



Informatica®

10.1

Informatica® PowerCenter Express

10.1

開発者用マッピングガイド マッピングガイド

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>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 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.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 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://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: 2018-06-25

目次

序文	12
Informatica のリソース	12
Informatica Network	12
Informatica ナレッジベース	13
Informatica マニュアル	13
Informatica 製品可用性マトリックス	13
Informatica Velocity	13
Informatica Marketplace	13
Informatica グローバルカスタマサポート	13
第 1 章: マッピング	15
マッピングの概要	15
マッピングオブジェクト	16
マッピング内のオブジェクトの依存関係	17
マッピングへのオブジェクトの追加	17
マッピングの開発	18
マッピングの作成	18
SQL クエリからのマッピングの生成	18
マッピングを生成するための SQL 構文	19
SQL クエリからのマッピングまたは論理データオブジェクトの生成	21
ポートのリンク	22
1 対 1 のリンク	23
1 対多のリンク	23
ポートの手動リンク	24
ポートの自動リンク	24
ポートのリンクに関するルールおよびガイドライン	25
ポート属性のプロパゲート	26
依存関係のタイプ	26
リンクパスの依存関係	26
暗黙の依存関係	27
トランスフォーメーションでプロパゲートされるポート属性	27
マッピングランタイムプロパティ	29
ターゲットロード順の制約	32
ターゲットの [ロード順序] タブ	32
挿入行と削除行に関する制約	33
ターゲットのロード順に関するルールとガイドライン	34
制約の作成	34
ターゲットのロード順の例	35
マッピングの検証	36
接続の検証	36

式の検証.	37
オブジェクトの検証.	37
マッピングの検証.	37
マッピングの実行.	37
セグメント.	37
セグメントのコピー.	38
第2章: マプレット.	39
マプレットの概要.	39
マプレットのタイプ.	41
マプレットの入力と出力.	41
マプレットの入力.	41
マプレットの出力.	42
生成されたマプレット.	42
生成されたマプレットのルールおよびガイドライン.	42
マプレットの生成.	42
マプレットの作成.	44
マプレット検証.	44
マプレットの検証.	44
ルール検証としてのマプレット.	44
第3章: マッピングパラメータ.	45
マッピングパラメータの概要.	45
システムパラメータ.	46
ユーザー定義のパラメータ.	47
日付/時刻パラメータ.	48
ユーザー定義のパラメータを作成する場所.	48
パラメータを割り当てる場所.	49
式のパラメータ.	52
SQL 文のパラメータ.	54
リレーショナルテーブルリソースのパラメータ.	55
フィールドおよびプロパティ値のパラメータ.	56
ポートリストのパラメータ.	57
マッピングのパラメータ.	57
パラメータインスタンス値.	58
マプレットのパラメータ.	59
マプレットのパラメータインスタンス値.	59
マッピングのマプレットパラメータ.	60
マプレットのパラメータの例.	60
論理データオブジェクトのパラメータ.	61
仮想テーブルマッピングのパラメータ.	62
パラメータセット.	63
Infacmd からパラメータセットを使用してマッピングを実行する.	64

パラメータファイル.	64
パラメータファイルの構造.	64
project 要素.	65
application 要素.	66
パラメータファイルに関するルールとガイドライン.	67
パラメータファイルの例.	67
パラメータファイルの作成.	70
パラメータファイルを使用した Mapping の実行.	70
パラメータの設定方法.	71
トランスフォーメーションプロパティのパラメータの作成.	72
式内のパラメータの作成.	74
マッピングパラメータとしてトランスフォーメーションパラメータを公開する.	77
パラメータインスタンス値の設定.	78
パラメータセットの作成.	79
第 4 章: マッピング出力.	82
マッピング出力の概要.	82
ユーザー定義マッピング出力.	83
[出力] ビュー.	83
マッピング出力式.	85
システム定義のマッピング出力.	86
保持されたマッピング出力.	87
保持された値の保守.	88
保持されたマッピング出力およびデプロイメント.	88
マッピング出力のワークフロー変数へのバインド.	89
マップレットのマッピング出力.	90
マップレット出力のマッピング出力へのバインド.	91
論理データオブジェクトのマッピング出力.	93
マッピング出力の設定方法.	93
マッピングの作成.	94
マッピング出力の定義.	96
マッピング出力式の設定.	97
マッピング出力の保持.	99
マッピングタスクの入力への保持された出力の割り当て.	100
マッピング出力のワークフロー変数へのバインド.	101
マップレット出力をマッピング出力にバインドする方法.	102
マップレット出力の定義.	103
マップレットでのマッピング出力式の設定.	104
マップレットからの出力のマッピング出力へのバインド.	105
第 5 章: 動的マッピング.	107
動的マッピングの概要.	107
動的マッピングの設定.	108

動的データソース.....	108
動的マッピングのポートおよびリンク.....	109
動的マッピングのルール.....	110
動的マッピングのパラメータ.....	110
動的ソース.....	111
データソースからカラムを取得する.....	112
フラットファイル名にパラメータを割り当てる.....	113
リレーショナルソースのプロパティにパラメータを割り当てる.....	113
ソースデータオブジェクトにパラメータを割り当てる.....	114
動的ターゲット.....	115
データソースからカラムを取得する.....	116
マッピングフローに基づくターゲットの定義.....	117
データオブジェクトに基づくターゲットの定義.....	117
実行時にターゲットを作成または置換する.....	117
リレーショナルターゲットのプロパティにパラメータを割り当てる.....	119
ターゲットデータオブジェクトにパラメータを割り当てる.....	119
動的ターゲットのルールおよびガイドライン.....	120
動的ポートおよび生成されたポート.....	120
動的ポートおよび生成されたポートの設定.....	121
動的ポートおよび生成されたポートのルールおよびガイドライン.....	122
動的式.....	122
入力ルール.....	123
入力ルールの設定.....	124
ポートを含めるまたは除外する.....	125
残りのすべてのポートを含める.....	125
生成されたポートの名前の変更.....	126
生成されたポートの順序変更.....	129
選択ルールとポートセレクト.....	132
ポートセレクトの設定.....	132
選択ルール.....	133
選択ルールとポートセレクトの例.....	134
設計時リンク.....	134
リンクの解決.....	136
ランタイムリンク.....	136
ランタイムリンクの設定.....	137
ランタイムリンクの例.....	139
動的マッピングのトラブルシューティング.....	139
第 6 章: 動的マッピングを開発して実行する方法.....	142
動的マッピングの開発と実行.....	142
動的ソースの設定.....	143
動的マッピングのソースとしてのパラメータの使用.....	144
実行時にメタデータの変更を取得するためのソースの設定.....	144

動的ポートの作成.	145
入力ルールを使用した動的ポートの設定.	146
手順 1. [入力ルール] ダイアログボックスを開く.	147
手順 2. 入力ルールの定義.	147
手順 2a. 演算子と選択条件の選択.	148
手順 2b. 名前選択条件の詳細の設定.	148
手順 2c. タイプ選択条件の詳細の設定.	149
手順 2d. パターン選択条件の詳細の設定.	149
手順 3. 生成されたポートの名前の変更.	150
手順 4. 生成されたポートの順序変更.	150
手順 5. 動的ポート設定の検証.	150
ポートセレクタの作成.	150
動的式の作成.	151
動的ターゲットの設定.	153
動的マッピングのターゲットとしてのパラメータの使用.	154
実行時のデータソースからのターゲットオブジェクトカラムの取得.	155
実行時にターゲットを作成または置換するための DDL クエリの定義.	155
書き込みトランスフォーメーションポートの定義.	156
ランタイムリンクの作成と設定.	158
ランタイムリンクの作成.	160
動的マッピングの検証と実行.	160
第 7 章 : 動的マッピングの使用例.	162
使用例: リレーショナルソースのメタデータ変更のための動的マッピング.	162
ソーステーブル.	162
ターゲットテーブル.	163
動的マッピング.	163
手順 1. 読み取りトランスフォーメーションの設定.	164
手順 2. ジョイナトランスフォーメーションの設定.	165
手順 3. アグリゲータトランスフォーメーションの設定.	166
手順 4. 書き込みトランスフォーメーションの設定.	169
手順 5. ランタイムリンクの作成と設定.	170
手順 6. マッピングの検証と実行.	170
手順 7. ソーススキーマ変更後のマッピングの実行.	171
使用例: さまざまなソースおよびターゲットの動的マッピングの再利用.	173
ソースファイル.	173
ターゲットファイル.	175
動的マッピング.	175
手順 1. Read_Customer_FF 読み取りトランスフォーメーションの設定.	176
手順 2. Exp_TRIM 式トランスフォーメーションの設定.	176
手順 3. Exp_Output 式トランスフォーメーションの設定.	180
手順 4. Write_customerTrim_FF 書き込みトランスフォーメーションの設定.	182
手順 5. マッピングの検証と保存.	184

手順 6. 異なるソースおよびターゲットに対する動的マッピングの実行.	184
第 8 章 : マッピング管理.	187
マッピング管理の概要.	187
マッピングジョブのプロパティの表示.	188
マッピングジョブのサマリ統計の表示.	188
マッピングジョブの詳細統計の表示.	188
マッピングジョブのログの表示.	189
デプロイ済みのマッピングジョブの再発行.	189
マッピングジョブのキャンセル.	190
拒否ファイル.	190
拒否ファイルの場所.	190
拒否ファイルの内容.	191
第 9 章 : PowerCenter へのエクスポート.	193
PowerCenter へのエクスポートの概要.	193
PowerCenter のリリースの互換性.	194
互換性レベルの設定.	194
マプレットのエクスポート.	194
パラメータを含むマッピングのエクスポート.	195
PowerCenter へのエクスポートのオプション.	195
PowerCenter へのオブジェクトのエクスポート.	197
エクスポートの制限.	198
PowerCenter へのエクスポートに関するルールおよびガイドライン.	199
PowerCenter へのエクスポートのトラブルシューティング.	200
第 10 章 : PowerCenter からのインポート.	201
PowerCenter からのインポートの概要.	201
オーバーライドプロパティ.	202
競合の解決.	202
インポートサマリ.	202
データ型変換.	203
トランスフォーメーションの変換.	203
トランスフォーメーションのプロパティの制限.	204
パラメータを含むマッピングのインポート.	211
システム定義パラメータを含むマッピングのインポート.	211
PowerCenter リポジトリ接続パラメータ.	212
接続の割り当て.	213
PowerCenter からのオブジェクトのインポート.	214
インポートの制限.	215
インポートパフォーマンス.	216

第 11 章 : パフォーマンスのチューニング	217
パフォーマンスのチューニングの概要	217
最適化方式	218
初期プロジェクション最適化方法	218
初期選択最適化方式	219
ブランチ刈り込み最適化方式	219
述部最適化方式	219
コストベースの最適化方式	220
データシップ結合最適化方式	220
準結合最適化方式	221
最適化されたマッピングの表示	222
最適化レベル	222
Developer ツールのマッピングに対する最適化レベルの設定	223
デプロイ済みのマッピングに対する最適化レベルの設定	224
第 12 章 : プッシュダウンの最適化	225
プッシュダウンの最適化の概要	225
プッシュダウンタイプ	226
完全なプッシュダウンの最適化	226
ソースプッシュダウン	227
プッシュダウンの設定	227
トランスフォーメーションロジックのプッシュダウン	228
ソースへのプッシュダウンの最適化	229
リレーショナルソースへのプッシュダウンの最適化	229
ネイティブソースへのプッシュダウンの最適化	231
PowerExchange 非リレーショナルソースへのプッシュダウンの最適化	232
ODBC ソースへのプッシュダウンの最適化	232
SAP ソースへのプッシュダウンの最適化	232
プッシュダウンの最適化の式	233
関数	233
演算子	246
データ統合サービスとソースの出力の比較	248
第 13 章 : パーティション化されたマッピング	249
パーティション化されたマッピングの概要	249
各パイプラインステージごとに 1 つのスレッド	250
各パイプラインステージごとに複数のスレッド	251
パーティション化されたフラットファイルソース	253
同時読み込みのパーティション化	253
パーティション化されたリレーショナルソース	254
パーティション化のリレーショナル接続タイプ	255
パーティション化されたリレーショナルソースの SQL クエリ	255

リレーショナルソースパーティションに関するルールおよびガイドライン.....	256
パーティション化されたフラットファイルターゲット.....	256
パーティション化されたファイルターゲットに対応するための出力ファイルディレクトリの 最適化.....	257
パーティション化されたファイルターゲットのマージオプション.....	257
パーティション化されたファイルターゲットのコマンド.....	259
パーティション化されたリレーショナルターゲット.....	260
パーティション化のリレーショナル接続タイプ.....	261
リレーショナルターゲットパーティションに関するルールおよびガイドライン.....	261
パーティション化されたトランスフォーメーション.....	262
パーティション化されたトランスフォーメーションに関する制限.....	262
トランスフォーメーションのキャッシュのパーティション化.....	263
トランスフォーメーションのためのパーティション化の無効化.....	264
パーティション化されたマッピングにおける順序の維持.....	265
安定ソートの維持.....	266
マッピングの最大並行処理のオーバーライド.....	266
トランスフォーメーションのための提案された並行処理.....	267
アドレスバリデータおよび一致トランスフォーメーションの実行インスタンス数.....	268
並行処理の最大値のオーバーライド.....	269
パーティション化されたマッピングのトラブルシューティング.....	270
索引.....	271

序文

『Informatica Developer マッピングガイド』にはマッピングやマップレットを作成および使用方法、動的マッピングを設定する方法、オブジェクトをエクスポートおよびインポートする方法、パフォーマンスのチューニングおよびプッシュダウンの最適化を有効にする方法についての情報が含まれます。『Informatica Developer マッピングガイド』は Data Services や Data Transformation の開発者など、マッピングやマップレットを開発する開発者を対象として書かれています。このガイドでは、フラットファイルとリレーショナルデータベースの概念および作業環境のデータベースエンジンについて理解していることを前提としています。

