



Informatica®  
10.1.0

# Web サービスガイド

Informatica Web サービスガイド

10.1.0

2016 年 6 月

© 著作権 Informatica LLC 1998, 2018

本ソフトウェアおよびマニュアルには、Informatica LLC の所有権下にある情報が収められています。これらは使用および開示の制限等を定めた使用許諾契約のもとに提供され、著作権法により保護されています。当該ソフトウェアのリバースエンジニアリングは禁じられています。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。このソフトウェアは、米国および/または国際的な特許、およびその他の出願中の特許によって保護されています。

合衆国政府によるソフトウェアの使用、複製または開示は、DFARS 227.7202-1 (a) および 227.7702-3 (a) (1995 年)、DFARS 252.227-7013(C) (1) (ii) (1988 年 10 月)、FAR 12.212 (a) (1995 年)、FAR 52.227-19、または FAR 52.227-14 (ALT III) に記載されているとおりに、当該ソフトウェア使用許諾契約に定められた制限によって規制されます。

本製品または本書の情報は、予告なしに変更されることがあります。お客様が本製品または本書内に問題を発見された場合は、書面に当社までお知らせください。

Informatica、Informatica Platform、Informatica Data Services、PowerCenter、PowerCenterRT、PowerCenter Connect、PowerCenter Data Analyzer、PowerExchange、PowerMart、Metadata Manager、Informatica Data Quality、Informatica Data Explorer、Informatica B2B Data Transformation、Informatica B2B Data Exchange、Informatica On Demand、Informatica Identity Resolution、Informatica Application Information Lifecycle Management、Informatica Complex Event Processing、Ultra Messaging、Informatica Master Data Management、および Live Data Map は、Informatica LLC の米国および世界中の管轄地での商標または登録商標です。その他のすべて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トの一部は、次のサードパーティが有する著作権に従います（ただし、これらに限定されません）。Copyright DataDirect Technologies. All rights reserved. Copyright (C) Sun Microsystems. All rights reserved. Copyright (C) RSA Security Inc. All rights reserved. Copyright (C) Ordinal Technology Corp. All rights reserved. Copyright (C) Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright (C) Meta Integration Technology, Inc. All rights reserved. Copyright (C) Intalio. All rights reserved. Copyright (C) Oracle. All rights reserved. Copyright (C) Adobe Systems Incorporated. All rights reserved. Copyright (C) DataArt, Inc. All rights reserved. Copyright (C) ComponentSource. All rights reserved. Copyright (C) Microsoft Corporation. All rights reserved. Copyright (C) Rogue Wave Software, Inc. All rights reserved. Copyright (C) Teradata Corporation. All rights reserved. Copyright (C) Yahoo! Inc. All rights reserved. Copyright (C) Glyph & Cog, LLC. All rights reserved. Copyright (C) Thinkmap, Inc. All rights reserved. Copyright (C) Clearpace Software Limited. All rights reserved. Copyright (C) Information Builders, Inc. All rights reserved. Copyright (C) OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright (C) International Organization for Standardization 1986. All rights reserved. Copyright (C) ej-technologies GmbH. All rights reserved. Copyright (C) Jaspersoft Corporation. All rights reserved. Copyright (C) International Business Machines Corporation. All rights reserved. Copyright (C) yWorks GmbH. All rights reserved. Copyright (C) Lucent Technologies. All rights reserved. Copyright (C) University of Toronto. All rights reserved. Copyright (C) Daniel Veillard. All rights reserved. Copyright (C) Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright (C) MicroQuill Software Publishing, Inc. All rights reserved. Copyright (C) PassMark Software Pty Ltd. All rights reserved. Copyright (C) LogiXML, Inc. All rights reserved. Copyright (C) 2003-2010 Lorenzi Davide, All rights reserved. Copyright (C) Red Hat, Inc. All rights reserved. Copyright (C) The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright (C) EMC Corporation. All rights reserved. Copyright (C) Flexera Software. All rights reserved. Copyright (C) Jinfonet Software. All rights reserved. Copyright (C) Apple Inc. All rights reserved. Copyright (C) Telerik Inc. All rights reserved. Copyright (C) BEA Systems. All rights reserved. Copyright (C) PDFlib GmbH. All rights reserved. Copyright (C) Orientation in Objects GmbH. All rights reserved. Copyright (C) Tanuki Software, Ltd. All rights reserved. Copyright (C) Ricebridge. All rights reserved. Copyright (C) Sencha, Inc. All rights reserved. Copyright (C) Scalable Systems, Inc. All rights reserved. Copyright (C) jQWidgets. All rights reserved. Copyright (C) Tableau Software, Inc. All rights reserved. Copyright (C) MaxMind, Inc. All rights reserved. Copyright (C) TMate Software s.r.o. All rights reserved. Copyright (C) MapR Technologies Inc. All rights reserved. Copyright (C) Amazon Corporate LLC. All rights reserved. Copyright (C) Highsoft. All rights reserved. Copyright (C) Python Software Foundation. All rights reserved. Copyright (C) BeOpen.com. All rights reserved. Copyright (C) CNRI. All rights reserved.

本製品には、Apache Software Foundation (<http://www.apache.org/>) によって開発されたソフトウェア、およびさまざまなバージョンの Apache License（まとめて「License」と呼んでいます）の下に許諾された他のソフトウェアが含まれます。これらのライセンスのコピーは、<http://www.apache.org/licenses/> で入手できます。適用法にて要求されないか書面に合意されない限り、ライセンスの下に配布されるソフトウェアは「現状のまま」で配布され、明示的あるいは黙示的かを問わず、いかなる種類の保証や条件も付帯することはありません。ライセンス下での許諾および制限を定める具体的文言については、ライセンスを参照してください。

本製品には、Mozilla (<http://www.mozilla.org/>) によって開発されたソフトウェア、ソフトウェア copyright The JBoss Group, LLC、コンテンツの無断複製・転載を禁じます、ソフトウェア copyright, Red Hat Middleware, LLC、コンテンツの無断複製・転載を禁じます、Copyright (C) 1999-2006 by Bruno Lowagie and Paulo Soares および GNU Lesser General Public License Agreement (<http://www.gnu.org/licenses/lgpl.html> を参照) に基づいて許諾されたその他のソフトウェアが含まれています。資料は、Informatica が無料で提供しており、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は市場性および特定の目的の適合性の黙示の保証などを含めて、一切の明示的及び黙示的保証の責任を負いません。

製品には、ワシントン大学、カリフォルニア大学アーバイン校、およびバンダービルト大学の Douglas C. Schmidt および同氏のリサーチグループが著作権を持つ ACE (TM) および TAO (TM) ソフトウェアが含まれています。Copyright (C) 1993-2006, All rights reserved.

本製品には、OpenSSL Toolkit を使用するために OpenSSL Project が開発したソフトウェア（copyright The OpenSSL Project. コンテンツの無断複製・転載を禁じます）が含まれています。また、このソフトウェアの再配布は、<http://www.openssl.org> および <http://www.openssl.org/source/license.html> にある使用条件に従います。

本製品には、Curl ソフトウェア Copyright 1996-2013, Daniel Stenberg, <[daniel@haxx.se](mailto:daniel@haxx.se)>が含まれます。All Rights Reserved. 本ソフトウェアに関する許諾および制限は、<http://curl.haxx.se/docs/copyright.html> にある使用条件に従います。すべてのコピーに上記の著作権情報とこの許諾情報が記載されている場合、目的に応じて、本ソフトウェアの使用、コピー、変更、ならびに配布が有償または無償で許可されます。

本製品には、MetaStuff, Ltd. のソフトウェアが含まれます。Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved. 本ソフトウェアに関する許諾および制限は、<http://www.dom4j.org/license.html> にある使用条件に従います。

製品には、The Dojo Foundation のソフトウェアが含まれます。Copyright (C) 2004-2007. All Rights Reserved. 本ソフトウェアに関する許諾および制限は、<http://dojotoolkit.org/license> にある使用条件に従います。

本製品には、ICU ソフトウェアおよび他のソフトウェアが含まれます。Copyright International Business Machines Corporation. All rights reserved. 本ソフトウェアに関する許諾および制限は、<http://source.icu-project.org/repos/icu/icu/trunk/license.html> にある使用条件に従います。

本製品には、Per Bothner のソフトウェアが含まれます。Copyright (C) 1996-2006. All rights reserved. お客様がこのようなソフトウェアを使用するための権利は、ライセンスで規定されています。<http://www.gnu.org/software/kawa/Software-License.html> を参照してください。

本製品には、OSSP UUID ソフトウェアが含まれます。Copyright (C) 2002 Ralf S. Engelschall, Copyright (C) 2002 The OSSP Project Copyright (C) 2002 Cable & Wireless Deutschland. 本ソフトウェアに関する許諾および制限は、<http://www.opensource.org/licenses/mit-license.php> にある使用条件に従います。

本製品には、Boost (<http://www.boost.org/>) によって開発されたソフトウェア、または Boost ソフトウェアライセンスの下で開発されたソフトウェアが含まれます。本ソフトウェアに関する許諾および制限は、[http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt) にある使用条件に従います。

本製品には、University of Cambridge のが含まれます。Copyright (C) 1997-2007. 本ソフトウェアに関する許諾および制限は、<http://www.pcre.org/license.txt> にある使用条件に従います。

