フォールト
- Web サービスフォールト処理 [62](#)
- 汎用フォールト出力
- Web サービスコンシューマトランスフォーメーションの有効化 [97](#)

ひ

- 非正規化した出力
- SOAP メッセージ解析 [67](#)
- 非正規化した入力
- Web サービスポート [79](#)
- ピボット化した出力
- SOAP メッセージ解析 [68](#)
- ピボット化したデータ
- SOAP メッセージ [78](#)
- ビューのカスタマイズオプション
- 説明 [63](#)

ふ

- フィルタポート
- Web サービスコンシューマトランスフォーメーション [102](#)
- フォールトアクタ
- Web サービスフォールト処理 [59](#)
- フォールトコード
- Web サービスフォールト処理 [59](#)
- フォールトトランスフォーメーション
- Web サービスでの設定 [57](#)
- XML として出力するためにマップ [55](#)
- 階層の第 1 レベルのマップ [56](#)
- 作成 [57](#)
- 説明 [55](#)
- 定義済みフォールト [57](#)

フォールトトランスフォーメーション (続く)

汎用フォールト [57](#)

ビューのカスタマイズオプション [63](#)

[ポート] タブ [56](#)

マッピングのルールとガイドライン [56](#)

フォールト文字列

Web サービスフォールト処理 [59](#)

フォールト要素

SOAP 1.1 [59](#)

SOAP 1.2 [59](#)

フォールトをエラーとして扱う

Web サービスコンシューマトランスフォーメーションの有効化 [97](#)

複合型

詳細プロパティ [28](#)

複合キー

REST Web サービスコンシューマトランスフォーメーション [154](#)

Web サービスコンシューマトランスフォーメーション [91](#)

プッシュイン最適化

Web サービスコンシューマトランスフォーメーション [102](#)

ブロックプロパティ

スキーマオブジェクト [26](#)

プロパティからの継承

スキーマオブジェクト [28](#)

プロパティによる継承

スキーマオブジェクト [28](#)

へ

ヘッダ

説明 [15](#)

ヘッダー

HTTP POST [47](#)

作成 [47](#)

ほ

方法

データオブジェクトからの REST Web サービスの作成 [137](#)

データオブジェクトを REST Web サービスとしてデプロイ [140](#)

ポート

SOAP メッセージへのマップ [76](#)

非正規化した Web サービストランスフォーメーション [79](#)

[ポート] タブ

出力トランスフォーメーション [52](#)

入力トランスフォーメーション [50](#)

フォールトトランスフォーメーション [56](#)

汎用 SOAP フォールト

Web サービスコンシューマトランスフォーメーション [99](#)

め

メンバタイプ

スキーマオブジェクト [27](#)

よ

要求メッセージ

REST Web サービス [118](#)

要素

共用体 [84](#)

作成 [46](#)

り

リストの要素

SOAP メッセージの解析 [71](#)

説明 [83](#)

リソースキー

REST Web サービス [113](#)

で検索 [119](#)

要求 URI で参照 [119](#)

リソース識別子

データレビューのフィルタ [121](#)

リソースの同期

REST Web サービス [113](#)

リソースマッピング

REST Web サービス [114](#)

カスタム [115](#)

[詳細設定] タブ [120](#)

説明 [109](#)

デフォルト [114](#)

リソース

REST Web サービス [109, 111](#)

れ

列挙プロパティ

スキーマオブジェクト [25](#)

ろ

ロール

Web サービスフォールト処理 [59](#)