『Informatica PowerCenter Express マッピングガイド』にはマッピングやマップレットの作成や使用についての情報が含まれます。『Informatica PowerCenter Express マッピングガイド』にはパフォーマンスのチューニングおよびプッシュダウンの最適化を有効化する方法についての情報も含まれます。『Informatica PowerCenter Express マッピングガイド』はデータ統合開発者を対象として書かれています。このガイドでは、フラットファイルとリレーショナルデータベースの概念、作業環境のデータベースエンジン、およびデータ統合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。

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。
- 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) 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 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) 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 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

マッピング

この章では、以下の項目について説明します。

- [マッピングの概要, 15 ページ](#)
- [マッピングオブジェクト, 16 ページ](#)
- [マッピングの開発, 18 ページ](#)
- [マッピングの作成, 18 ページ](#)
- [SQL クエリからのマッピングの生成, 18 ページ](#)
- [ポートのリンク, 22 ページ](#)
- [ポート属性のプロパゲート, 26 ページ](#)
- [マッピングランタイムプロパティ, 29 ページ](#)
- [ターゲットロード順の制約, 32 ページ](#)
- [マッピングの検証, 36 ページ](#)
- [マッピングの実行, 37 ページ](#)
- [セグメント, 37 ページ](#)

マッピングの概要

マッピングは、ソースとターゲットの間のデータフローを表す入力と出力のセットです。データトランスフォーメーションのルールを定義するトランスフォーメーションオブジェクトによりリンクできます。データ統合サービスは、マッピングに設定された命令を使用して、データの読み込み、変換、および書き込みを行います。

マッピングに含める入力と出力のタイプによって、マッピングのタイプが決まります。Developer ツールでは、以下のタイプのマッピングを作成できます。

- 入力および出力として物理データオブジェクトを持つマッピング
- マッピングの入力または出力として論理データオブジェクトを持つ論理データオブジェクトマッピング
- マッピングの入力または出力あるいはその両方として操作を持つ操作マッピング
- マッピングの出力として仮想テーブルを持つ仮想テーブルマッピング

フラットファイルまたはリレーショナルソースのスキーマまたはメタデータが頻繁に変更される場合、動的マッピングを開発して実行時のスキーマまたはメタデータの変更を管理します。動的マッピングを開発してマッピングロジックを再利用し、他の物理ソースおよびターゲットに対するマッピングを実行することもできます。動的マッピングとは、ソース、ターゲット、トランスフォーメーションのロジックを、定義したパラメータおよびルールに基づいて実行時に変更できるマッピングです。

物理データオブジェクトを含むマッピングを、ワークフローのマッピングタスクにおよび出力として含めることができます。複数のマッピングを順次実行できるように、ワークフローからマッピングを実行する場合があります。または、マッピング実行の前後にコマンドを実行して手順を実行するワークフローを開発できます。

次の方法でマッピングを開発し、実行することもできます。

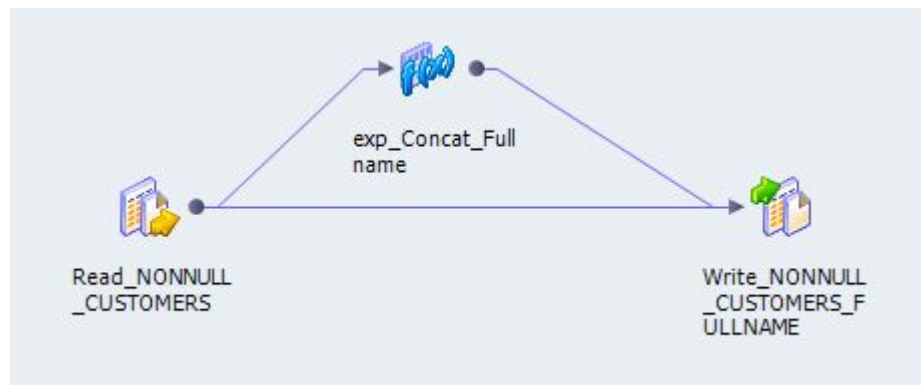
動的マッピングの開発と実行

フラットファイルまたはリレーショナルソースのスキーマまたはメタデータが頻繁に変更される場合、動的マッピングを開発して実行時のスキーマまたはメタデータの変更を管理します。動的マッピングを開発してマッピングロジックを再利用し、他の物理ソースおよびターゲットに対するマッピングを実行することもできます。動的マッピングとは、ソース、ターゲット、トランスフォーメーションのロジックを、定義したパラメータおよびルールに基づいて実行時に変更できるマッピングです。

ワークフローからのマッピングの実行

複数のマッピングを順次実行できるように、ワークフローからマッピングを実行する場合があります。または、マッピング実行の前後にコマンドを実行して手順を実行するワークフローを開発できます。物理データオブジェクトを含むマッピングを、ワークフローのマッピングタスクにおよび出力として含めることができます。

以下の図に、マッピングの例を示します。



マッピングオブジェクト

マッピングオブジェクトは、ソースとターゲットの間のデータフローを決定します。

各マッピングには以下のオブジェクトを含める必要があります。

- 入力。マッピングソースの特性を示します。
- 出力。マッピングターゲットの特性を示します。

各マッピングには以下のコンポーネントを含めることもできます。

- トランスフォーメーション。データをターゲットに書き込む前に変更します。さまざまなトランスフォーメーションオブジェクトを使用して、種々の関数を実行します。
- マップレット。複数のマッピングで使用できるトランスフォーメーションのセットを含む再利用可能なオブジェクト。

オブジェクトをマッピングに追加する際に、Data Integration Service のデータ変換方法に応じて、プロパティを設定します。また、Data Integration Service のデータ移動方法に応じて、マッピングオブジェクトを接続することもできます。オブジェクトは、ポートを介して接続することができます。

エディタはオブジェクトを以下の方法で表示します。

- アイコン化。オブジェクトのアイコンを、オブジェクト名と共に表示します。
- ノーマル。カラムと、入力ポートおよび出力ポートのインジケータを表示します。 ノーマルビュー内のオブジェクトは接続することができます。

マッピング内のオブジェクトの依存関係

マッピングは、独立したオブジェクトとしてリポジトリに格納されているいくつかのオブジェクトに依存しています。

オブジェクトのメタデータが変わると、Developer ツールはマッピングへのこれらの変更の影響を追跡しません。 マッピングは、編集していなくても無効になることがあります。 マッピングが無効になると、Data Integration Service はマッピングを実行できません。

以下のオブジェクトは、独立したオブジェクトとしてリポジトリに格納されます。

- 論理データオブジェクト
- 物理データオブジェクト
- 再利用可能なトランスフォーメーション
- マップレット

マッピングは、これらのオブジェクトに依存しています。

マッピング内の以下のオブジェクトは、依存リポジトリオブジェクトとしてリポジトリに格納されます。

- 仮想テーブル。仮想テーブルは SQL データサービスの一部として格納されます。
- マッピング内に構築した再利用不可能なトランスフォーメーション 再利用不可能なトランスフォーメーションは、マッピング内のみ格納されます。

マッピング内に構築した再利用不可能なトランスフォーメーションは、依存リポジトリオブジェクトとしてマッピング内に格納されます。

マッピングへのオブジェクトの追加

ソースとターゲットの間のデータフローを決定するにはマッピングにオブジェクトを追加します。

1. マッピングを開きます。
2. 物理データオブジェクトをエディタにドラッグして、[読み取り] を選択し、データオブジェクトをソースとして追加します。
3. 物理データオブジェクトをエディタにドラッグして、[書き込み] を選択し、データオブジェクトをターゲットとして追加します。
4. ルックアップトランスフォーメーションを追加するには、フラットファイルデータオブジェクト、論理データオブジェクト、参照テーブル、リレーショナルデータオブジェクトのいずれかをエディタにドラッグして、[ルックアップ] を選択します。ルックアップトランスフォーメーションを追加するには、フラットファイルデータオブジェクト、論理データオブジェクト、リレーショナルデータオブジェクトのいずれかをエディタにドラッグして、[ルックアップ] を選択します。
5. 再利用可能なトランスフォーメーションを追加するには、トランスフォーメーションを **[Object Explorer]** ビューの [トランスフォーメーション] フォルダからエディタにドラッグします。
この手順を、追加する再利用可能なトランスフォーメーションごとに繰り返します。
6. 再利用不可能なトランスフォーメーションを追加するには **[トランスフォーメーション]** パレットでトランスフォーメーションを選択して、エディタにドラッグします。
この手順を、追加する再利用不可能なトランスフォーメーションごとに繰り返します。

7. 再利用不可能なトランスフォーメーションごとに、ポートとプロパティを設定します。
8. 必要に応じて、マップレットをエディタにドラッグします。

マッピングの開発

ビジネスニーズに従ってデータの読み取り、変換、および書き込みを行うためのマッピングを開発します。

1. 作成するマッピングのタイプを決定します。
2. マッピングで使用する入力、出力、および再利用可能オブジェクトを作成します。マッピング入力または出力として使用する物理データオブジェクト、論理データオブジェクト、または仮想テーブルを作成します。マッピング入力または出力として使用する物理データオブジェクトまたは論理データオブジェクトを作成します。使用する再利用可能なトランスフォーメーションを作成します。マップレットを使用する場合は、マップレットも作成する必要があります。
3. マッピングを作成します。
4. オブジェクトをマッピングに追加します。入力および出力オブジェクトをマッピングに追加する必要があります。必要に応じて、トランスフォーメーションおよびマップレットを追加します。
5. マッピングオブジェクト間のポートをリンクしてソースからターゲットへのデータフローを作成します。マップレットやトランスフォーメーションを通じて、このフローに沿ってデータの追加、削除、または修正が行われます。
6. マッピングを検証してエラーを特定します。
7. モデルリポジトリにマッピングを保存します。

マッピングの開発を終了したら、実行してマッピングの出力を確認します。

マッピングの作成

マッピングを作成して、ソースとターゲット間でデータを移動し、データを変換します。

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] > [マッピング]** をクリックします。
3. マッピング名を入力します。
4. **[完了]** をクリックします。
空のマッピングがエディタに表示されます。

SQL クエリからのマッピングの生成

Developer tool で SQL クエリからマッピングを生成できます。マッピングを生成するために、SQL クエリを入力するか、クエリが含まれるテキストファイルをロードできます。必要に応じて、クエリテーブルのソースを定義できます。Developer tool が SQL クエリを検証し、マッピングを生成します。

SELECT 文のみが含まれる SQL クエリから論理データオブジェクトを生成することもできます。

マッピングを生成するための SQL 構文

ANSI 準拠の SQL 文を使用して、Developer tool でマッピングを生成できます。

Developer tool は、標準的な SELECT クエリからマッピングを生成できます。以下に例を示します。

```
SELECT column_list FROM table-name  
[WHERE clause]  
[GROUP BY clause]  
[HAVING clause]  
[ORDER BY clause]
```

SELECT SQL 文に相関サブクエリが含まれている場合、クエリを単一の標準的なクエリとしてフラット化または書き換えることが可能であれば、クエリは有効です。

相関サブクエリ

相関サブクエリとは、WHERE 句に外部クエリからの値を使用するサブクエリです。データ統合サービスはクエリを実行する前に相関サブクエリをフラット化します。

以下の表に、データ統合サービスがフラット化した相関サブクエリの結果を示します。

タイプ	クエリ
フラット化なし	SELECT huge.* FROM huge WHERE c1 IN (SELECT c1 FROM tiny)
フラット化済み	SELECT huge.* FROM huge, tiny WHERE huge.c1 = tiny.c1

データ統合サービスが相関サブクエリをフラット化できるのは、相関サブクエリが次の要件を満たす場合です。

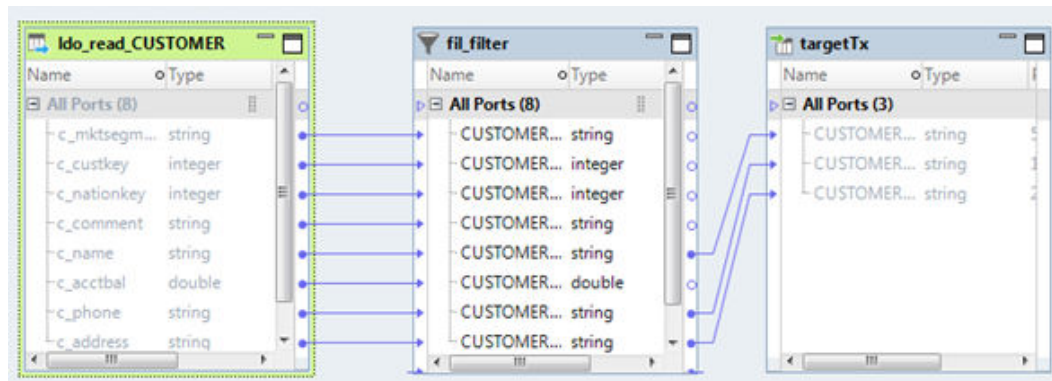
- タイプが IN または定量化された比較である。
- OR 演算子内にない、または SELECT リストの一部でない。
- LIMIT キーワードは含まれていない。
- GROUP BY 句を含まず、SELECT リスト、EXIST または NOT IN 論理演算子に集約する。
- 一意の結果を生成する。相関サブクエリの 1 つの列がプライマリキーである。例えば、r_regionkey 列が vs.nation 仮想テーブルのプライマリキーである場合、次のクエリを発行できます: SELECT * FROM vs.nation WHERE n_regionkey IN (SELECT b.r_regionkey FROM vs.region b WHERE b.r_regionkey = n_regionkey)。
- クエリに FROM リストが含まれる場合、FROM リスト内の各テーブルは SQL データサービス内の仮想テーブルです。