本製品には、The Eclipse Foundation のソフトウェアが含まれます。Copyright (C) 2007. All Rights Reserved. 本ソフトウェアに関する許諾および制限は、<http://www.eclipse.org/org/documents/epl-v10.php> および <http://www.eclipse.org/org/documents/edl-v10.php> にある使用条件に従います。

本製品には、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.bosrup.com/web/overlib/?License>、<http://www.stlport.org/doc/license.html>、<http://www.asm.ow2.org/license.html>、<http://www.cryptix.org/LICENSE.TXT>、<http://hsqldb.org/web/hsqldbLicense.html>、<http://httpunit.sourceforge.net/doc/license.html>、<http://jung.sourceforge.net/license.txt>、[http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html)、<http://www.openldap.org/software/release/license.html>、<http://www.libssh2.org>、<http://slf4j.org/license.html>、<http://www.sente.ch/software/OpenSourceLicense.html>、<http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>、<http://antlr.org/license.html>、<http://aopalliance.sourceforge.net/>、<http://www.bouncycastle.org/license.html>、<http://www.jgraph.com/jgraphdownload.html>、<http://www.jcraft.com/jsch/LICENSE.txt>、[http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html) に基づいて許諾されたソフトウェアが含まれています。<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>、<http://www.slf4j.org/license.html>、<http://nanoxml.sourceforge.net/orig/copyright.html>、<http://www.json.org/license.html>、<http://forge.ow2.org/projects/javaservice/>、<http://www.postgresql.org/about/licence.html>、<http://www.sqlite.org/copyright.html>、<http://www.tcl.tk/software/tcltk/license.html>、<http://www.jaxen.org/faq.html>、<http://www.jdom.org/docs/faq.html>、<http://www.slf4j.org/license.html>、<http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>、<http://www.keplerproject.org/md5/license.html>、<http://www.toedter.com/en/jcalendar/license.html>、<http://www.edankert.com/bounce/index.html>、<http://www.net-snmp.org/about/license.html>、<http://www.openmdx.org/#FAQ>、[http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt)、<http://srp.stanford.edu/license.txt>、<http://www.schneider.com/blowfish.html>、<http://www.jmock.org/license.html>、<http://xsom.java.net>、<http://benalman.com/about/license/>、<https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>、<http://www.h2database.com/html/license.html#summary>、<http://jsoncpp.sourceforge.net/LICENSE>、<http://jdbc.postgresql.org/license.html>、<http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>、<https://github.com/rantav/hector/blob/master/LICENSE>、<http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>、<http://jibx.sourceforge.net/jibx-license.html>、<https://github.com/lyokato/libgeohash/blob/master/LICENSE>、<https://github.com/hjiang/jsonxx/blob/master/LICENSE>、<https://code.google.com/p/lz4/>、<https://github.com/jedisct1/libsodium/blob/master/LICENSE>、<http://one-jar.sourceforge.net/index.php?page=documents&file=license>、<https://github.com/EsotericSoftware/kryo/blob/master/license.txt>、<http://www.scala-lang.org/license.html>、<https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>、<http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>、<https://aws.amazon.com/asl/>、<https://github.com/twbs/bootstrap/blob/master/LICENSE>、および <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>。

本製品には、Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>)、Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>)、Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>)、Sun Binary Code License Agreement Supplemental License Terms、BSD License (<http://www.opensource.org/licenses/bsd-license.php>)、BSD License (<http://opensource.org/licenses/BSD-3-Clause>)、MIT License (<http://www.opensource.org/licenses/mit-license.php>)、Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>)、Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>) に基づいて許諾されたソフトウェアが含まれています。

本製品には、ソフトウェア copyright (C) 2003-2006 Joe Walnes, 2006-2007 XStream Committers が含まれています。All rights reserved.本ソフトウェアに関する許諾および制限は、<http://j.org/license.html> にある使用条件に従います。本製品には、Indiana University Extreme! Lab によって開発されたソフトウェアが含まれています。詳細については、<http://www.extreme.indiana.edu/> を参照してください。

本製品には、ソフトウェア Copyright (C) 2013 Frank Balluffi and Markus Moeller が含まれています。All rights reserved.本ソフトウェアに関する許諾および制限は、MIT ライセンスの使用条件に従います。

特許については、<https://www.informatica.com/legal/patents.html> を参照してください。

免責: 本文書は、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は他社の権利の非侵害、市場性および特定の目的への適合性の黙示の保証などを含めて、一切の明示的および黙示的保証の責任を負いません。Informatica LLC では、本ソフトウェアまたはドキュメントに誤りのないことを保証していません。本ソフトウェアまたはドキュメントに記載されている情報には、技術的に不正確な記述や誤植が含まれる場合があります。本ソフトウェアまたはドキュメントの情報は、予告なしに変更されることがあります。

## NOTICES

この Informatica 製品（以下「ソフトウェア」）には、Progress Software Corporation（以下「DataDirect」）の事業子会社である DataDirect Technologies からの特定のドライバ（以下「DataDirect ドライバ」）が含まれています。DataDirect ドライバには、次の用語および条件が適用されます。

1. DataDirect ドライバは、特定物として現存するままの状態提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。
2. DataDirect または第三者は、予見の有無を問わず発生した ODBC ドライバの使用に関するいかなる直接的、間接的、偶発的、特別、あるいは結果的損害に対して責任を負わないものとします。本制限事項は、すべての訴訟原因に適用されます。訴訟原因には、契約違反、保証違反、過失、厳格責任、詐称、その他の不法行為を含みますが、これらに限るものではありません。

発行日: 2018-07-18

# 目次

<b>序文</b> .....	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
 <b>第 1 章 : Web サービス</b> .....	12
Web サービスの概要.....	12
Web サービスプロセス.....	12
Web サービスのコンポーネント.....	13
操作.....	13
WSDL.....	14
SOAP.....	14
SOAP Web サービスの開発.....	15
Web サービスの例.....	15
 <b>第 2 章 : WSDL データオブジェクト</b> .....	17
WSDL データオブジェクトの概要.....	17
WSDL データオブジェクトの概要ビュー.....	18
WSDL データオブジェクトの詳細ビュー.....	18
WSDL データオブジェクトのインポート.....	18
WSDL の同期.....	19
WSDL データオブジェクトの同期.....	19
証明書管理.....	20
Informatica Developer の証明書プロパティ.....	20
Informatica Developer への証明書の追加.....	20
 <b>第 3 章 : スキーマオブジェクト</b> .....	21
スキーマオブジェクトの概要.....	21
スキーマオブジェクトの概要ビュー.....	21
スキーマファイル.....	22
スキーマオブジェクトのスキーマビュー.....	22
名前空間のプロパティ.....	23
要素のプロパティ.....	23
単純型のプロパティ.....	25
複合型のプロパティ.....	26

属性プロパティ.....	27
スキーマオブジェクトの詳細ビュー.....	27
スキーマオブジェクトの作成.....	28
スキーマの更新.....	29
スキーマの同期.....	29
スキーマファイルの編集.....	30
証明書管理.....	32
Informatica Developer の証明書プロパティ.....	33
Informatica Developer への証明書の追加.....	33
<b>第 4 章 : Web サービスの作成.....</b>	<b>34</b>
Web サービスの作成の概要.....	34
型と要素.....	35
Web サービスの概要ビュー.....	35
Web サービスの WSDL ビュー.....	37
WSDL データオブジェクトからの Web サービスの作成.....	37
手順 1.WSDL データオブジェクトからの Web サービスの作成.....	38
手順 2.Web サービスへの操作の追加.....	38
WSDL データオブジェクトを Web サービスに関連付ける.....	39
手動での Web サービスの作成.....	39
手順 1.Web サービスの手動作成.....	39
手順 2.操作の作成.....	40
手順 3.要素の作成.....	44
手順 4.定義済みフォールトの作成.....	45
手順 5.ヘッダーの作成.....	45
<b>第 5 章 : 操作マッピング.....</b>	<b>46</b>
操作マッピングの概要.....	46
操作マッピングの [全般] タブ.....	47
操作マッピングの [操作] タブ.....	47
操作マッピングの [詳細] タブ.....	47
入力トランスフォーメーション.....	48
入力トランスフォーメーションの [ポート] タブ.....	48
操作入力ポートにマップするためのルールとガイドライン.....	48
入力トランスフォーメーションの設定.....	49
出力トランスフォーメーション.....	50
出力トランスフォーメーションの [ポート] タブ.....	50
出力トランスフォーメーションの [詳細] タブ.....	51
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.....	51
出力トランスフォーメーションの設定.....	51
フォールトトランスフォーメーション.....	53
フォールトトランスフォーメーションの [ポート] タブ.....	54
フォールトトランスフォーメーションの [詳細] タブ.....	54

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. . . . .	54
フォールトのトランスフォーメーションの作成. . . . .	55
フォールトトランスフォーメーションの設定. . . . .	55
フォールト処理. . . . .	57
システム定義フォールト. . . . .	58
定義済みフォールト. . . . .	59
汎用フォールト. . . . .	60
操作マッピングのテスト. . . . .	60
操作マッピングのテスト. . . . .	60
ビューをカスタマイズするためのオプション. . . . .	61