例

次の SQL 文でマッピングを生成できます。

```
SELECT c_name , c_address , c_phone  
  from customer  
 where c_custkey>101 and c_nationkey=2
```

次の図に、SQL 文から作成されたマッピングを示します。



マッピングを生成するクエリでの関数のサポート

Informatica では、ANSI SQL-92 規格を満たす関数をサポートしています。

さらに、一部の関数には特定の構文要件があります。次の表に、関数とサポートされる構文を示します。

機能	構文
DATE()	<p>日付の形式を指定する:</p> <p>DATE(format '<format>')</p> <p>ここで、<format>は標準的な日付形式です。</p> <p>例:</p> <p>SELECT DATE(format 'dd-mm-yyyy') from table</p>
POSITION()	<p>リテラル文字列のサブストリングの位置を判断する:</p> <p>POSITION('<substring>', '<string>')</p> <p>例:</p> <p>POSITION('MA', 'James Martin')</p> <p>テーブルカラムのサブストリングの位置を判断する:</p> <p>POSITION('<substring>', <column_name>)</p> <p>例:</p> <p>POSITION('MA', FULL_NAME)</p>

サポートされない関数を使用した SQL クエリからのマッピングの生成

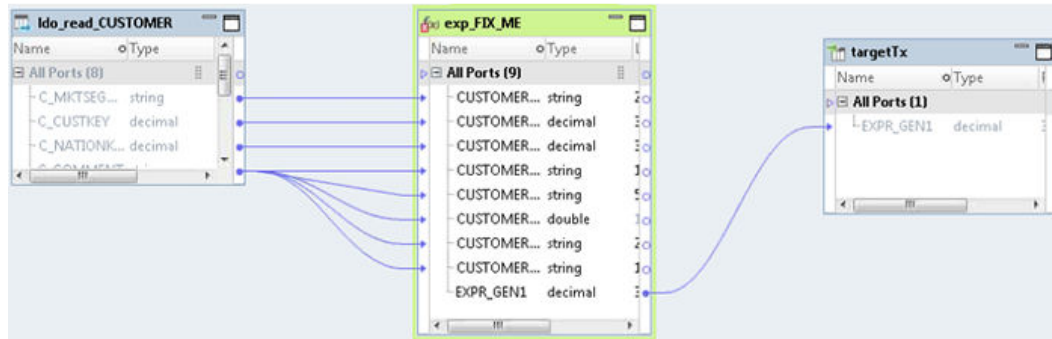
Developer tool は SQL からマッピングを生成するときに、クエリの実関数を検証します。ANSI 準拠の SQL を使用して、有効なマッピング生成であることを確認します。

Developer tool が、有効な SQL 文に不明な関数を検出すると、FIX_ME というラベルの付いたトランスフォーメーションまたは FIX_EXPR というラベルの付いた式が含まれるマッピングを生成することがあります。マッピングを修正して有効な結果を得るには、これらのオブジェクトを編集します。不明な関数は、マッピングログファイルで警告メッセージとして表示されます。

例えば、次の SQL 文を使用してマッピングを生成します。

```
SELECT unknownFunctionABC(c_custkey,c_comment) from customer
```

次の図に、修正が必要な式トランスフォーメーションが含まれる SQL 文から生成されたマッピングを示します。



式トランスフォーメーションにエラーアイコンが付いている点に注意してください。[ポート] タブを使用して、正しくない式を編集します。エラーを修正するまでマッピングは有効になりません。

INSERT、UPDATE、および DELETE の構文

次の構文を使用して、有効な INSERT 文、UPDATE 文、および DELETE 文を作成します。

- INSERT 文には次の構文を使用します。
`INSERT INTO <TABLENAME> [<list>]
 <select query>`
- UPDATE 文には次の構文を使用します。
`UPDATE [schema .] { table | view } [alias]
 SET column = { expr | subquery }
 [, column = { expr | subquery }]... [WHERE condition]`
- DELETE 文には次の構文を使用します。
`DELETE FROM <Table> [[<AS>] <ALIAS>] [WHERE condition]`

INSERT 文、UPDATE 文、および DELETE 文のルールおよびガイドライン

INSERT 文、UPDATE 文、および DELETE 文では以下のルールおよびガイドラインについて検討してください。

- INSERT 文、UPDATE 文、または DELETE 文により、論理データオブジェクトであるマッピングでソースおよびターゲットオブジェクトが作成される。
- 1 つの INSERT 文、UPDATE 文、または DELETE 文のみが有効。例えば、INSERT とネストされた UPDATE 文が含まれる文は有効ではありません。
- INSERT、UPDATE、または DELETE の SQL 文に相関サブクエリが含まれる場合、Developer tool はマッピングを生成できない。
- UPDATE 文または DELETE 文により、マッピングでアップデートストラテジトランスフォーメーションが作成される。アップデートストラテジトランスフォーメーションにはプライマリキーが必要であるため、データターゲットにはプライマリキーを含める必要があります。マッピング生成後に、プライマリキーを確認します。

SQL クエリからのマッピングまたは論理データオブジェクトの生成

SQL 文をマッピングまたは論理データオブジェクトに変換できます。論理データオブジェクトを生成して、他のマッピングで再利用できるオブジェクトを作成することもできます。

1. [ファイル] > [新規] > [SQL クエリからのマッピング] をクリックします。
 [SQL クエリからのマッピングまたは論理データオブジェクトの生成] ダイアログボックスが開きます。

2. SQL クエリを入力するか、SQL クエリが含まれるファイルを選択するかを選択します。
 - 編集可能な SQL クエリを入力するには、**[SQL クエリを入力してください]** を選択し、エディタで SQL クエリを入力するか貼り付けます。次に、**[クエリの検証]** をクリックします。
 - SQL クエリが含まれるファイルを選択するには、**[SQL ファイルを選択してください]** を選択し、SQL クエリが含まれるファイルを参照します。

Developer tool が SQL 構文を検証します。構文が有効でない場合、修正してからでないと続行することはできません。
3. マッピングではなく論理データオブジェクトを生成するには、**[マッピングを論理データオブジェクトとして生成します]** を選択します。
4. 必要に応じて、生成するマッピングまたは論理データオブジェクトの名前を変更します。
5. **[次へ]** をクリックします。

ダイアログボックスに、データソースに対応するテーブルが表示されます。
6. **[データソース]** でテーブルの行をクリックし、マッピングのデータソースを選択します。

テーブルにデータソースがある場合、クリックして必要に応じてデータソースを変更できます。

[データソースの選択] ダイアログボックスが開き、アクセスできるモデルリポジトリのテーブルがリストされます。
7. モデルリポジトリで任意のデータソースを選択します。
8. **[完了]** をクリックします。

生成されたマッピングまたは論理データオブジェクトがエディタで開きます。

マッピングで任意のオブジェクトを選択し、表示または編集できます。その後、マッピングを実行したり、データ統合サービスにデプロイするアプリケーションまたはワークフローに含めたりすることができます。

論理データオブジェクトを作成した場合、他のマッピングで再利用できます。例えば、生成された論理データオブジェクトをマッピングでソースとして再利用できます。

ポートのリンク

入力、出力、トランスフォーメーション、およびマプレットオブジェクトをマッピングで追加したら、マッピングオブジェクト間のポートをリンクしてマッピングを完成させます。

データは以下のポートを介してマッピングを通過します。

- 入力ポート。データを受け取ります。
- 出力ポート。データを渡します。
- 入出力ポート。データを受け取り、変更せずに渡します。

各入力オブジェクト、出力オブジェクト、マプレット、およびトランスフォーメーションには、ポートの集まりが含まれます。各ポートはデータの列を表します。

- 入力オブジェクトはデータを提供するため、出力ポートのみを含みます。
- 出力オブジェクトはデータを受け取るため、入力ポートのみを含みます。
- マプレットは入力ポートと出力ポートのみを含みます。
- トランスフォーメーションは、その種類と用途に応じて、入力ポート、出力ポート、および入出力ポートのいずれをも含みます。

ポートを接続するには、異なるマッピングオブジェクト内のポートをリンクします。Developer ツールは、接続がリンクの検証および連結条件を満たした場合のみ、接続を作成します。

ポートは未接続のままにすることができます。Data Integration Service では、未接続のポートは無視されません。

入力オブジェクト、トランスフォーメーション、マプレット、および出力オブジェクト間でポートをリンクする場合、以下のタイプのリンクを作成できます。

- 1 対 1
- 1 対多

ポートは手動でリンクすることも自動的にリンクすることもできます。

関連項目：

- [「設計時リンク」 \(ページ 134\)](#)

1 対 1 のリンク

入力オブジェクトまたはトランスフォーメーションの 1 つのポートを、出力オブジェクトまたはトランスフォーメーションの 1 つのポートにリンクします。

1 対多のリンク

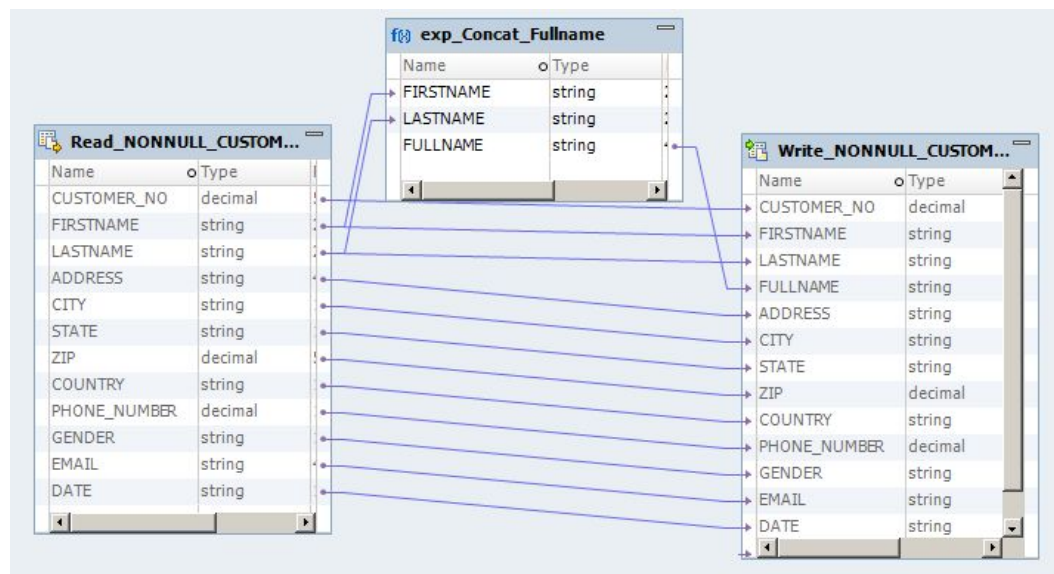
同一のデータを異なる目的で使用する場合には、このデータを提供するポートを、マッピング内の複数のポートにリンクすることができます。

以下の方法で、1 対多のリンクを作成できます。

- 1 つのポートを複数のトランスフォーメーションまたは出力オブジェクトにリンクする。
- 1 つのトランスフォーメーションの複数のポートを複数のトランスフォーメーションまたは出力オブジェクトにリンクする。

例えば、アグリゲータトランスフォーメーションを介して、銀行支店の給与情報を使用して平均給与を計算したい場合、各従業員の毎月の給与計算用に設定した式トランスフォーメーション内の同じ情報を使用できます。

以下の図に、1 対多のリンクを持つマッピングの例を示します。



ポートの手動リンク

1つのポートまたは複数のポートを手動でリンクできます。

入力オブジェクトまたはトランスフォーメーションのポートから出力オブジェクトまたはトランスフォーメーションのポートにドラッグします。

Ctrl キーまたは Shift キーを使用して、別のトランスフォーメーションまたは出力オブジェクトにリンクする複数のポートを選択します。Developer ツールは、最上段のペアから順にポートをリンクします。検証条件を満たしているポートがすべてリンクされます。

空のポートにポートをドラッグした場合、Developer ツールはポートをコピーしてリンクを作成します。

ポートの自動リンク

ポートを自動的にリンクする場合は、位置または名前によってリンクを作成できます。

名前によってポートをリンクする場合は、接頭語または接尾語を指定してポートをリンクできます。接頭語または接尾語は、マッピング内のどこにポートがあるかを示すために使用します。

名前によるポートのリンク

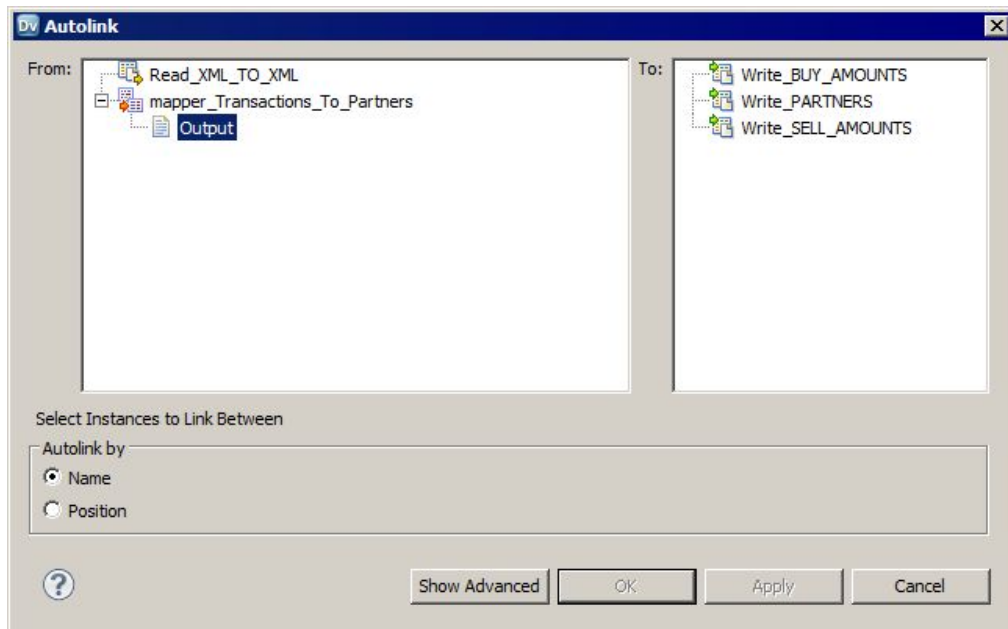
名前でポートをリンクする場合、Developer ツールは同じ名前を持つ入力ポートおよび出力ポート間にリンクを追加します。トランスフォーメーション間で同じポート名を使用している場合に、名前によるリンクを行います。