## 第 6 章 : Web サービス SOAP メッセージの解析. . . . . 62

Web サービス SOAP メッセージの解析の概要. . . . .	62
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. . . . .	63
複数出現出力設定. . . . .	64
正規化したリレーショナル出力. . . . .	64
生成キー. . . . .	64
非正規化したリレーショナル出力. . . . .	65
ピボット化したリレーショナル出力. . . . .	66
anyType 要素の解析. . . . .	66
派生型の解析. . . . .	67
QName 要素の解析. . . . .	68
代替グループの解析. . . . .	68
SOAP メッセージ内の XML 構造の解析. . . . .	68
choice 要素. . . . .	69
list 要素. . . . .	69
union 要素. . . . .	69

## 第 7 章 : Web サービス SOAP メッセージの生成. . . . . 70

Web サービス SOAP メッセージの生成の概要. . . . .	70
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. . . . .	71
[入力ポート] 領域. . . . .	71
操作領域. . . . .	72
ポートと階層レベルのリレーション. . . . .	72
キー. . . . .	73
ポートのマッピング. . . . .	74
ポートのマッピング. . . . .	75
グループのマッピング. . . . .	75
複数のポートのマッピング. . . . .	76
複数出現ポートのピボット化. . . . .	76
非正規化データのマッピング. . . . .	77
派生型および要素の置き換え. . . . .	79
派生型の生成. . . . .	79

anyType 要素および属性の生成. . . . .	79
置き換えグループの生成. . . . .	80
SOAP メッセージ内の XML 構造の生成. . . . .	80
choice 要素. . . . .	80
list 要素. . . . .	81
union 要素. . . . .	82
<b>第 8 章 : Web サービスの管理. . . . .</b>	<b>83</b>
Web サービスの管理の概要. . . . .	83
Web サービスのプロパティの構成. . . . .	83
Web サービスのプロパティ. . . . .	84
Web サービス操作のプロパティ. . . . .	86
Web サービスの結果セットキャッシュ処理. . . . .	86
Web サービスのセキュリティ管理. . . . .	87
Web サービスの権限. . . . .	88
SOAP リクエスト内のユーザー名トークン. . . . .	88
Web サービスのログ. . . . .	90
Web サービスのトレースレベル. . . . .	91
Web サービスの監視. . . . .	91
Web サービスの [プロパティ] ビュー. . . . .	91
Web サービスの [レポート] ビュー. . . . .	92
Web サービスの [操作] ビュー. . . . .	92
Web サービスの [要求] ビュー. . . . .	92
<b>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. . . . .</b>	<b>93</b>
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. . . . .	93
SOAP メッセージ. . . . .	94
WSDL ファイル. . . . .	94
操作. . . . .	95
Web サービスのセキュリティ. . . . .	95
WSDL の選択. . . . .	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ポート. . . . .	97
HTTP ヘッダー入力ポート. . . . .	98
その他の入力ポート. . . . .	98
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. . . . .	99
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. . . . .	100
[ビューのカスタマイズ] のオプション. . . . .	100
操作入力への入力ポートのマッピング. . . . .	100
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. . . . .	102
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. . . . .	103
SOAP メッセージを XML としてマップ. . . . .	103
[ビューのカスタマイズ] のオプション. . . . .	103
出力ポートへの操作出力のマッピング. . . . .	104

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.....	105
Web サービスのエラー処理.....	107
メッセージの圧縮.....	108
並行処理.....	108
フィルタの最適化.....	109
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.....	109
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.....	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作成.....	111
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.....	113
入力ファイル.....	113
論理データオブジェクトモデル.....	113
論理データオブジェクトマッピング.....	114
Web サービスコンシューマトランスフォーメーション.....	115

## 第 10 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..... 117

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.....	117
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.....	119
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.....	119
メッセージの設定.....	119
リソースの識別.....	120
HTTP メソッド.....	121
HTTP Get メソッド.....	121
HTTP Post メソッド.....	122
HTTP Put メソッド.....	122
HTTP Delete メソッド.....	123
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.....	124
入力ポート.....	124
出力ポート.....	124
パススルーポート.....	124
引数ポート.....	125
URL ポート.....	125
HTTP ヘッダポート.....	125
クッキーポート.....	126
出力 XML ポート.....	126
応答コードポート.....	126
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.....	126
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.....	127
メソッド入力への入力ポートのマッピング.....	127
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.....	129
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.....	129
ビューのカスタマイズのオプション.....	130
出力ポートへのメソッド出力のマッピング.....	130



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.....	131
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.....	132
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.....	132
<b>付録 A: データ型の互換性.....</b>	<b>133</b>
データ型リファレンスの概要.....	133
XML データ型とトランスフォーメーションデータ型.....	134
Decimal.....	136
<b>索引.....</b>	<b>137</b>

# 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 Network

Informatica Network は、Informatica グローバルカスタマサポート、Informatica ナレッジベースなどの製品リソースをホストします。Informatica Network には、<https://network.informatica.com> からアクセスしてください。

メンバーは以下の操作を行うことができます。

- 1 つの場所からすべての Informatica のリソースにアクセスできます。
- ドキュメント、FAQ、ベストプラクティスなどの製品リソースをナレッジベースで検索できます。
- 製品の提供情報を表示できます。
- 自分のサポート事例を確認できます。
- 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

ドキュメント、ハウツー記事、ベストプラクティス、PAM などの製品リソースを Informatica Network で検索するには、Informatica ナレッジベースを使用します。

ナレッジベースには、<https://kb.informatica.com> から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](mailto:KB_Feedback@informatica.com)) です。

### Informatica マニュアル

使用している製品の最新のドキュメントを取得するには、[https://kb.informatica.com/\\_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx](https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx) にある Informatica ナレッジベースを参照してください。

このマニュアルに関する質問、コメント、ご意見の電子メールの送付先は、Informatica マニュアルチーム ([infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)) です。

## 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 Network メンバである場合は、PAM (<https://network.informatica.com/community/informatica-network/product-availability-matrices>) にアクセスできます。

## 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ヒントおよびベストプラクティスのコレクションです。数多くのデータ管理プロジェクトの経験から開発された Informatica Velocity には、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が集約されています。

Informatica Network メンバである場合は、Informatica Velocity リソース (<http://velocity.informatica.com>) にアクセスできます。

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](mailto: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 製品を強化したり拡張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。Informatica の開発者およびパートナーの何百ものソリューションを利用して、プロジェクトで実装にかかる時間を短縮したり、生産性を向上させたりできます。Informatica Marketplace には、<https://marketplace.informatica.com> からアクセスできます。

## Informatica グローバルカスタマサポート

Informatica Network の電話またはオンラインサポートからグローバルカスタマサポートに連絡できます。

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 (<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>) を参照してください。

Informatica Network メンバである場合は、オンラインサポート (<http://network.informatica.com>) を使用できます。

# 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](#)
- [Web サービスプロセス, 12 ページ](#)
- [Web サービスのコンポーネント, 13 ページ](#)
- [SOAP Web サービスの開発, 15 ページ](#)
- [Web サービスの例, 15 ページ](#)

## Web サービスの概要

Informatica Data Services は、Web サービスを介してデータ統合機能を提供します。Web サービスは、Developer ツールで作成できます。Web サービスクライアントは、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。

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てアプリケーションを統合します。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、顧客情報と注文情報を取得し、その情報を SOAP レスポンスでクライアントに返します。

## 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。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 サービスクライアントからの Web サービス要求が、Data Integration Service でどのように処理されるかを示しています。

1. Data Integration Service は、Web サービスクライアントから SOAP リクエストを受信します。

2. Data Integration Service の Web サービスモジュールが、操作マッピングを実行して SOAP リクエストを処理します。
3. Data Integration Service の Web サービスモジュールが、Web サービスクライアントに SOAP レスポンスを送信します。

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. トランスフォーメーションは、SOAP リクエストを生成し、Web サービスに接続します。Web サービスコンシューマトランスフォーメーションは、Web サービス接続オブジェクトを使用して Web サービスに接続します。
2. トランスフォーメーションは、Web サービスから SOAP レスポンスを受信します。
3. 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抽出し、出力ポートでデータを返します。

## Web サービスのコンポーネント

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。

## Web サービスの例

さまざまなビジネス要件を満たす Web サービスを作成します。例えば、顧客データにアクセスする Web サービスや顧客のアドレスデータを検証する Web サービスを作成できます。

### 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 ツールで、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 Data Integration Service 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 ツールで、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 Data Integration Service 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 2 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, 17 ページ](#)
- [WSDL データオブジェクトの概要ビュー, 18 ページ](#)
- [WSDL データオブジェクトの詳細ビュー, 18 ページ](#)
- [WSDL データオブジェクトのインポート, 18 ページ](#)
- [WSDL の同期, 19 ページ](#)
- [証明書管理, 20 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 3 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, 21 ページ](#)
- [スキーマオブジェクトの概要ビュー, 21 ページ](#)
- [スキーマオブジェクトのスキーマビュー, 22 ページ](#)
- [スキーマオブジェクトの詳細ビュー, 27 ページ](#)
- [スキーマオブジェクトの作成, 28 ページ](#)
- [スキーマの更新, 29 ページ](#)
- [証明書管理, 32 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。

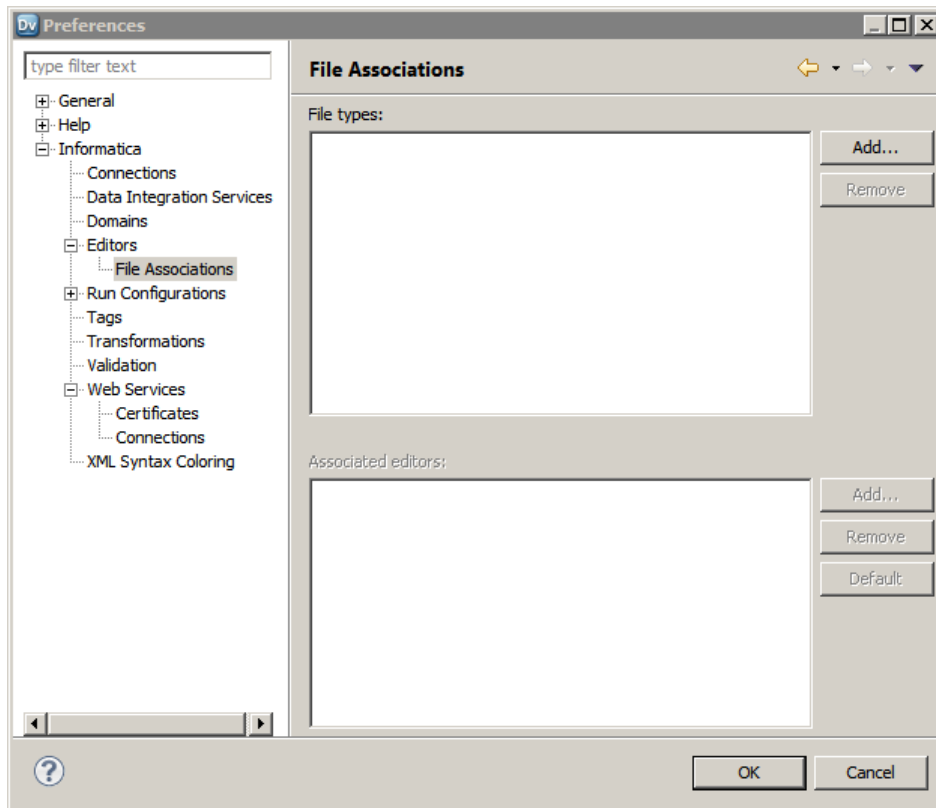
**注:** 一時スキーマファイルを編集および保存する場合、Developer ツールは【スキーマの場所】リストのスキーマファイルを更新しません。Developer ツールでスキーマファイルを編集した後にスキーマオブジェクトを同期した場合、同期操作によって編集内容が上書きされます。

### 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 4 章

# Web サービスの作成

この章では、以下の項目について説明します。

- [Web サービスの作成の概要, 34 ページ](#)
- [型と要素, 35 ページ](#)
- [Web サービスの概要ビュー, 35 ページ](#)
- [Web サービスの WSDL ビュー, 37 ページ](#)
- [WSDL データオブジェクトからの Web サービスの作成, 37 ページ](#)
- [手動での Web サービスの作成, 39 ページ](#)

## Web サービスの作成の概要

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。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 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。

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。

## 手動での Web サービスの作成

WSDL データオブジェクトを使用せずに Web サービスを作成する場合は、Web サービスのプロパティ、操作、および操作コンポーネントを各自で定義します。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 ツールによって、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 5 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, 46 ページ](#)
- [操作マッピングの \[全般\] タブ, 47 ページ](#)
- [操作マッピングの \[操作\] タブ, 47 ページ](#)
- [操作マッピングの \[詳細\] タブ, 47 ページ](#)
- [入力トランスフォーメーション, 48 ページ](#)
- [出力トランスフォーメーション, 50 ページ](#)
- [フォールトトランスフォーメーション, 53 ページ](#)
- [フォールト処理, 57 ページ](#)
- [操作マッピングのテスト, 60 ページ](#)
- [ビューをカスタマイズするためのオプション, 61 ページ](#)

## 操作マッピングの概要

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 ツールで 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、Data Integration Service が SOAP リクエスト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 リクエストは、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 リクエストを受信し、マッピングのダウストリームの変換にデータを返します。データを返された変換は、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 レスポンスメッセージを生成します。

エラーが発生した場合、Data Integration Service はフォールトを生成します。Data Integration Service 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。



## 入力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力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. **【入力マッピング】** 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	<b>【操作入力】</b> 領域内のグループノードまたは子ノードから、 <b>【ポート】</b> 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、 <b>【新しいグループにマップ】</b> 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	<b>【新規】</b> > <b>【グループ】</b> をクリックしてグループを追加します。
<b>【階層の第 1 レベルの抽出】</b> を選択する	<b>【階層の第 1 レベルの抽出】</b> 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、 <b>【ポート】</b> 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。
<b>【マップ】</b> ボタンをクリックする	<b>【操作入力】</b> 領域でノードを 1 つ以上選択します。 <b>【ポート】</b> 領域で抽出先を選択します。 <b>【マップ】</b> 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、 <b>【操作入力】</b> 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 <b>【コピー】</b> ボタンと <b>【貼り付け】</b> 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	<b>【新規】</b> > <b>【フィールド】</b> をクリックしてポートを追加します。
<b>【階層の第 1 レベルの抽出】</b> を選択する	<b>【階層の第 1 レベルの抽出】</b> 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
<b>【クリア】</b> ボタンをクリックします。	<b>【ポート】</b> 領域でポートを 1 つ以上選択し、 <b>【クリア】</b> 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	<b>【操作出力】</b> 領域のグループノードまたは子ノードから、 <b>【ポート】</b> 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	<b>【新規】</b> ボタンの横にある矢印をクリックし、 <b>【新しいグループ】</b> をクリックします。
<b>【階層の第 1 レベルの抽出】</b> を選択する	<b>【階層の第 1 レベルの抽出】</b> 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	<b>【新規】</b> ボタンの横にある矢印をクリックし、 <b>【新しいポート】</b> 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、 <b>【操作出力】</b> 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 <b>【コピー】</b> ボタンと <b>【貼り付け】</b> ボタンを使用します。
<b>【階層の第 1 レベルの抽出】</b> を選択する	<b>【階層の第 1 レベルの抽出】</b> 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pons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。

- Web サービスを開きます。
- 【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。
- 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。
- 【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。
- 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。
- 【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。

- 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。
- 【プロパティ】ビューの【ポート】タブをクリックします。
- 【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。
- 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。
- 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pons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。

データをプレビューするには、事前にデフォルトの Data Integration Service 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

各階層レベルのキーを表示します。

## 第 6 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, 62 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, 63 ページ](#)
- [複数出現出力設定, 64 ページ](#)
- [anyType 要素の解析, 66 ページ](#)
- [派生型の解析, 67 ページ](#)
- [QName 要素の解析, 68 ページ](#)
- [代替グループの解析, 68 ページ](#)
- [SOAP メッセージ内の XML 構造の解析, 68 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。

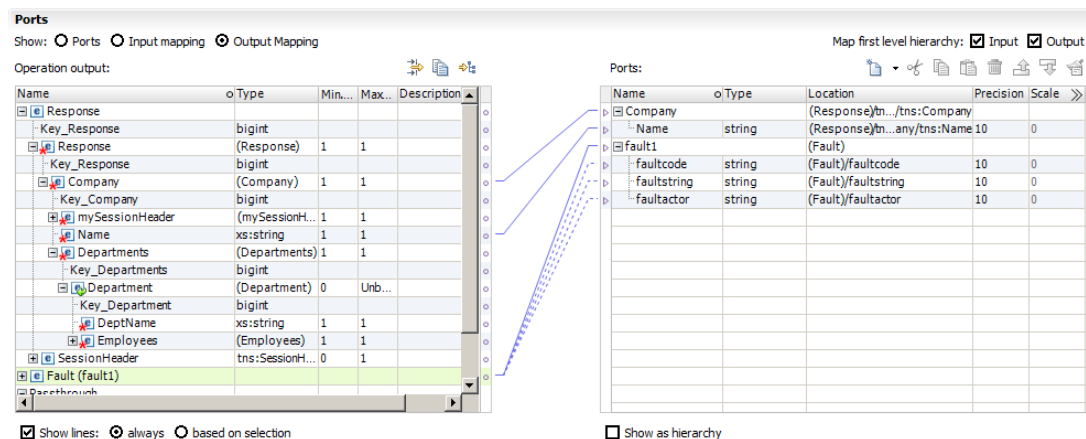
SOAP メッセージを解析するようにトランスフォーメーションを設定するには、SOAP メッセージ階層と同様の構造の出力ポートを作成します。SOAP メッセージ階層内のノードを、作成した出力ポートにマッピングします。

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換えます。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 7 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, 70 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, 71 ページ](#)
- [ポートと階層レベルのリレーション, 72 ページ](#)
- [キー, 73 ページ](#)
- [ポートのマップ, 74 ページ](#)
- [複数出現ポートのピボット化, 76 ページ](#)
- [非正規化データのマップ, 77 ページ](#)
- [派生型および要素の置き換え, 79 ページ](#)
- [SOAP メッセージ内の XML 構造の生成, 80 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。