ユーザーが定義したプレフィックスおよびサフィックスに基づき、ポートをリンクできます。接頭語または接尾語は、マッピング内のどこにポートがあるかを示すために使用します。ポート名でプレフィックスまたはサフィックスを使用してマッピングまたはマップレット内の出現位置を区別している場合、名前およびプレフィックス/サフィックスに基づいてリンクします。

名前によるリンクでは大文字と小文字は区別されません。

1. [マッピング] > [オートリンク] をクリックします。

[オートリンク] ダイアログボックスが表示されます。



2. **【リンク元】** ウィンドウで、リンク元のオブジェクトを選択します。
3. **【リンク先】** ウィンドウで、リンク先のオブジェクトを選択します。
4. **【名前】** を選択します。
5. 必要に応じて、**【詳細の表示】** をクリックして、プレフィックスまたはサフィックスに基づいてポートをリンクします。
6. **【OK】** をクリックします。

位置によるポートのリンク

位置によるリンクを実行すると、Developer ツールは各出力ポートを対応する入力ポートにリンクします。例えば、1 番目の出力ポートは 1 番目の入力ポートに、2 番目の出力ポートは 2 番目の入力ポートにリンクされます。位置によるリンクは、関連するポートが同じ順で設定されているトランスフォーメーションを作成する場合に使用します。

1. **【マッピング】** > **【オートリンク】** をクリックします。
【オートリンク】 ダイアログボックスが表示されます。
2. **【リンク元】** ウィンドウで、リンク元のオブジェクトを選択します。
3. **【リンク先】** ウィンドウで、リンク先のオブジェクトを選択します。
4. **【位置】** を選択し、**【OK】** をクリックします。
Developer ツールは各出力ポートを対応する入力ポートにリンクします。例えば、1 番目の出力ポートは 1 番目の入力ポートに、2 番目の出力ポートは 2 番目の入力ポートにリンクされます。

ポートのリンクに関するルールおよびガイドライン

ポートをリンクする際に適用されるルールとガイドラインがあります。

マッピングオブジェクトを接続する場合には、以下のルールおよびガイドラインに従います。

- 2 つのマッピングオブジェクト間でポートをリンクしようとした時にエラーが検出されると、ポートをリンクできないことを示す記号が表示されます。
- マッピングのデータフローのロジックに従います。以下のタイプのポートをリンクできます。
 - 受け取る側のポートは入力または入出力ポートでなければなりません。
 - 送る側のポートは、出力または入出力ポートでなければなりません。
 - 入力ポートを入力ポートにリンクしたり、出力ポートを出力ポートにリンクすることはできません。
- 少なくとも 1 つの入力グループのポートを先行するトランスフォーメーションにリンクする必要があります。
- 少なくとも 1 つの出力グループのポートを後続のトランスフォーメーションにリンクする必要があります。
- 1 つのアクティブなトランスフォーメーション、またはアクティブなトランスフォーメーションの 1 つの出力グループから、別のトランスフォーメーションの入力グループにリンクできます。
- アクティブなトランスフォーメーションとパッシブなトランスフォーメーションを同じ後続トランスフォーメーションまたはトランスフォーメーション入力グループに接続することはできません。
- 2 つ以上のアクティブなトランスフォーメーションを同じ後続トランスフォーメーションまたはトランスフォーメーション入力グループに接続することはできません。
- 任意の数のパッシブなトランスフォーメーションは、同じ後続トランスフォーメーション、トランスフォーメーション入力グループ、またはターゲットに接続できます。

- 両方の出力グループのデータがソートされている場合、同一のトランスフォーメーション内の 2 つの出力グループから、ソート済みデータに設定されている 1 つのジョイナトランスフォーメーションにポートをリンクできます。
- 互換性のあるデータ型を持つポートのみをリンクできます。Developer ツールは 2 つのデータ型間でマッピングが可能であることを検査してから、リンクします。Data Integration Service は、データ型に互換性がないポート間ではデータを変換することはできません。
- マッピングがデータフロー検証に違反する場合、Developer ツールによって無効とマークされます。

ポート属性のプロパゲート

変更された属性をマッピング全体のポートに渡すには、ポート属性をプロパゲートします。

1. エディタでトランスフォーメーションのポートを選択します。
2. **【マッピング】 > 【属性のプロパゲート】** をクリックします。
【属性のプロパゲート】 ダイアログボックスが表示されます。
3. 属性をプロパゲートする方向を選択します。
4. プロパゲートする属性を選択します。
5. 必要に応じて、結果をプレビューします。
6. **【適用】** をクリックします。
ポート属性がプロパゲートされます。

依存関係のタイプ

ポート属性をプロパゲートすると、Developer ツールは依存関係を更新します。

Developer ツールは以下の依存関係を更新できます。

- リンクパスの依存関係
- 暗黙の依存関係

リンクパスの依存関係

リンクパスの依存関係とは、プロパゲートされたポートおよびリンクパス内のポートの間の依存関係です。

リンクパスの依存関係をプロパゲートする to、Developer ツールは、前方リンクパスのすべての入力および入出力ポート、後方リンクパスのすべての出力および入出力ポートを更新します。Developer ツールは以下の更新を実行します。

- プロパゲートされたポートのリンクパスにおけるすべてのポートで、ポート名、データ型、精度、位取り、および説明を更新します。
- 変更されたポート名を持つプロパゲートされたポートを参照するすべての式または条件を更新します。
- 関連ポート名が変更された場合、動的 Lookup トランスフォーメーション内の関連ポートのプロパティを更新します。

暗黙の依存関係

暗黙の依存関係とは、式または条件に基づく 2 つのポートの間のトランスフォーメーション内にある依存関係です。

暗黙の依存関係を持つポートに対して、データ型、精度、位取り、および説明をプロパゲートできます。条件と式を解析して、プロパゲートされるポートのデータ型の暗黙の依存関係を特定できます。暗黙の依存関係を持つポートはすべて、出力または入出力ポートです。

条件を含めると、Developer ツールは以下の依存関係を更新します。

- リンクパスの依存関係
- プロパゲートされたポートと同じルックアップ条件で使用される出力ポート
- プロパゲートされたポートに関連付けられた、動的ルックアップトランスフォーメーションの関連ポート
- 明細ポートと同じ結合条件で使用されるマスタポート

式を含めると、Developer ツールは以下の依存関係を更新します。

- リンクパスの依存関係
- プロパゲートされたポートを使用する式を含む出力ポート

Developer ツールは、同一のトランスフォーメーション内の暗黙の依存関係にはプロパゲートしません。変更した属性は別のトランスフォーメーションからプロパゲートする必要があります。例えば、ルックアップ条件で使用されるポートのデータ型を変更して、その変更をルックアップトランスフォーメーションからプロパゲートすると、Developer ツールは、同じルックアップトランスフォーメーション内の条件に依存する他のポートにその変更をプロパゲートしません。

トランスフォーメーションでプロパゲートされるポート属性

Developer ツールは、各トランスフォーメーションの依存関係と属性をプロパゲートします。

以下の表に、Developer ツールが各トランスフォーメーションに対してプロパゲートする依存関係および属性を示します。

トランスフォーメーション	依存関係	プロパゲートされる属性
アドレスバリデータ	なし。	None このトランスフォーメーションは定義済みのポート名とデータ型を持ちます。
アグリゲータ	<ul style="list-style-type: none">- リンクパス内のポート- 式- 暗黙の依存性	<ul style="list-style-type: none">- ポート名、データ型、精度、位取り、説明- ポート名- データ型、精度、スケール
関連付け	<ul style="list-style-type: none">- リンクパス内のポート	<ul style="list-style-type: none">- ポート名、データ型、精度、位取り、説明
大文字小文字変換プログラム	<ul style="list-style-type: none">- リンクパス内のポート	<ul style="list-style-type: none">- ポート名、データ型、精度、位取り、説明
分類子	<ul style="list-style-type: none">- リンクパス内のポート	<ul style="list-style-type: none">- ポート名、データ型、精度、位取り、説明
比較	<ul style="list-style-type: none">- リンクパス内のポート	<ul style="list-style-type: none">- ポート名、データ型、精度、位取り、説明
コンソリデータ	なし。	None このトランスフォーメーションは定義済みのポート名とデータ型を持ちます。

トランスフォーメーション	依存関係	プロパゲートされる属性
データマスキング	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
データプロセッサ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ディシジョン	- リンクパスのダウストリームポート	- ポート名、データ型、精度、位取り、説明
式	- リンクパス内のポート - 式 - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - ポート名 - データ型、精度、スケール
フィルタ	- リンクパス内のポート - 条件	- ポート名、データ型、精度、位取り、説明 - ポート名
階層型からリレーショナル	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ジョイナ	- リンクパス内のポート - 条件 - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - ポート名 - データ型、精度、スケール
キージェネレータ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ラベラ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ルックアップ	- リンクパス内のポート - 条件 - 関連ポート（動的ルックアップ） - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - ポート名 - ポート名 - データ型、精度、スケール
一致	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
マージ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ノーマライザ	- リンクパス内のポート	- ポート名
パーサー	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ランク	- リンクパス内のポート - 式 - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - ポート名 - データ型、精度、スケール
読み取り		
REST Web サービス コンシューマ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
ルータ	- リンクパス内のポート - 条件	- ポート名、データ型、精度、位取り、説明 - ポート名
シーケンスジェネレータ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明

トランスフォーメーション	依存関係	プロパゲートされる属性
ソータ	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
SQL	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
標準化	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
共有体	- リンクパス内のポート - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - データ型、精度、スケール
アップデートストラテジ	- リンクパス内のポート - 式 - 暗黙の依存性	- ポート名、データ型、精度、位取り、説明 - ポート名 - データ型、精度、スケール
加重平均	- リンクパス内のポート	- ポート名、データ型、精度、位取り、説明
書き込み		

マッピングランタイムプロパティ

マッピングランタイムプロパティはマッピングに選択した実行環境により異なります。

以下のマッピングランタイムプロパティを設定します。

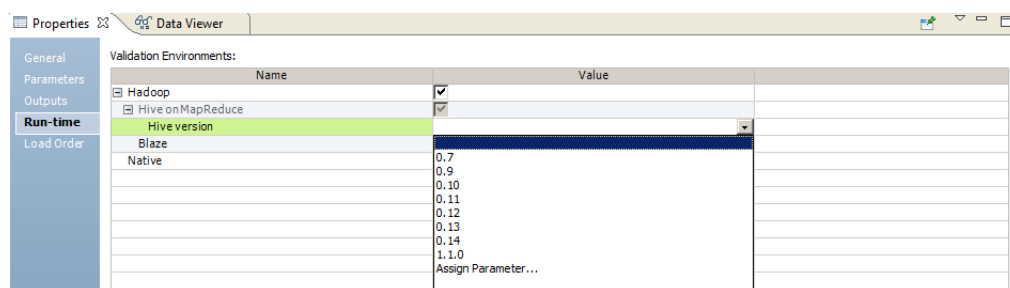
検証環境

Developer tool がネイティブ実行環境、Hadoop 実行環境、または両環境のいずれのマッピング定義を確認するかを示します。ネイティブ環境でマッピングを実行する場合は、データ統合サービスがマッピングを処理します。

ライセンスに基づいて Hadoop 環境でマッピングを行うことができます。Hadoop 環境でマッピングを行う場合、データ統合サービスは Hadoop 接続を使用してマッピング実行を Hadoop クラスタにプッシュします。Hadoop クラスタがマッピングを処理します。

Hadoop 実行環境を選択すると、MapReduce 上の Hive エンジンまたは Blaze エンジンを選択してマッピングを処理できます。Hive バージョンも選択できます。リストからバージョン番号を選択するか、パラメータを Hive バージョンに割り当てます。パラメータを Hive バージョンプロパティに割り当てるには、**[パラメータ]** ビューでパラメータを定義する必要があります。パラメータは Hive バージョンリストのバージョンを含む文字列である必要があります。

次の画像は、検証環境を示しています。



Hadoop 環境でマッピングを実行する前にネイティブ環境でマッピングのテストを行う場合は、両方の検証環境を選択します。また、マッピング実行時にパラメータの実行環境値を定義する場合も、両方の検証環境を選択します。

両方の環境を選択する場合は、[ランタイム] プロパティでマッピングの実行環境を選択する必要があります。

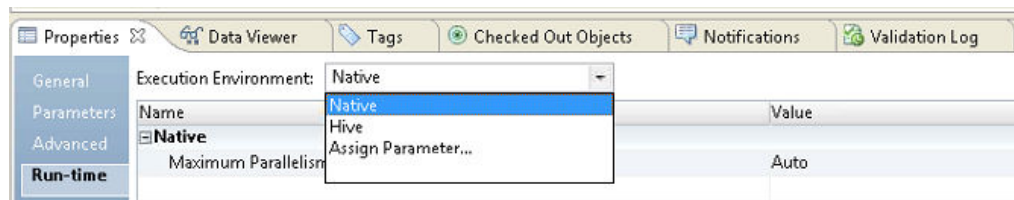
デフォルトはネイティブです。

実行環境

マッピング実行時に使用する実行環境を選択します。ネイティブ環境でマッピングを実行する場合は、データ統合サービスがマッピングを処理します。PowerCenter Big Data Edition をインストールした場合は、マッピングを Hive 環境で実行できます。Hive 環境でマッピングを実行する場合、データ統合サービスはトランスフォーメーションロジックを Hive 接続を介して Hadoop クラスタにプッシュします。Hadoop クラスタがデータを処理します。