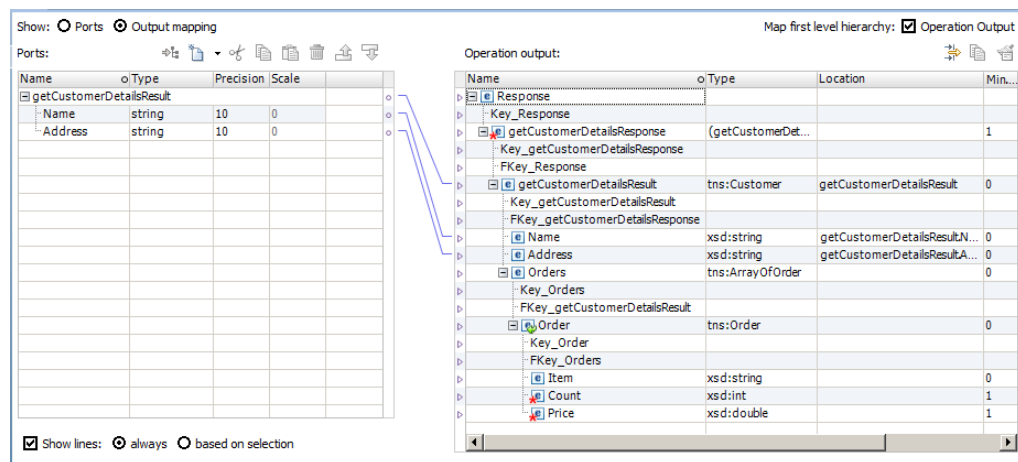
Developer ツールで SOAP メッセージ階層を表示すると、階層にキーが含まれていることがわかります。このキーは、SOAP メッセージには現れません。Data Integration Service では、このキーを使用して、SOAP メッセージ内のグループ間に親子リレーションが定義されます。キー値を設定するには、SOAP メッセージ内のキーに入力データをマップします。

# 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、 <b>【操作】</b> 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 8 章

# 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, 83 ページ](#)
- [Web サービスのプロパティの構成, 83 ページ](#)
- [Web サービスのセキュリティ管理, 87 ページ](#)
- [Web サービスのログ, 90 ページ](#)
- [Web サービスの監視, 91 ページ](#)

## Web サービスの管理の概要

Web サービスをデータ統合サービスにデプロイした後に、Web サービスのセキュリティの設定、Web サービスの設定、Web サービスのログの確認、および Web サービス要求の監視を行うことができます。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 Data Integration Service 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 サービスの設定。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。
- Web サービスのログの確認。Web サービスに関する Data Integration Service 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

Data Integration Service にデプロイする各 Web サービスについて、そのプロパティおよび Web サービス操作のプロパティを設定します。

Web サービスのプロパティを Administrator ツールで表示または編集するには、Data Integration Service の **アプリケーションビュー** を選択し、上部パネルでアプリケーション名を展開して、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。

Web サービス操作のプロパティを Administrator ツールで表示または編集するには、Data Integration Service の **アプリケーションビュー** を選択し、上部パネルでアプリケーション名を展開し、Web サービスを展開して、Web サービス操作を選択します。 **プロパティビュー** にプロパティが表示されます。

## Web サービスのプロパティ

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によって Web サービスの実行時に使用される設定を設定するプロパティが含まれます。

アプリケーションビューの上パネルで Web サービスを展開すると、Web サービスに含まれる Web サービス操作にアクセスできます。

アプリケーションビューには、Web サービスと Web サービス操作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。

次の表に、Web サービスと Web サービス操作の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。
WSDL URL	Web サービスに接続するために使用される WSDL URL。Web サービスについて表示されます。

次の表に、設定可能な Web サービスのプロパティを示します。

プロパティ	説明
スタートアップタイプ	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	<p>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。</p> <ul style="list-style-type: none"> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> </ul> <p>デフォルトは [INFO] です。</p>
要求タイムアウト	データ統合サービスで 1 つの操作マッピングを実行する最大時間（ミリ秒）。この時間を越えると Web サービス要求はタイムアウトします。デフォルトは 3,600,000 です。
最大同時リクエスト数	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	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。
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。
WS-Security を有効にする	データ統合サービスが、ユーザー資格情報を検証し、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。
最適化レベル	<p>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。</p> <ul style="list-style-type: none"> <li>- 0。データ統合サービスは最適化を適用しません。</li> <li>- 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。</li> <li>- 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。</li> <li>- 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。</li> </ul>

プロパティ	説明
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。  整数を指定する必要があります。負の整数を指定すると、データ統合サービスの DTM キープアライブ時間が使用されます。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 サービス操作のプロパティ

Data Integration Service で Web サービス操作を実行する際に使用する設定を設定します。

次の表に、設定可能な Web サービス操作の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、キャッシュには期限がありません。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。有効期間の変更は、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間を使用する場合、期限が切れると結果セットキャッシュはパージされます。デフォルトは 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、Data Integration Service 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、Data Integration Service 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 Data Integration Service 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、Data Integration Service 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、Data Integration Service 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。Data Integration Service 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。Data Integration Service 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. Data Integration Service 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。
2. Web サービス操作のプロパティでキャッシュの有効期間を設定します。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 リクエストの 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。

Web サービスには、以下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 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。

Web サービスまたは Web サービス操作に対する権限を設定するには、Administrator ツールを使用します。

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 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	付与権限	実行権限
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。
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。

## 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。



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>
```

# 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

Web サービスのプロパティ、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確認するには、Web サービスを監視します。

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 ツールで、**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 Data Integration Service 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。**【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 Web サービスに関する情報が表示されます。

Web サービスを監視するには、ナビゲータでアプリケーションを展開して、**Web 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。

# 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。

## Web サービス操作の [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。

## Web サービス操作の [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。

## Web サービス操作の [レポート] ビュー

【レポート】ビューには、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。

## 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, 93 ページ](#)
- [WSDL の選択, 96 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, 97 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, 99 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, 102 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, 105 ページ](#)
- [フィルタの最適化, 109 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, 111 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, 113 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するとします。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ponse メッセージの構造が定義されます。

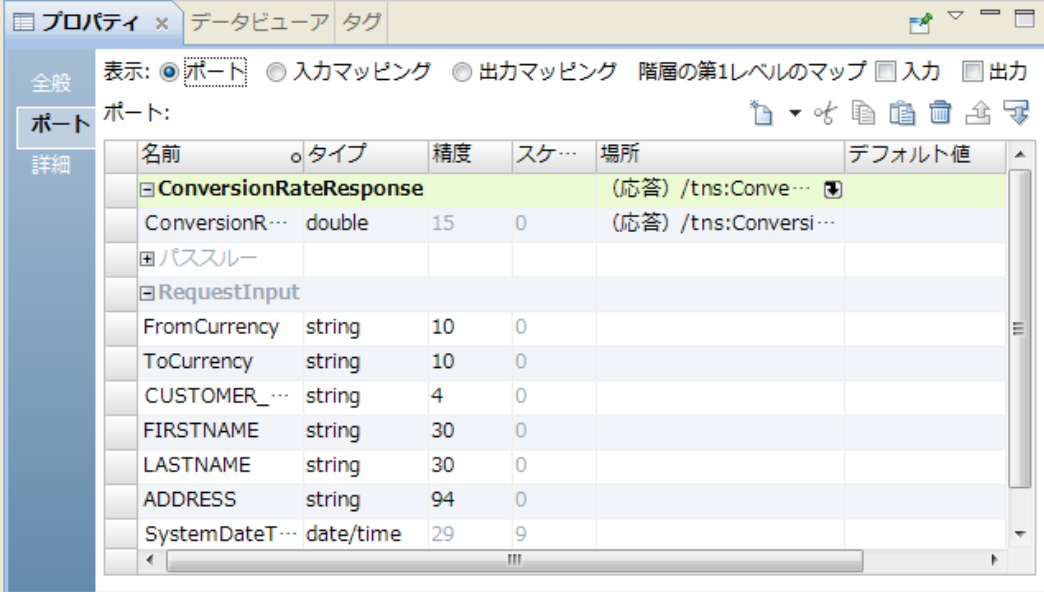
1 方向入力処理を使用して WSDL をインポートすることもできます。1 方向入力処理を使用して WSDL をインポートする場合は、ダミー出力ポートを作成する必要があります。



# 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。



The screenshot shows a software window with tabs: 'プロパティ' (Properties), 'データビューア' (Data Viewer), and 'タグ' (Tags). The 'Ports' tab is active. It has a sidebar with '全般' (General) and 'ポート' (Ports) sections. The 'Ports' section is expanded, showing a table of ports. The table has columns: '名前' (Name), 'タイプ' (Type), '精度' (Precision), 'スケ...' (Scale), '場所' (Location), and 'デフォルト値' (Default Value). The table contains several rows, including 'ConversionRateResponse', 'ConversionR...', 'パススルー' (Pass Through), 'RequestInput', 'FromCurrency', 'ToCurrency', 'CUSTOMER\_...', 'FIRSTNAME', 'LASTNAME', 'ADDRESS', and 'SystemDateT...'. The 'RequestInput' row is highlighted in blue.

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。

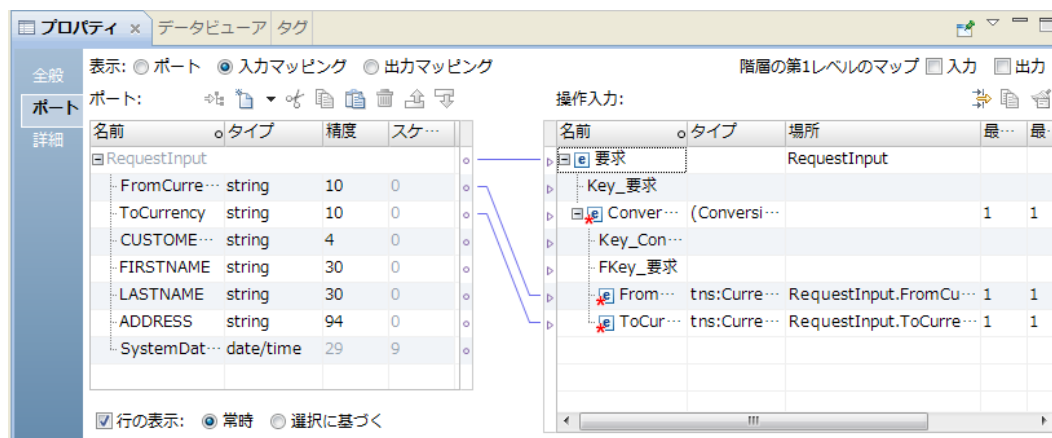
## ポート

【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。

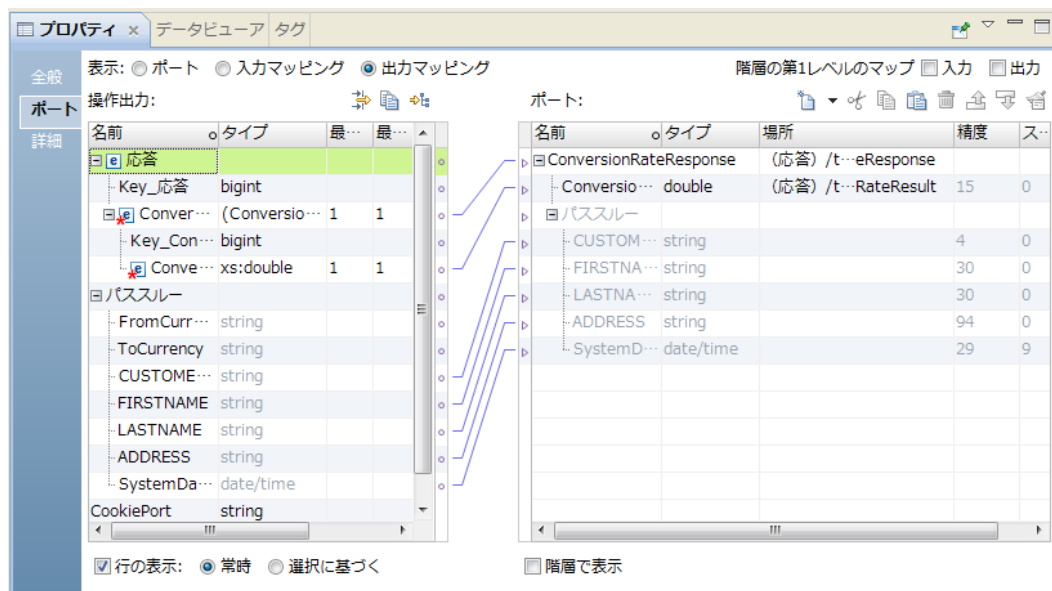
## 操作出力

【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ponse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。

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。
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。

オプション	説明
ノードをドラッグする	<b>【操作出力】</b> 領域のグループノードまたは子ノードを、 <b>【ポート】</b> 領域の空のコ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、 <b>【新規】</b> ボタンの隣の矢印をクリックし、 <b>【グループ】</b> をクリックします。ポートを追加するには、 <b>【新規】</b> ボタンの隣の矢印をクリックし、 <b>【フィールド】</b> 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、 <b>【ポート】</b> 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 <b>【コピー】</b> ボタンと <b>【貼り付け】</b> ボタンを使用します。
<b>【階層の第 1 レベルのマップ】</b> を選択する	<b>【階層の第 1 レベルのマップ】</b> 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。

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。
5. 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
<b>【クリア】</b> をクリックする	<b>【ポート】</b> 領域でポートを 1 つ以上選択し、 <b>【クリア】</b> 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。 <b>【ファイル】 &gt; 【新規】 &gt; 【トランスフォーメーション】</b> 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、 <b>【次へ】</b> 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。

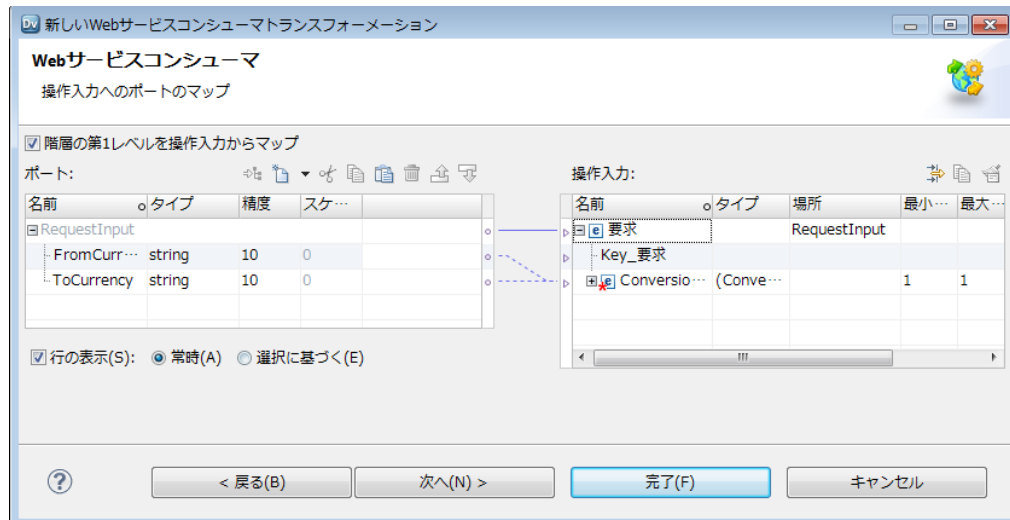
リポジトリ内に WSDL が存在しない場合、[新しい Web サービスコンシューマトランスフォーメーション] ダイアログボックスから WSDL をインポートすることができます。

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。

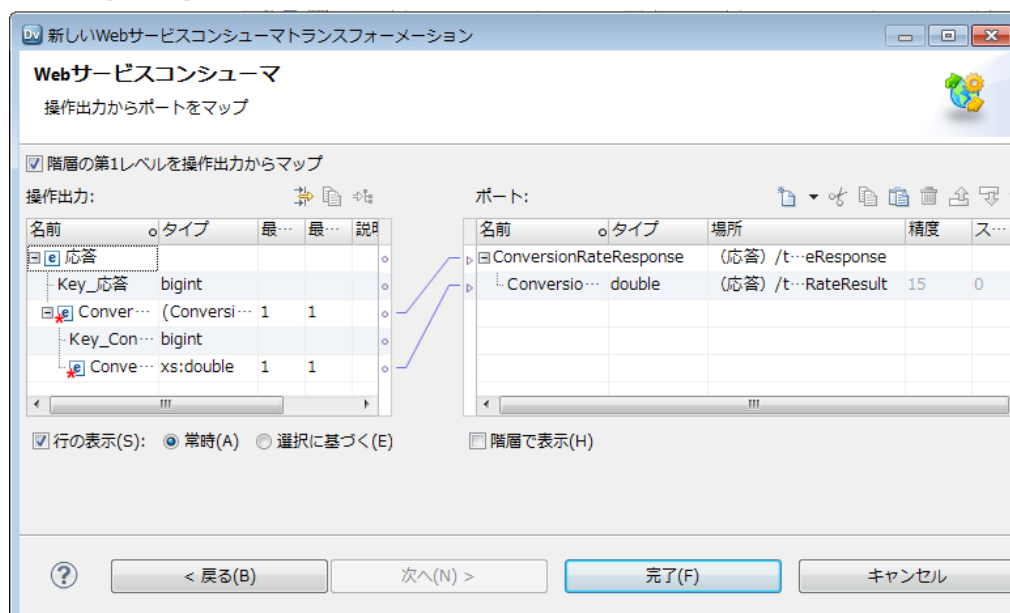
[操作入力へのポートのマップ] 画面が表示されます。[ポート] 領域には、トランスフォーメーションの入力グループと入力ポートが表示されます。[操作入力] 領域には、要求メッセージ階層が表示されます。



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プします。

6. [次へ] をクリックします。

[操作出力からポートをマップ] 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。

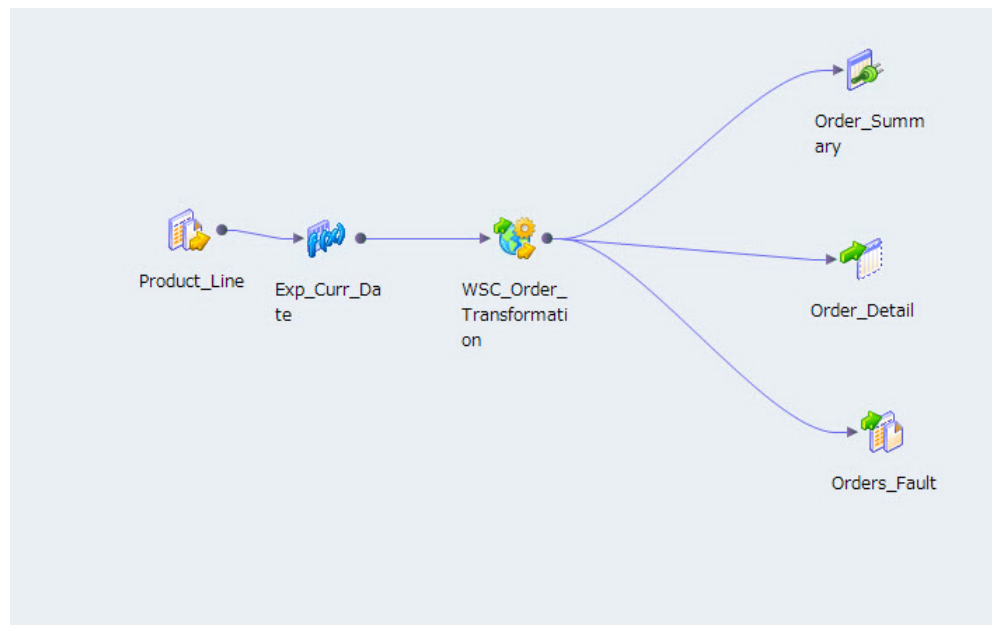
### 論理データオブジェクトモデル

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, 117 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, 119 ページ](#)
- [HTTP メソッド, 121 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, 124 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, 126 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, 129 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, 131 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, 132 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	<a href="http://www.HypoStores.com/products/ProductDetails">http://www.HypoStores.com/products/ProductDetails</a>
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](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 サービスが一意の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。メソッド入力はありません。

### 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メーションのポート

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、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。

トランスフォーメーション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。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メーションからのデータ、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メーション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メーション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、リクエストと応答メッセージの中の 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。

ヘッダ情報を応答から抽出するには、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 HTTP ヘッダポートを設定できます。

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が、**【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。	<b>【メソッド出力】</b> 領域のグループまたは子要素を、 <b>【ポート】</b> 領域の空のカラムにドラッグします。 <b>【ポート】</b> 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、 <b>【新規】</b> ボタンの隣の矢印をクリックし、 <b>【グループ】</b> をクリックします。ポートを追加するには、 <b>【新規】</b> ボタンの隣の矢印をクリックし、 <b>【フィールド】</b> 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、 <b>【ポート】</b> 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 <b>【コピー】</b> ボタンと <b>【貼り付け】</b> 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