マッピングパラメータを使用して実行環境を示すことができます。文字列パラメータを設定します。デフォルト値は [ネイティブ] または [Hive] に設定します。実行環境を選択する場合、[パラメータの割り当て] をクリックして、設定したパラメータを選択します。

以下の図に、マッピング実行環境を選択する場所を示します。



実行環境を選択する場合、Developer tool はマッピングの実行に関連付けられた検証環境のいずれかを保存します。

PowerCenter Express では、ネイティブの実行環境を使用できます。

最大並行処理

ネイティブの実行環境に対して有効です。単一のマッピングパイプラインステージを処理する並行スレッドの最大数。管理者は、マッピングのパーティション化を有効にするため、データ統合サービスの最大並行処理を 2 以上の値に設定します。管理者は、Administrator ツールで最大並行処理を設定します。

マッピングのデフォルトの最大並行処理値は、[自動] です。各マッピングは、データ統合サービスに定義されている最大並行処理値を使用します。デフォルトの最大並行処理値を変更し、特定のマッピングの最大値を定義できます。データ統合サービスとマッピングで設定されている最大並行処理の整数値が異なる場合、データ統合サービスでは最小値が使用されます。

PowerCenter Express はこのオプションに対応していません。最大並行処理の値を入力すると、データ統合サービスはそれを無視します。

デフォルトは [自動] です。最大値は 64 です。

パーティション化の詳細については、[「パーティション化されたマッピングの概要」](#) (ページ 249) を参照してください。

ターゲットコミット間隔

コミットのベースとして使用する行数。データ統合サービスは、処理するターゲット行数およびターゲットテーブルに対する制約に基づいてデータをコミットします。データ統合サービスはコミット間隔をチューニングします。デフォルトのコミット間隔は 10,000 行です。

コミット間隔はデータ統合サービスがコミットを発行するおおよその間隔です。データ統合サービスはコミット間隔より前、間隔通り、またはコミット間隔より後にコミットを発行することがあります。通常、データ統合サービスはライタバッファブロック全体を書き込んだ後にターゲットコミット間隔をチェックします。

エラー時の停止

読み取り、書き込み、またはトランスフォーメーションのスレッドで致命的でないエラーが発生した場合にマッピングを停止します。デフォルトでは無効になっています。

エラー時の停止を有効にすると、次のタイプのエラーによりマッピングが停止します。

読み取りエラー

ソースデータベースまたはソースファイルの読み取り中にデータ統合サービスで発生するエラー。読み取りエラーには Unicode モードでセッションを実行中のアライメントエラーが含まれます。

書き込みエラー

ターゲットデータベースまたはターゲットファイルへの書き込み中データ統合サービスに発生するエラー。書き込みエラーにはキー制約違反、非 NULL フィールドへの NULL のロード、およびデータベーストリガ応答が含まれます。

トランスフォーメーションエラー

データを変換中にデータ統合サービスに発生するエラー。トランスフォーメーションエラーには、変換エラーおよび NULL 入力などの ERROR として設定されたすべての条件が含まれます。

偽装ユーザー名のマッピング

ネイティブの実行環境に対して有効です。別のユーザーがデータ統合サービスのユーザーを偽装して、Kerberos 認証を使用する Hive、HBase、または HDFS のソースとターゲットに接続できるようにします。

ユーザー名を <Hadoop service name>/<hostname>@<YOUR-REALM> の形式で入力します。

説明：

- Hadoop service name は、Hive、HBase、または HDFS のソースとターゲットが存在する Hadoop サービスの名前です。
- hostname は、Hadoop サービスの名前または IP アドレスです。
- YOUR-REALM は、Kerberos レalm です。

特殊文字の「/」と「@」は、区切り文字としてのみ使用できます。

提案された並行処理

最大並行処理プロパティが 1 より大きな値またはパラメータに割り当てられている場合にネイティブ実行環境に対して有効です。トランスフォーメーションパイプラインステージを処理する並行スレッドの推奨数。推奨並行処理値を特定のトランスフォーメーションに定義すると、データ統合サービスによりそのトランスフォーメーションパイプラインステージの最適なスレッド数を判断するときにこの値が考慮されます。多数のポートが含まれるトランスフォーメーションや複雑な計算を実行するトランスフォーメーションでは、パフォーマンスを最適化するために推奨並行処理値を定義することをお勧めします。

PowerCenter Express はこのオプションに対応していません。推奨並行処理に値を入力すると、データ統合サービスはそれを無視します。

デフォルトは [自動] で、トランスフォーメーションはマッピングに定義された最大並行処理数を使用します。最大値は 64 です。

推奨並行処理値の定義の詳細については、[「トランスフォーメーションのための提案された並行処理」](#) (ページ 267) を参照してください。

Hive 接続

Hive 実行環境に対して有効です。Hive 接続は、データ統合サービスがマッピングの実行を Hadoop クラスタにプッシュするために必要な接続情報を定義します。Hive 接続を選択し、Hadoop クラスタでマッピングを実行します。Hive 接続にユーザー定義のパラメータを割り当てることができます。パラメータの定義は、マッピングの【パラメータ】ビューで行います。

ターゲットロード順の制約

ターゲットのロード順序の制約は同じマッピングで互いに関係している 2 つのターゲットインスタンスに対して、データ統合サービスが行をロードおよびコミットする方法を制限します。

Developer tool で、制約を設定しデータ統合サービスが行をターゲットテーブルにロードする順序を制限できます。

データ統合サービスにプライマリターゲットインスタンスのデータを完全にロードした後に、セカンダリターゲットインスタンスにデータをロードさせるよう制約を設定することができます。プライマリターゲットとセカンダリターゲットとして定義するテーブルは入力行のトランザクションによって異なります。

ターゲットロード順の制約について次のシナリオを考慮します。

マスタおよび詳細ターゲットへの行の挿入

プライマリキーと外部キーのリレーションがあるターゲットに行を挿入する場合、ターゲットのロード順序の制約を設定することがあります。プライマリキーを持つターゲットをプライマリターゲットインスタンスとして設定します。外部キーを持つターゲットをセカンダリターゲットインスタンスとして設定します。データ統合サービスは、プライマリターゲットのロードが完了するまで、セカンダリターゲットのデータをステージングできません。

マスタおよび詳細ターゲットからの行の削除

プライマリキーと外部キーのリレーションがあるターゲットから行を削除する必要がある場合は、異なる制約を設定します。外部キーを持つターゲットをプライマリターゲットインスタンスとして設定し、詳細ターゲットから先に行を削除します。プライマリキーを持つターゲットをセカンダリターゲットインスタンスとして設定します。

同じリレーショナルテーブルへの行の挿入と行の更新

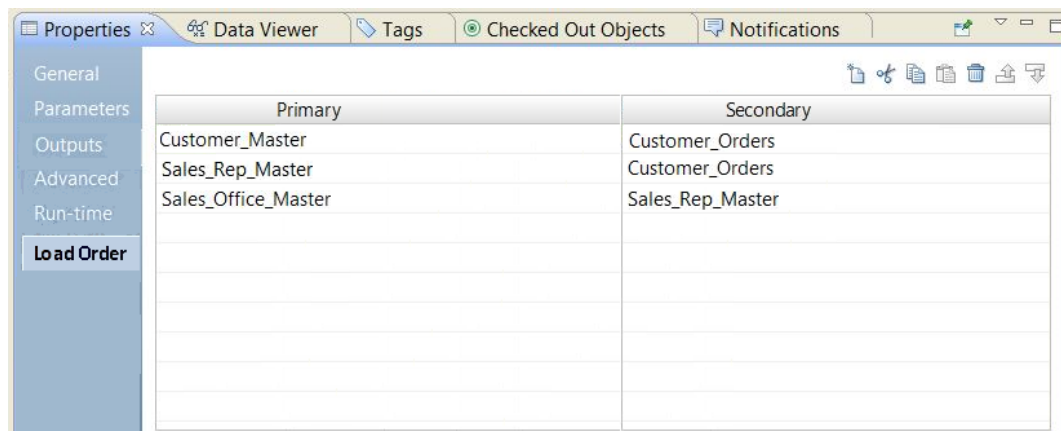
2 つの別のトランスフォーメーションからリレーショナルテーブルに行の挿入と行の更新をロードするマッピングに対して、ターゲットのロード順序の制約を設定できます。データ統合サービスで行の挿入のロードが完了するまで行の更新のロードが制限されるよう制約を設定します。

1 つのマッピングに複数の制約を設定できます。データ統合サービスは制約を違反せずにターゲットをロードする最も効率的な実行計画を決定します。

ターゲットの【ロード順序】タブ

マッピングの【ロード順序】タブで、ターゲットロード順の制約を定義します。マッピングキャンバスでクリックして、【ロード順序】タブにアクセスします。

次の画像は、【ロード順序】タブを示しています。



この画像には3つの制約が示されています。各制約には、プライマリターゲットおよびセカンダリターゲットが含まれています。これらの制約は、データ統合サービスが Customer_Orders ターゲットの前に Customer_Master ターゲットをロードする必要があることを指定しています。Sales_Rep_Master は Customer_Orders ターゲットの前にロードする必要があります。Sales_Office_Master は Sales_Rep_Master の前にロードする必要があります。

制約は任意の順序で入力できます。次に示す制約のペアを入力しても、前の画像と同じ制約を指定することができます。

プライマリ	セカンダリ
Sales_Office_Master	Sales_Rep_Master
Sales_Rep_Master	Customer_Master
Customer_Master	Customer_Orders

挿入行と削除行に関する制約

ターゲットロード順序の制約には、同じファイルの挿入行、更新行、および削除行の特殊な処理は含まれていません。

挿入行、更新行、および削除行を処理する必要がある場合、削除行とは異なるターゲットインスタンスに挿入行および更新行を返すようにルータートランスフォーメーションを設定できます。ターゲットロード順序の制約を設定して、ターゲットをロードする順序を指定します。

例えば、Order_Header および Order_Detail というターゲットがあるとします。Order_Detail テーブルには、Order_Header テーブルへの OrderID 外部キーがあります。両方のテーブルの挿入、更新、および削除を処理する必要があります。

ルータートランスフォーメーションを使用して、挿入行と更新行を削除行から切り離すことができます。ルータートランスフォーメーションからの次の出力グループを設定します。

1. Order_Header の挿入行および更新行
2. Order_Header の削除行
3. Order_Detail の挿入行および更新行
4. Order_Detail の削除行

これらの行をターゲットにロードする場合、次の制約を作成できます。

```
Group #4 before group #2
Group #2 before group #1
Group #1 before group #3
```

これらの制約により、データ統合サービスは、Order_Header の削除より前に Order_Detail の削除を処理します。データ統合サービスは、挿入および更新の前にすべての削除を処理します。また、Order_Detail の挿入および更新の前に Order_Header の挿入および更新を処理します。

ターゲットのロード順に関するルールとガイドライン

ターゲットロード順の制約を定義する場合は、以下のルールおよびガイドラインに従います。

- Developer tool で、一部のターゲットカラムをプライマリーまたは外部キーとして設定できます。ロード順序の制約はこれらのキーを無視します。ターゲットにプライマリーと外部キーの制約がある場合は、ロード順序の制約を定義する必要があります。
- Developer tool は定義されたロード順序の制約を検証しません。Developer tool はマッピングを検証するときにロード順序の制約を検証します。
- データ統合サービスは、ターゲットロード順の制約のセカンダリターゲットインスタンスのデータをローカルディスクにステージングできます。マッピングに複数のセカンダリターゲットインスタンスがある場合、データ統合サービスは制約に違反することなく、ステージングされたデータをターゲットにロードします。
- 一時ファイルまたはファイル内のデータの使用を避けることを強くお勧めします。ステージングファイルやテーブル内のデータをユーザーが変更したことによりデータが破損しても、Informatica 社では責任を負いかねます。ステージングファイルの構造は、Informatica のバージョンによって異なる場合があります。
- データ統合サービスは、行が挿入、削除、または更新のいずれであるかを判別することなく、1つのターゲットインスタンスに続いて別のターゲットインスタンスをロードします。プライマリーと外部キーの制約があるターゲットテーブルの場合、孤立行はプライマリーターゲットに一致する行がない外部キーターゲットの行です。データ統合サービスは孤立行をチェックしません。データ統合サービスはロード順序の制約に指定された順序ですべての行をロードします。