<b>【クリア】</b> をクリック。	<b>【ポート】</b> 領域でポートを 1 つまたは複数選択し、 <b>【クリア】</b> 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。

## 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, 133 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, 134 ページ](#)
- [Decimal, 136 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。Data Integration Service 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 ツールは、以下の 2 種類のデータ型を扱います。

- ネイティブデータ型。リレーショナルテーブルまたはフラットファイル固有の型で、物理データオブジェクトとして使用されます。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で扱われるデータ型のセットです。これは ANSI SQL-92 汎用データタイプに基づく内部データタイプで、Data Integration Service がプラットフォーム間でデータを移動するために使用します。トランスフォーメーションデータタイプはマッピングのすべてのトランスフォーメーション内に表示されます。

Data Integration Service は、ソースデータからの読み込み時に、ネイティブデータ型を同等の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。Data Integration Service は、ターゲットへの書き込み時に、トランスフォーメーションデータタイプを同等の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、データタイプは 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 [130](#)  
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 [99](#), [102](#)  
anyType  
ポートのマップ [79](#)  
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 [99](#), [102](#)  
解析 [66](#)  
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 [99](#), [102](#)  
attributeFormDefault  
スキーマオブジェクト [27](#)

## 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 [130](#)  
SOAP メッセージの解析 [69](#)  
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 [100](#), [103](#)  
説明 [80](#)

## E

elementFormDefault  
スキーマオブジェクト [27](#)

## G

gzip  
SOAP メッセージの圧縮 [108](#)

## H

HTTP POST  
ヘッダー [45](#)  
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 [105](#)  
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 [126](#)  
HTTP 接続  
REST Web サービス [131](#)  
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 [125](#)  
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 [98](#)

## N

ノード  
Web サービスフォールト処理 [57](#)  
NULL 可能  
スキーマオブジェクト [23](#)

## Q

Qname 要素  
SOAP メッセージの解析 [68](#)

## R

理由  
Web サービスフォールト処理 [57](#)  
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 [123](#)  
GET メソッド [121](#)  
HTTP ヘッダポート [125](#)  
HTTP メソッド [121](#)  
Post メソッド [122](#)  
Put メソッド [122](#)  
RequestInput ポート [124](#)  
URL ポート [125](#)  
XML スキーマ検証 [131](#)  
インターネット媒体のタイプ [131](#)  
応答コードポート [126](#)  
概要 [117](#)  
クッキーポート [126](#)  
再利用可能 [132](#)  
再利用不可 [132](#)  
作成 [132](#)  
出力 XML ポート [126](#)  
出力ポート [124](#)  
出力ポートの割付 [130](#)  
出力マッピング [129](#)  
出力マッピングのルール [129](#)  
出力マッピングビューのカスタマイズ [130](#)  
詳細プロパティ [131](#)  
セキュリティ [95](#)  
接続プロパティ [131](#)  
設定 [119](#)  
ソート済み入力 [131](#)  
トランスポートレイヤセキュリティ [95](#)  
トレースレベル [131](#)  
入力ポート [124](#)  
入力マッピング [126](#)  
入力マッピングのルール [127](#)  
パススルーポート [124](#)  
引数ポート [125](#)  
プロキシサーバーのサポート [117](#)  
プロセス [119](#)  
ベース URL の設定 [131](#)

## 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)

- ポート [124](#)
- ポートへの要素のマッピング [124](#)
- マッピング出力 [117](#)
- マッピング入力 [117](#)
- マッピング入力ポート [127](#)
- メッセージの設定 [119](#)
- リソースの識別 [120](#)

## 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 [130](#)

### SOAP 1.1

- フォールト要素 [57](#)

### SOAP 1.2

- フォールト要素 [57](#)

### SOAP Web サービス

- 開発 [15](#)

### 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 [105](#)

### 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 [105](#)

### 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 [108](#)

### SOAP 階層

- 入力ポートへのリレーション [72](#)

### SOAP メッセージ

- anyType 要素の解析 [66](#)
- choice 要素の解析 [69](#)
- choice 要素のまっぴんぐ [80](#)
- union 要素へのポートのマッピング [82](#)
- 概要 [94](#)
- keys [73](#)
- 代替グループの解析 [68](#)
- ピボット化するデータ [76](#)
- 複数出現ノードのマッピング [64](#)
- 複数の入力ポートをマッピング [76](#)
- ポートのマッピング [74](#)
- リストの要素の解析 [69](#)
- リストの要素のマッピング [81](#)

### SOAP メッセージ解析

- Qname 要素 [68](#)
- union 要素 [69](#)
- 正規化した出力 [64](#)
- 説明 [62](#)
- 派生型 [67](#)
- 非正規化した出力 [65](#)
- ピボット化した出力 [66](#)

## U

### union 要素

- SOAP メッセージの解析 [69](#)
- 説明 [82](#)

### UsernameToken

- パスワードダイジェスト [89](#)
- パスワードタイプ [88](#)
- ハッシュパスワード [89](#)
- プレーンテキストパスワード [88](#)

## 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 [105](#)
- HTTP ヘッダーの追加 [98](#)
- SOAP 圧縮 [108](#)
- SOAP メッセージ [94](#)
- エラー処理 [107](#)
- エンドポイント URL [98](#)
- 概要 [93](#)
- キーの表示 [100, 103](#)
- クッキー認証 [98](#)
- 作成 [111](#)
- 出力マッピング [102](#)
- 詳細プロパティ [105](#)
- セキュリティ [95](#)
- 操作 [95](#)
- 同時 Web サービス要求のメッセージ [105](#)
- 動的 Web サービス URL [98](#)
- 動的な WS-Security 名 [98](#)
- トランスポートレイヤセキュリティ [95](#)
- 入力マッピング [99](#)
- 汎用フォールト出力の有効化 [105](#)
- フィルタの最適化 [110](#)
- プッシュイン最適化 [110](#)
- 汎用 SOAP フォールト [107](#)
- マッピング出力ノード [102](#)
- マッピング入力ポート [99](#)
- 最適化にプッシュインの有効化 [111](#)
- 初期選択の最適化 [109](#)

### Web サービス

- anyType へのポートのマッピング [79](#)
- Output トランスフォーメーションの設定 [51](#)
- WSDL [14](#)
- WSDL URL [14](#)
- WSDL からの作成 [37](#)
- WSDL なしでの作成 [39](#)
- WSDL の関連付け [37](#)
- WSDL ビュー [37](#)
- 演算 [13](#)
- 概要 [12](#)
- 概要ビュー [35](#)
- 監視 [91](#)
- 結果セットキャッシュの設定 [86](#)
- 権限 [88](#)
- 権限のタイプ [88](#)
- コンポーネント [13](#)
- 作成 [34](#)
- システム定義フォールト [58](#)
- 手動による作成 [39](#)
- セキュリティ [87](#)
- 操作プロパティ [86](#)
- 代替グループ [80](#)
- 定義済みフォールト [59](#)
- 定義済みフォールトの作成 [45](#)
- デプロイメント [15](#)
- トレースレベル [91](#)
- 入力トランスフォーメーション [48](#)
- 入力トランスフォーメーションの設定 [49](#)
- 派生型 [79](#)
- 汎用フォールト [60](#)
- フォールト処理 [57](#)
- フォールトトランスフォーメーションの設定 [55](#)
- プロセス [12](#)
- プロパティ [84](#)
- プロパティの設定 [83](#)
- ヘッダーの作成 [45](#)
- 要素の作成 [44](#)

Web サービス (続く)

例 [15](#)

ログ [90](#)

Web サービス接続

概要 [105](#)

Web サービストランスフォーメーション

[場所] カラム [72](#)

Web サービスのセキュリティ

HTTPS [87](#)

HTTP クライアントフィルタ [87](#)

トランスポートレイヤセキュリティ [87](#)

パススルーセキュリティ [87](#)

メッセージレイヤセキュリティ [87](#)

権限 [87](#)

認証 [87](#)

WS-Security ユーザー名

動的ポート [98](#)

WSDL

Web サービスとの関連付け [37](#)

WSDL からの Web サービスの作成 [37](#)

説明 [14](#)

WSDL URL

説明 [14](#)

WSDL データオブジェクト

インポート [17](#)

概要ビュー [18](#)

作成 [18](#)