制約の作成

マッピングの【ロード順序】タブで、ターゲットロード順序の制約を作成します。

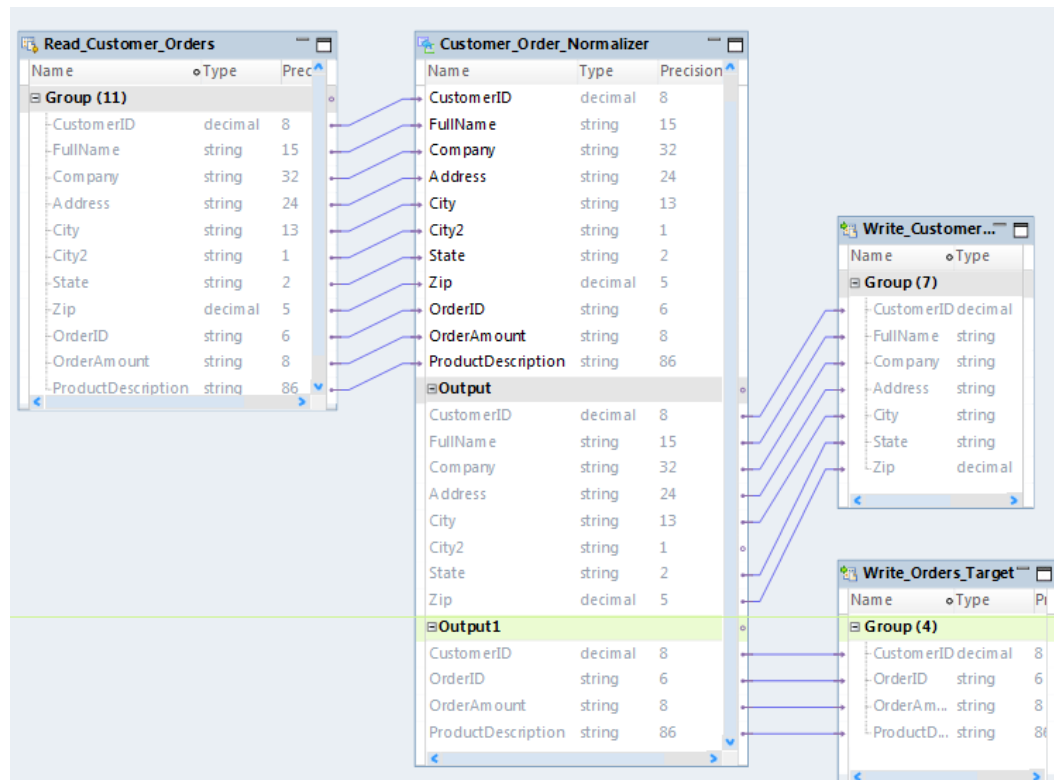
1. 複数のターゲットロードを含むマッピングを作成します。
2. マッピングキャンバスをクリックします。
マッピングの【プロパティ】タブが表示されます。
3. 【ロード順序】タブをクリックします。
4. 制約を入力するには、【新規】をクリックします。
Developer tool によって、制約の行が作成されます。
5. 【プライマリ】フィールドをクリックします。
マッピングのターゲットインスタンスのリストが表示されます。
6. 最初にロードするターゲットインスタンスを選択します。
7. 【セカンダリ】フィールドで、2番目にロードするターゲットインスタンスを選択します。
8. 必要な数だけ制約を入力します。

ターゲットのロード順の例

組織は顧客注文を 1 日に 2 回処理します。同じトランザクションファイルで顧客情報と注文情報を受信します。組織は注文ファイル进行处理するマッピングに、注文をロードする前に顧客情報をロードすることを確実に実行させる必要があります。

開発者は顧客情報を Customer_Target テーブルに返すマッピングを作成します。マッピングは注文を Orders_Target テーブルに返します。Customer_Master のプライマリキーは CustomerID です。注文データの各注文には Customer_Master の CustomerID への外部キーがあります。開発者はターゲットロード順の制約を作成します。制約は、顧客情報のターゲットへのロードが完了するまで、データ統合サービスが注文をロードするのを制限します。

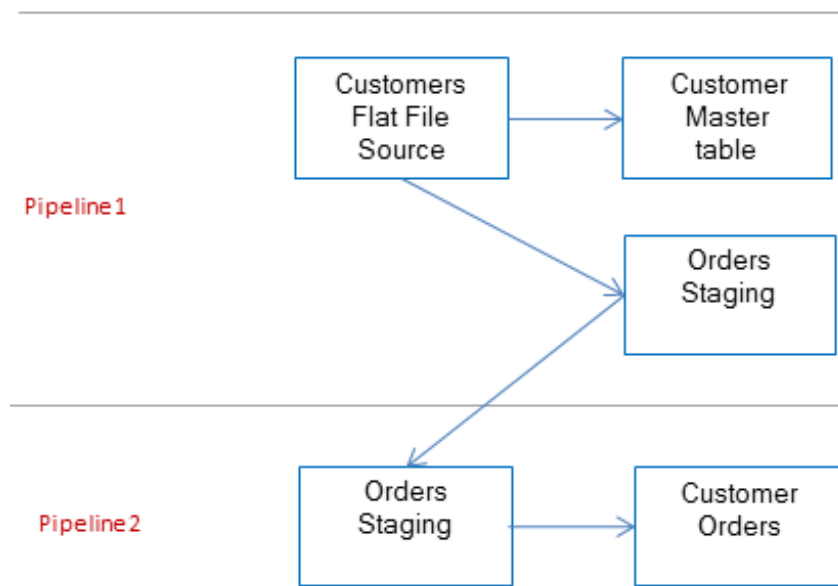
次の画像は、マッピングを示しています。



ノーマライズトランスフォーメーションは、顧客および注文データの個別の出力グループを作成します。開発者は注文データより前に顧客データのターゲットへのロードが確実に行われるようにする必要があります。

データ統合サービスでは、別のストラテジを使用してターゲットロード順の制約を実現できます。次の例では、データ統合サービスは 2 つのパイプラインを作成して、顧客データと注文データをターゲットテーブルにロードします。

次の図に、顧客データと注文データをターゲットテーブルにロードするパイプラインを示します。



最初のパイプラインで、データ統合サービスは顧客を Customer_Master にロードして、注文をローカルディスクファイルにステージングします。2 番目のパイプラインでは、ステージングされた注文を注文テーブルにロードします。

マッピングの検証

マッピングを開発する場合、Data Integration Service がマッピング全体を読み込み、そして処理できるようにマッピングを設定する必要があります。Data Integration Service でのマッピングの実行を妨げるエラーが Developer ツールで検出された場合、そのマッピングは無効とマークされます。

Developer ツールは次のタイプの検証を考慮します。

- 接続
- 式
- オブジェクト
- データフロー

接続の検証

マッピング内のポートの接続、マッピングの検証を行うたびに、Developer ツールによって接続の検証が行われます。

ポートを接続すると、Developer ツールは有効な接続かどうかを検証します。マッピングを検証すると、Developer ツールは接続が有効かどうか、必要なポートがすべて接続されているかどうかを検証します。Developer ツールは以下の接続の検証を実行します。

- 少なくとも 1 つの入力オブジェクトおよび出力オブジェクトが接続されている。
- 少なくとも 1 つのマプレット入力ポートおよび出力ポートがマッピングに接続されている。

- ポート間のデータ型に互換性がある。ポートのデータ型を、接続先のポートと互換性のないものに変更すると、Developer ツールによってエラーが生成されてマッピングが無効になります。しかし、Char と Varchar のように接続したポートとの互換性が維持される場合は、データ型を変更することができます。

式の検証

マッピングを作成するときに、トランスフォーメーション内の式を検証することができます。エラーを修正しなかった場合、マッピングを検証すると **【検証ログ】** ビューにエラーメッセージが表示されます。

式で使用した入力ポートを削除すると、Developer ツールはマッピングを無効とマークします。

オブジェクトの検証

マッピングを検証すると、Developer ツールは入力トランスフォーメーションやマプレットなど独立したオブジェクトの定義がマッピング内のインスタンスと一致するかどうかを検証します。

マッピングを設定する際にオブジェクトを 1 つでも変更すると、マッピングにエラーが含まれる場合があります。マッピングの設定中ではないときにオブジェクトのいずれかが変更された場合、Developer ツールは、その変更がマッピングに与える影響について追跡します。

マッピングの検証

マッピングを検証して、Data Integration Service がマッピング全体を読み取りおよび処理できることを確認します。

1. **【編集】** > **【検証】** をクリックします。
エラーは **【検証ログ】** ビューに表示されます。
2. エラーを修正し、マッピングを再度検証します。

マッピングの実行

マッピングを実行し、出力をソースからターゲットに移動してデータを変換します。

デフォルトのデータ統合サービスを選択していない場合、選択するように求められます。

- ▶ エディタの空の領域を右クリックし、**【マッピングの実行】** をクリックします。
データ統合サービスによって、マッピングが実行され、出力がターゲットに書き込まれます。
オペレーティングシステムのプロファイルを使用するようにデータ統合サービスが設定されている場合、そのオペレーティングシステムのプロファイルを使用してマッピングを実行します。

セグメント

セグメントは、マッピング、マプレット、ルール、または仮想ストアドプロシージャ内の 1 つ以上のオブジェクトで構成されます。セグメントには、ソース、ターゲット、トランスフォーメーション、またはマプレットを含めることができます。セグメントはマッピング、マプレット、またはルール内の 1 つ以上のオブジ

エクトで構成されます。セグメントには、ソース、ターゲット、トランスフォーメーション、またはマップレットを含めることができます。

セグメントはコピーできます。セグメントをコピーする場合、以下のルールとガイドラインを考慮します。

- フォルダーまたはプロジェクト間でセグメントをコピーできます。
- Developer ツールは、可能な場合には依存関係を再利用します。再利用できない場合、依存関係をコピーします。
- マッピング、マップレット、ルール、または仮想ストアドプロシージャにパラメータが含まれ、そのパラメータを参照するトランスフォーメーションをコピーした場合、コピー先のオブジェクトのトランスフォーメーションではパラメータのデフォルト値が使用されます。
- マッピング、マップレット、またはルールにパラメータが含まれ、そのパラメータを参照するトランスフォーメーションをコピーした場合、ターゲットオブジェクトのトランスフォーメーションではパラメータのデフォルト値が使用されます。
- 入力トランスフォーメーションおよび出力トランスフォーメーションはコピーできません。
- セグメントをコピーした後、前のアクションを取り消すことはできません。

マッピングまたはマップレットのセグメントからマップレットを生成できます。マッピングまたはマップレットに、再使用する予定の接続されたトランスフォーメーションフローが含まれている場合に、マップレットを生成する場合があります。

セグメントのコピー

別のマッピング、マップレット、ルール、または仮想ストアドプロシージャのマッピングロジックの一部を再利用する場合、セグメントをコピーできます。別のマッピング、マップレット、またはルールのマッピングロジックの一部を再利用する場合、セグメントをコピーできます。

1. コピーするセグメントを含むオブジェクトを開きます。
2. コピーする各オブジェクトをハイライト表示して、セグメントを選択します。
複数のオブジェクトを選択するには Ctrl キーを押します。また、エディタでマウスをドラッグしてオブジェクトを囲んでもセグメントを選択できます。
3. **【編集】 > 【コピー】** をクリックして、セグメントをクリップボードにコピーします。
4. コピー先のマッピング、マップレット、ルール、または仮想ストアドプロシージャを開きます。
5. コピー先のマッピング、マップレット、またはルールを開きます。
6. **【編集】 > 【貼り付け】** をクリックします。

第 2 章

マプレット

この章では、以下の項目について説明します。

- [マプレットの概要, 39 ページ](#)
- [マプレットのタイプ, 41 ページ](#)
- [マプレットの入力と出力, 41 ページ](#)
- [生成されたマプレット, 42 ページ](#)
- [マプレットの作成, 44 ページ](#)
- [マプレット検証, 44 ページ](#)

マプレットの概要

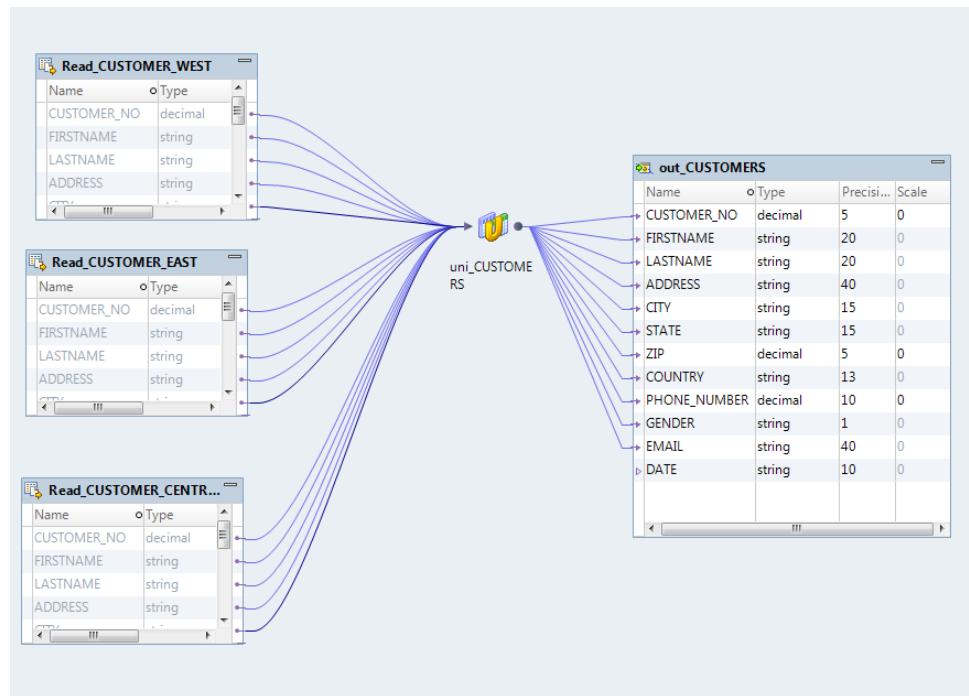
マプレットは、複数のマッピングで使える一連のトランスフォーメーションを含む再利用可能オブジェクトです。マプレットはマッピングで使用するか、またはルールとして検証します。

マプレットのトランスフォーメーションには、再利用できるものと再利用できないものがあります。シーケンスジェネレータトランスフォーメーションをマプレットに追加する場合、それは再利用可能である必要があります。