詳細ビュー [18](#)

スキーマビュー [17](#)

同期 [19](#)

WSDL ファイル

サービス要素 [94](#)

操作の要素 [94](#)

バインディング要素 [94](#)

ポート要素 [94](#)

## 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 [50](#)

フォールトトランスフォーメーション [53](#)

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 [48](#)

## え

演算

説明 [13](#)

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 [125](#)

Web サービスコンシューマトランスフォーメーション [98](#)

## 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 [126](#)

## 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 [50](#)

入力トランスフォーメーション [48](#)

フォールトトランスフォーメーション [54](#)

## き

keys

SOAP メッセージ階層 [73](#)

基本型のプロパティ

スキーマオブジェクト [25](#)

## く

空白を省略プロパティ

スキーマオブジェクト [25](#)

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 [126](#)

Web サービスコンシューマトランスフォーメーション [98](#)

## 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 [86](#)

設定 [86](#)

## こ

コード

Web サービスフォールト処理 [57](#)

固定値プロパティ

スキーマオブジェクト [23](#)

## さ

サービス

WSDL ファイル要素 [94](#)

最小出現回数

スキーマオブジェクト [23](#)

最小長

スキーマオブジェクト [23](#)

最大出現回数

スキーマオブジェクト [23](#)

最大長

スキーマオブジェクト [23](#)

作成

フォールトトランスフォーメーション [55](#)

## し

システム定義フォールト

Web サービスフォールト処理 [58](#)

出力トランスフォーメーション

Web サービスでの設定 [51](#)

XML として出力するためにマップ [50](#)

階層の第 1 レベルのマップ [50](#)

[詳細設定] タブ [51](#)

説明 [50](#)

ビューのカスタマイズオプション [61](#)

[ポート] タブ [50](#)

マッピングのルールとガイドライン [51](#)

出力マッピング

REST Web サービスコンシューマトランスフォーメーション [129](#)

Web サービスコンシューマトランスフォーメーション [102](#)

種類プロパティ

スキーマオブジェクト [25](#)

[詳細設定] タブ

出力トランスフォーメーション [51](#)

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 [131](#)  
  Web サービスコンシューマトランスフォーメーション [105](#)  
詳細要素  
  Web サービスフォールト処理 [57](#)  
認証  
  UsernameToken [88](#)  
証明書  
  証明書の管理 [20, 32](#)  
  証明書のプロパティ [20, 33](#)  
  信頼されていない証明書 [20, 32](#)  
  信頼されていない証明書の追加 [20, 33](#)

## す

スキーマオブジェクト  
  attributeFormDefault [27](#)  
  elementFormDefault [27](#)  
  インポート [28](#)  
  概要 [21](#)  
  概要ビュー [21](#)  
  スキーマビュー [22](#)  
  スキーマファイル [22](#)  
  スキーマファイルの編集 [29](#)  
  属性プロパティ [27](#)  
  代替グループ [24](#)  
  単純型 [25](#)  
  抽象プロパティ [23](#)  
  デフォルトエディタとして設定 [30](#)  
  同期 [29](#)  
  名前空間 [23](#)  
  ファイルの場所 [27](#)  
  複合要素 [26](#)  
  ブロックプロパティ [24](#)  
  プロパティからの継承 [26](#)  
  プロパティによる継承 [26](#)  
  要素の詳細プロパティ [24, 26](#)  
  要素の詳細プロパティ [24, 26](#)  
  要素のプロパティ [23](#)  
スキーマビュー  
  スキーマオブジェクト [22](#)  
  単純型の詳細プロパティ [25](#)  
スキーマファイル  
  スキーマオブジェクトからの削除 [22](#)  
  スキーマオブジェクトへの追加 [22](#)  
  デフォルトエディタとして設定 [30](#)  
  編集 [31](#)  
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 [100, 103](#)

## せ

生成キー  
  Web サービス出力グループ [64](#)  
生成されたプレフィックス  
  名前空間の変更 [22](#)  
セキュリティ  
  UsernameToken [88](#)  
  Web サービスのセキュリティ [87](#)  
接続  
  REST Web サービス [131](#)  
  Web サービス [105](#)

## そ

操作  
  WSDL ファイル要素 [94](#)  
操作出力  
  説明 [13](#)  
[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 [103](#)  
操作入力  
  説明 [13](#)  
[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 [100](#)  
操作フォール  
  説明 [13](#)  
操作マッピング  
  概要 [46](#)  
  出力トランスフォーメーション [50](#)  
  テスト [60](#)  
  入力トランスフォーメーション [48](#)  
  フォールトトランスフォーメーション [53](#)  
操作領域  
  Web サービストランスフォーメーション [72](#)  
属性プロパティ  
  スキーマオブジェクト [27](#)

## た

代替グループ  
  SOAP メッセージの解析 [68](#)  
  Web サービス [80](#)  
  Web サービスコンシューマトランスフォーメーション [99, 102](#)  
  スキーマオブジェクト [24](#)  
単純型  
  スキーマオブジェクト [25](#)  
単純型ビュー  
  スキーマオブジェクト [25](#)

## ち

抽象プロパティ  
  スキーマオブジェクト [23](#)

## て

定義済みフォールト  
  Web サービスフォールト処理 [59](#)  
  作成 [45](#)  
データ型  
  XML [134](#)  
  概要 [133](#)

## 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 [105](#)  
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 [98](#)  
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 [95](#)  
  Web サービスコンシューマトランスフォーメーション [95](#)  
トレースレベル  
  Web サービス [91](#)

## な

### 名前空間

スキーマオブジェクト [23](#)

生成されたプレフィックスの変更 [22](#)

## に

### 入力トランスフォーメーション

Web サービスでの設定 [49](#)

XML として入力するためにマップ [48](#)

階層の第 1 レベルのマップ [48](#)

説明 [48](#)

ビューのカスタマイズオプション [61](#)

[ポート] タブ [48](#)

マッピングのルールとガイドライン [48](#)

### [入力ポート] 領域

SOAP メッセージの生成 [71](#)

### 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 [126](#)

Web サービスコンシューマトランスフォーメーション [99](#)

## は

### バインディング

WSDL ファイル要素 [94](#)

### [場所] カラム

Web サービストランスフォーメーション [72](#)

### パスワード

ハッシュ [89](#)

プレーンテキスト [88](#)

### 派生型

SOAP メッセージの解析 [67](#)

Web サービス [79](#)

### 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 [99](#), [102](#)

### パターンプロパティ

スキーマオブジェクト [23](#)

### 汎用フォールト

Web サービスフォールト処理 [60](#)

### 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 [105](#)

## ひ

### 非正規化した出力

SOAP メッセージ解析 [65](#)

### 非正規化した入力

Web サービスポート [77](#)

### ピボット化した出力

SOAP メッセージ解析 [66](#)

### ピボット化したデータ

SOAP メッセージ [76](#)

### ビューのカスタマイズオプション

説明 [61](#)

## 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 [110](#)

### フォールトアクタ

Web サービスフォールト処理 [57](#)

### フォールトコード

Web サービスフォールト処理 [57](#)

### 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 [55](#)

XML として出力するためにマップ [53](#)

階層の第 1 レベルのマップ [54](#)

作成 [55](#)

説明 [53](#)

定義済みフォールト [55](#)

汎用フォールト [55](#)

ビューのカスタマイズオプション [61](#)

[ポート] タブ [54](#)

マッピングのルールとガイドライン [54](#)

### フォールト文字列

Web サービスフォールト処理 [57](#)

### フォールト要素

SOAP 1.1 [57](#)

SOAP 1.2 [57](#)

### 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 [105](#)

### 複合型

詳細プロパティ [26](#)

### 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 [127](#)

Web サービスコンシューマトランスフォーメーション [99](#)

### 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 [110](#)

### ブロックプロパティ

スキーマオブジェクト [24](#)

### プロパティからの継承

スキーマオブジェクト [26](#)

### プロパティによる継承

スキーマオブジェクト [26](#)

## へ

### ヘッダ

説明 [13](#)

### ヘッダー

HTTP POST [45](#)

作成 [45](#)

## ほ

### ポート

SOAP メッセージへのマップ [74](#)

非正規化した Web サービストランスフォーメーション [77](#)

### [ポート] タブ

出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 [48](#)

フォールトトランスフォーメーション [54](#)

### 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 [107](#)

## め

### メンバタイプ

スキーマオブジェクト [25](#)

## よ

### 要素

共用体 [82](#)

作成 [44](#)

## り

リストの要素

SOAP メッセージの解析 [69](#)

説明 [81](#)

## れ

列挙プロパティ

スキーマオブジェクト [23](#)

## ろ

ロール

Web サービスフォールト処理 [57](#)