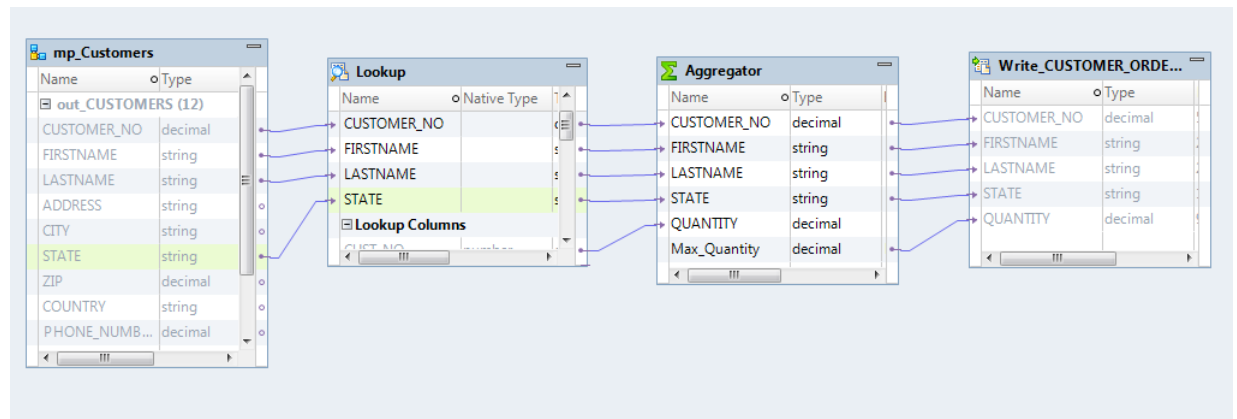
マッピング内でマプレットを使用する際には、そのインスタンスを用います。マプレットに施した変更はそのインスタンスすべてに引き継がれます。

マプレットには、他のマプレットを含めることができます。また、1つのマッピングまたはマプレットで、マプレットを複数回使用することもできます。マプレットの循環ネストを行うことはできません。例えば、マプレット A にマプレット B が含まれている場合、マプレット B にマプレット A を含めることはできません。

例えば、以下の図に共有体トランスフォーメーションを利用して東部、西部、中央部から顧客情報を結合するマップレットを示します。



以下の図に、顧客マップレットを再利用して州ごとの一番多く品物を購入した顧客を見つけることができるマッピングを示します。



マップレットは手動で作成できます。マッピングまたはマップレット内のセグメントからマップレットを生成することもできます。

マプレットのタイプ

マプレットのタイプは、マプレットの入力と出力によって決定されます。

以下のタイプのマプレットを作成または生成できます。

- ソース。マプレットには、入力としてデータソースが含まれ、出力として出力トランスフォーメーションが含まれます。
- ターゲット。マプレットには、入力として入力トランスフォーメーションが含まれ、出力としてデータソースが含まれます。
- 中間ストリーム。マプレットには、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が含まれます。入力または出力用のデータソースは含まれません。

マプレットの入力と出力

マッピング内でマプレットを使用するためには、マプレットの入力と出力を設定する必要があります。

マプレットは、以下の入力と出力の要素で構成されます。

- マプレットの入力。データソースや入力トランスフォーメーション、またはその両方を使用して、マプレットにデータを渡すことができます。マプレットをルールとして検証する場合は、入力トランスフォーメーションを使用してマプレットにデータを渡す必要があります。入力トランスフォーメーションを使用する場合は、マッピング内のソースまたはアップストリームトランスフォーメーションに接続します。
- マプレットの出力。データソースや出力トランスフォーメーション、またはその両方を使用して、マプレットからデータを渡すことができます。マプレットをルールとして検証する場合は、出力トランスフォーメーションを使用してマプレットからデータを渡す必要があります。出力トランスフォーメーションを使用する場合は、マッピング内のターゲットまたはダウンストリームトランスフォーメーションに接続します。
- マプレットポート。マッピングエディタでマプレットポートを表示できます。マプレット入力ポートおよび出力ポートは、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から生成されます。データソースからは生成されません。

マプレットの入力

マプレットは、データソースまたは入力トランスフォーメーションを入力元とすることができます。

1つのマプレット内に複数のパイプラインを作成することができます。複数のデータソースまたは入力トランスフォーメーションを使用します。また、データソースと入力トランスフォーメーションを組み合わせで使用することもできます。

1つ以上のデータソースを使用して、マプレットにソースデータを提供します。マッピング内でマプレットを使用する場合、マプレットはマッピングパイプラインにおける最初のオブジェクトとなり、入力ポートを含みません。

入力トランスフォーメーションを使用して、マッピングから入力を受け取ります。入力トランスフォーメーションは入力ポートを提供し、これによりマプレット内でデータを受け渡すことが可能になります。マプレット内の別のトランスフォーメーションに接続した Input トランスフォーメーションの各ポートがマプレットの入力ポートとなります。Input トランスフォーメーションは、1つのアクティブソースからのデータを受け取ることができます。接続されていないポートはマッピングエディタに表示されません。

入力トランスフォーメーションはマプレット内の複数のトランスフォーメーションに接続することができます。入力トランスフォーメーションの1つのポートをマプレット内の複数のトランスフォーメーションに接続することもできます。

マップレットの出力

ターゲットマップレットを作成するときは、データソースを出力として使用します。マップレット内の出力トランスフォーメーションは、マップレットを介してマッピングにデータを渡すために使用します。

1つ以上のデータソースを使用して、マップレットにターゲットデータを提供します。マッピング内でマップレットを使用する場合、マップレットはマッピングパイプラインにおける最後のオブジェクトとなり、出力ポートを含みません。

出力トランスフォーメーションを使用して、マッピング内のダウストリームトランスフォーメーションまたはターゲットに出力を渡します。出力トランスフォーメーション内で接続されている各ポートは、マッピング内でマップレットの出力ポートとして表示されます。マップレット内の各出力トランスフォーメーションは、出力グループとして表示されます。出力グループは、データをマッピング内の複数のパイプラインに渡すことができます。

生成されたマップレット

マッピングまたはマップレットのセグメントからマップレットを生成できます。マッピングまたはマップレットに、再使用する予定の接続されたトランスフォーメーションフローが含まれている場合に、マップレットを生成する場合があります。

Developer tool は、生成プロセスの一部としてセグメントをマップレットとして検証します。検証エラーを回避するには、生成されたマップレットのルールおよびガイドラインを確認してください。

生成されたマップレットのルールおよびガイドライン

次の条件が当てはまる場合、マップレット生成が失敗します。

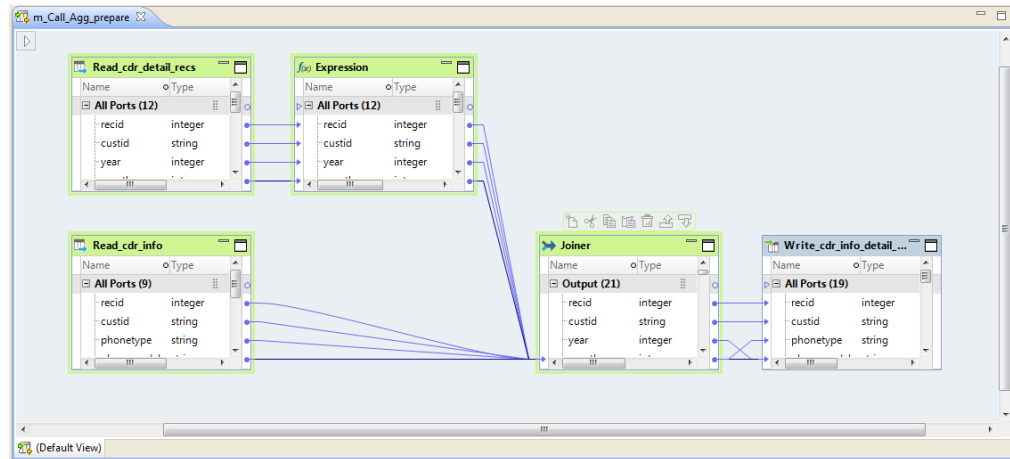
- 選択したトランスフォーメーションが連続していない。
- セグメントに読み取りトランスフォーメーションと書き込みトランスフォーメーションの両方が含まれている。ただし、セグメントに複数の読み取りトランスフォーメーションまたは複数の書き込みトランスフォーメーションが含まれていても問題ありません。
- セグメントに再利用不可能なシーケンスジェネレータトランスフォーメーション、入力トランスフォーメーション、出力トランスフォーメーション、または操作の設定を含むトランスフォーメーションが含まれている。
- 選択したセグメントにパイプラインブランチのすべてのトランスフォーメーションが含まれていない。
- セグメントの最初と最後のトランスフォーメーションに動的フィールドが含まれている。
- セグメントに最初のトランスフォーメーションへの入力ランタイムリンクまたは最後のトランスフォーメーションからの出力ランタイムリンクが含まれている。
- セグメントが1つのパラメータ化された読み取りトランスフォーメーション、書き込みトランスフォーメーション、またはルックアップトランスフォーメーションで構成されている。

マップレットの生成

接続されたトランスフォーメーションが含まれるセグメントからマップレットを生成します。セグメントに読み取りトランスフォーメーション、書き込みトランスフォーメーション、またはミッドストリームトランスフォーメーションが含まれる場合があります。

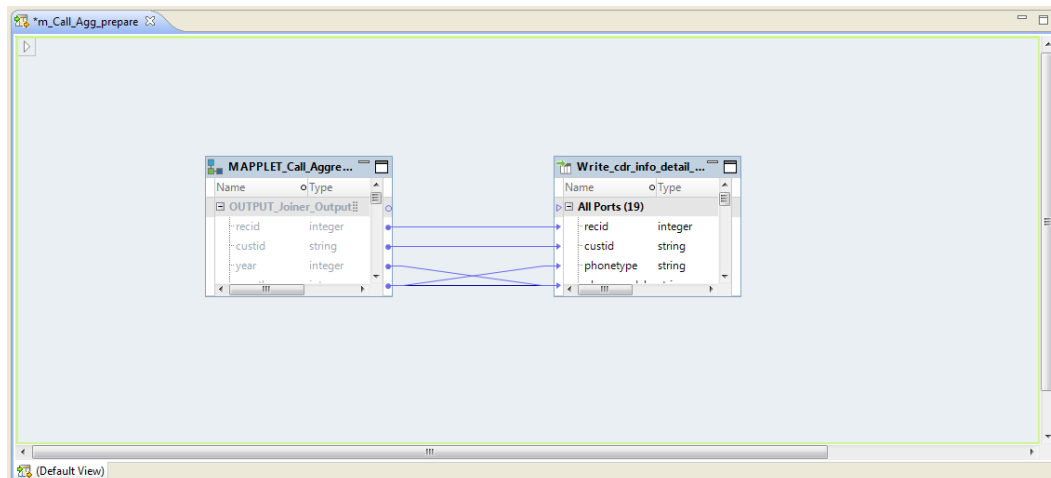
1. マップレットに生成するセグメントが含まれるマッピングまたはマップレットを開きます。

2. マップレットに組み込むトランスフォーメーションを選択します。
3. 選択したトランスフォーメーションのいずれかを右クリックし、**【マップレットの抽出】**を選択します。
次の図に、4つのトランスフォーメーションが選択されたマッピングを示します。



生成プロセスでセグメントが検証され、検証エラーがあれば報告されます。

4. マップレットを生成する **【マップレット】** ノードを参照します。
Developer tool は、デフォルトで現在のプロジェクト内の **【マップレット】** ノードにマップレットを生成します。
 5. **【完了】** をクリックします。
このマップレットは、元のマッピングまたはマップレットで選択されたトランスフォーメーションを置換します。このツールは、セグメントに読み取りトランスフォーメーション、書き込みトランスフォーメーション、またはミッドストリームトランスフォーメーションが含まれているかどうかに基づいて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をマップレットに追加します。
- 次の図に、選択したトランスフォーメーションがマップレットによって置換される様子を示します。



選択したトランスフォーメーションをマップレットで置換するには、変更されたマッピングまたはマップレットを明示的に保存する必要があります。マッピングまたはマップレットを元の状態に戻すには、**【ファイル】 > 【元に戻す】** を 3 回選択します。

マプレットの作成

マプレットを作成して、複数のマッピングで使用できる一連のトランスフォーメーションを含む再利用可能オブジェクトを定義します。

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】 > 【マプレット】** をクリックします。
3. マプレット名を入力します。
4. **【完了】** をクリックします。
空のマプレットがエディタに表示されます。
5. マプレットの入力、出力、およびトランスフォーメーションを追加します。

マプレット検証

マッピングに追加する前にマプレットを検証できます。プロファイルでルールと使用するマプレットを検証することもできます。

マプレットの検証

マプレットをマッピングに追加する前に検証します。マプレットをルールとして検証し、プロファイルに含めることもできます。

1. マプレットエディタを右クリックします。
2. **【検証】 > 【マプレット】** または **【検証】 > 【ルール】** を選択します。
検証ログに発生したエラーが表示されます。

ルール検証としてのマプレット

ルールは、プロファイルの実行時にソースデータに適用される条件を定義するビジネスロジックです。これは、プロファイルで使用する中間ストリームマプレットです。プロファイルでルールとして使用するマプレットを検証できます。

ルールは、以下の要件を満たす必要があります。

- ルールには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を含める必要があります。ルールでデータソースを使用することはできません。
- ルールには式トランスフォーメーション、ルックアップトランスフォーメーション、およびパッシブなデータ品質トランスフォーメーションを含めることができます。ルールには他のタイプのトランスフォーメーションを含めることはできません。例えば、一致トランスフォーメーションはアクティブなトランスフォーメーションなので、ルールに含めることはできません。
ルールには式トランスフォーメーションとルックアップトランスフォーメーションを含めることができます。ルールには他のタイプのトランスフォーメーションを含めることはできません。
- ルールでは、入力グループ間のカーディナリティを指定しません。

注: ルールの機能はプロファイリングに限定されません。Analyst ツールでルールとして検証したマプレットをプロファイルに追加できます。例えば、郵便アドレスを検証するように設定されているルールを選択してプロファイルに追加することで、郵便アドレスのデータ品質を評価できます。

第 3 章

マッピングパラメータ

この章では、以下の項目について説明します。

- [マッピングパラメータの概要, 45 ページ](#)
- [システムパラメータ, 46 ページ](#)
- [ユーザー定義のパラメータ, 47 ページ](#)
- [ユーザー定義のパラメータを作成する場所, 48 ページ](#)
- [パラメータを割り当てる場所, 49 ページ](#)
- [マッピングのパラメータ, 57 ページ](#)
- [マップレットのパラメータ, 59 ページ](#)
- [論理データオブジェクトのパラメータ, 61 ページ](#)
- [仮想テーブルマッピングのパラメータ, 62 ページ](#)
- [パラメータセット, 63 ページ](#)
- [パラメータファイル, 64 ページ](#)
- [パラメータの設定方法, 71 ページ](#)

マッピングパラメータの概要

マッピングパラメータとは、マッピングを実行するたびに変更できる定数値です。異なる値でマッピングを再実行できるように、パラメータを作成します。パラメータを使用して、接続、ファイルディレクトリ、式のコンポーネント、ポートリスト、ポートリンク、およびタスクプロパティの値を変更します。

システムパラメータまたはユーザー定義のパラメータを設定できます。

システムパラメータ。

データ統合サービスの組み込みパラメータ。システムパラメータは、データ統合サービスがログファイル、キャッシュファイル、拒否ファイル、ソースファイル、ターゲットファイル、一時ファイルを格納するディレクトリを定義します。管理者は、データ統合サービスのシステムパラメータのデフォルト値を Administrator ツールで定義します。

ユーザー定義のパラメータ。

トランスフォーメーション、論理データオブジェクト、マッピング、およびワークフロー内で定義するパラメータ。異なる接続、フラットファイル、キャッシュファイル、一時ファイル、式、ポート、または参照テーブルの値でマッピングを再実行できるように、ユーザー定義のパラメータを作成します。異なる式、接続、ファイル、ポート、またはリソースの値でマッピングを再実行できるように、ユーザー定義のパラメータを作成します。

パラメータを使用して、生成されたどのポートを、実行時に動的マッピングで使用するかを決定することができます。実行時にリンクするポートを示すように、パラメータを設定できます。読み取り、書き込み、またはルックアップトランスフォーメーションのデータオブジェクトを変更するように、パラメータを割り当てることができます。

パラメータセットまたはパラメータファイルをマッピングに割り当てることで、パラメータ値を上書きできます。パラメータセットは、マッピングパラメータの値が含まれるリポジトリオブジェクトです。パラメータファイルは、パラメータ値が含まれる XML ファイルです。パラメータセットまたはパラメータファイルを使用してマッピングを実行する場合、データ統合サービスはパラメータセットまたはパラメータファイルで定義されたパラメータ値を使用します。この値は、トランスフォーメーション、マッピング、マップレット、またはワークフローで設定されたデフォルトのパラメータ値を上書きします。

ワークフローパラメータの詳細については、『*Informatica Developer ワークフローガイド*』を参照してください。

関連項目：

- [「動的マッピングのパラメータ」 \(ページ 110\)](#)

システムパラメータ

システムパラメータは、データ統合サービスがキャッシュファイル、拒否ファイル、ソースファイル、ターゲットファイル、ログファイル、および一時ファイルを格納するディレクトリを定義する定数値です。

データ統合サービスの実行オプション内のいくつかのシステムパラメータの値を定義します。管理者は、Administrator ツールで値を更新できます。データ統合サービスは、他のシステムパラメータの値を実行時に決定します。パラメータファイルまたはパラメータセットのシステムパラメータ値を上書きすることはできません。

システムパラメータは作成できません。Developer ツールには定義済みのシステムパラメータのリストが用意されており、マッピングのデータオブジェクトまたはトランスフォーメーションに割り当てることができます。例えば、アグリゲータトランスフォーメーションを作成すると、キャッシュディレクトリのシステムパラメータがデフォルト値として、Informatica Administrator のキャッシュディレクトリフィールドに割り当てられます。異なるキャッシュディレクトリの場所を使用する場合は、ユーザー定義のパラメータを作成し、デフォルトのパラメータ値を設定します。

以下の表に、システムパラメータを示します。

システムパラメータ	タイプ	説明
CacheDir	String	インデックスファイルとデータキャッシュファイルのデフォルトディレクトリ。
LogDir	String	マッピングタスクのログファイルのデフォルトディレクトリ。
RejectDir	String	拒否ファイルのデフォルトディレクトリ。
SourceDir	String	ソースファイルのデフォルトディレクトリ。
TargetDir	String	ターゲットファイルのデフォルトディレクトリ。
TempDir	String	一時ファイルのデフォルトディレクトリ。

システムパラメータ	タイプ	説明
ApplicationName	String	アプリケーションの名前
ExecutionEnvironment	String	Hadoop またはネイティブ環境。
MappingName	String	実行中のマッピングの名前。
MappingRunStartTime	日付/時刻	実行中のマッピングの開始時刻。
ServiceName	String	データ統合サービス名。
UserName	String	マッピングを実行しているユーザーの名前。

ユーザー定義のパラメータ

ユーザー定義のパラメータは、マッピングの実行間で変更できる定数値を表します。

例えば、顧客の注文を処理するマッピングを作成するとします。このマッピングは、1つの国の顧客データを含むリレーショナルテーブルから顧客情報を読み取ります。米国、カナダ、メキシコの顧客に対してマッピングを使用するとします。この場合、顧客テーブルへの接続を表すユーザー定義のパラメータを作成します。米国の顧客テーブル、カナダの顧客テーブル、およびメキシコの顧客テーブルへの接続名を設定する3つのパラメータセットを作成します。マッピングのそれぞれの実行に異なるパラメータセットを使用して実行します。

以下の型パラメータを作成できます。

接続パラメータ

Informatica の接続名。

日付/時刻パラメータ

日付。

式

結合条件、フィルタ式、またはルックアップ条件を定義する式。

入力リンクセット

【ランタイムリンク】ダイアログボックス内の、リンクするポートのセット。

数値パラメータ

Integer、Bigint、Decimal、Double 型のパラメータ。

ポート

式内で参照するポートの名前。

ポートリスト

グループを含めるポートのリスト。例えば、ポートリストパラメータをアグリゲータトランスフォーマーシジョンやランクトランスフォーマーシジョン内で使用できます。

リソース

リレーショナルデータオブジェクトのテーブル、ビュー、またはシノニムの名前。リソース名をパラメータ化すると、データ統合サービスはそのパラメータ値をランタイムクエリ内で使用してオブジェクトを取得します。

ソートリスト

ソートトランスフォーメーションを使用してソートするポートのリスト。リストには、ポート名と、昇順または降順のソートシーケンスを示すインジケータが含まれます。

String

文字列パラメータは、フラットファイル名、ディレクトリ、テーブル名、またはランタイムプロパティを表します。文字列パラメータは、32768 文字以下の精度で定義します。

パラメータを作成する場合、パラメータ名の先頭にドル記号 (\$) を使用することはできません。

パラメータを使用してプロパティ値を設定する場合、プロパティに適切なパラメータのタイプを使用する必要があります。例えば、ターゲットファイル名に接続タイプパラメータは使用できません。数値式でパラメータを使用する場合は、数値タイプのパラメータを使用する必要があります。

リレーショナルデータオブジェクトでは、SQL オーバーライド、フィルタ条件、または結合条件でドル記号 (\$) をエスケープする必要はありません。データ統合サービスは、SQL 文におけるドル記号で始まるフィールドをパラメータとして扱います。

日付/時刻パラメータ

日付パラメータを作成して式内で使用できます。

日付パラメータは以下のいずれかの形式で定義する必要があります。

MM/DD/RR
MM/DD/YYYY
MM/DD/YYYY HH24:MI
MM/DD/RR HH24:MI
MM/DD/RR HH24:MI:SS
YYYY/MM/DD HH24:MI:SS
MM/DD/RR HH24:MI:SS.NS
MM/DD/YYYY HH24:MI:SS.NS

ユーザー定義のパラメータを作成する場所

ユーザー定義のパラメータは、フラットファイルデータオブジェクト、トランスフォーメーション、カスタムデータオブジェクト、マップレット、マッピング、およびワークフロー内に作成できます。パラメータを作成したら、条件、式、接続、ディレクトリ、ファイル名など、各種フィールドにパラメータを割り当てることができます。

トランスフォーメーション、論理データオブジェクト、マップレット、マッピング、またはワークフローに対してパラメータを作成した場合、パラメータはそのオブジェクトに適用されます。例えば、トランスフォーメーションでパラメータを作成します。次に、トランスフォーメーションをマップレットに追加します。これでトランスフォーメーションのデフォルトのパラメータ値を使用できます。また、マップレットパラメータを作成してトランスフォーメーションのパラメータ値を上書きすることもできます。

トランスフォーメーションパラメータのデフォルト値を上書きするには、マップレットパラメータをトランスフォーメーションパラメータにバインドします。マップレットパラメータのデフォルト値を変更します。マッピングパラメータでマップレットパラメータを上書きできます。

パラメータを作成できる場所を以下に示します。

```
Workflow parameters
  Mapping parameters
    Maplet parameters
      Logical data objects
        Transformation/data object parameters
```

パラメータセットまたはパラメータファイルでパラメータ値を設定することにより、ワークフローパラメータの値とマッピングパラメータの値を実行時に設定できます。

パラメータを作成すると同時に、フィールドとプロパティにパラメータを割り当てることができます。パラメータをフィールドに割り当てるときに、使用するパラメータを作成できます。以前作成したパラメータを参照することもできます。

ユーザー定義のパラメータは、トランスフォーメーションまたはデータオブジェクトの【パラメータ】タブで維持します。マッピング、マップレット、ワークフロー、論理データオブジェクトにも【パラメータ】タブがあります。【パラメータ】タブでは、パラメータの追加、変更、削除ができます。

注: 【パラメータ】タブでパラメータを作成する場合、パラメータ名の先頭にドル記号 (\$) を含めないでください。

パラメータを割り当てる場所

ユーザー定義のパラメータとシステムパラメータをフィールドに割り当てることができます。ユーザー定義のパラメータは、フィールドに割り当てる前に作成する必要があります。

オブジェクトおよびトランスフォーメーション内のいくつかのプロパティをパラメータ化できます。プロパティにパラメータを割り当てることができる場合、プロパティ値を設定するときにオプションが表示されます。

再利用可能または再利用不可能なトランスフォーメーションのプロパティをパラメータ化できます。再利用可能なトランスフォーメーションをマッピングに追加する場合、マッピングパラメータでデフォルト値を上書きできます。再利用不可能なトランスフォーメーションをパラメータ化する場合、パラメータがマッピングパラメータになります。

読み取りおよび書き込みトランスフォーメーションは、物理データオブジェクトから作成する再利用不可能なトランスフォーメーションです。トランスフォーメーションの作成元である物理データオブジェクトをパラメータ化できます。読み取りおよび書き込みトランスフォーメーション内のいくつかのプロパティにパラメータを割り当てることができます。

次の表に、パラメータを割り当てることができるオブジェクトおよびフィールドを示します。

オブジェクト	フィールド
すべてのトランスフォーメーション	リンクの解決順序
関連付けトランスフォーメーション	キャッシュファイルディレクトリ キャッシュファイルサイズ

オブジェクト	フィールド
アドレスバリデータトランスフォーメーション	大文字小文字表記 デフォルトの国 Geocode データ型 グローバル最大フィールド長 行セパレータ 最大結果数 最適化レベル 無効なアドレスの標準化
アグリゲータトランスフォーメーション	キャッシュディレクトリ 式の要素。完全な式ではありません。 グループ化
不良レコードの例外トランスフォーメーション	下限しきい値 上限しきい値
大文字小文字変換プログラムトランスフォーメーション	参照テーブル。
統合トランスフォーメーション	キャッシュファイルディレクトリ キャッシュファイルサイズ
カスタマイズデータオブジェクト	接続 データオブジェクト Owner (所有者) SQL クエリの要素 テーブル名
ディシジョントランスフォーメーション	ディシジョンスクリプト。
重複レコードの例外トランスフォーメーション	キャッシュファイルディレクトリ 下限しきい値 上限しきい値
式トランスフォーメーション	式の要素。完全な式ではありません。 ポートセクタ
フィルタトランスフォーメーション	フィルタ条件の要素 フィルタ条件。完全な式。

オブジェクト	フィールド
フラットファイルデータオブジェクト	制御ファイルディレクトリ 制御ファイル名 接続名 デフォルトのスケール フラットファイルの区切り文字 マージファイルディレクトリ ソースファイルのディレクトリ ソースファイル名 出力ファイル名 出力ファイルディレクトリ 拒否ファイルディレクトリ ターゲットディレクトリ
ジョイナトランスフォーメーション	キャッシュディレクトリ 結合条件の要素 ポートセクタ
キージェネレータトランスフォーメーション	キャッシュファイルディレクトリ キャッシュファイルサイズ
ラベラトランスフォーメーション	参照テーブル
ルックアップソースの物理データオブジェクトを含まないルックアップトランスフォーメーション	データオブジェクト。再利用不可能なトランスフォーメーション。 動的ポートのルール。再利用不可能なトランスフォーメーション。 ルックアップ条件。完全な式、再利用不可能なトランスフォーメーション。 ポートセクタ。再利用不可能なトランスフォーメーション。
マッピング	Hive バージョン ランタイム環境 最大並行処理
一致トランスフォーメーション	[照合出力] タブのキャッシュディレクトリ [一致タイプ] タブのキャッシュディレクトリ [一致タイプ] タブのインデックスディレクトリ 永続方法 しきい値
非リレーショナルデータオブジェクト	接続
ランクトランスフォーメーション	キャッシュディレクトリ 式の要素。完全な式ではありません。 グループ化ポート ランクポート

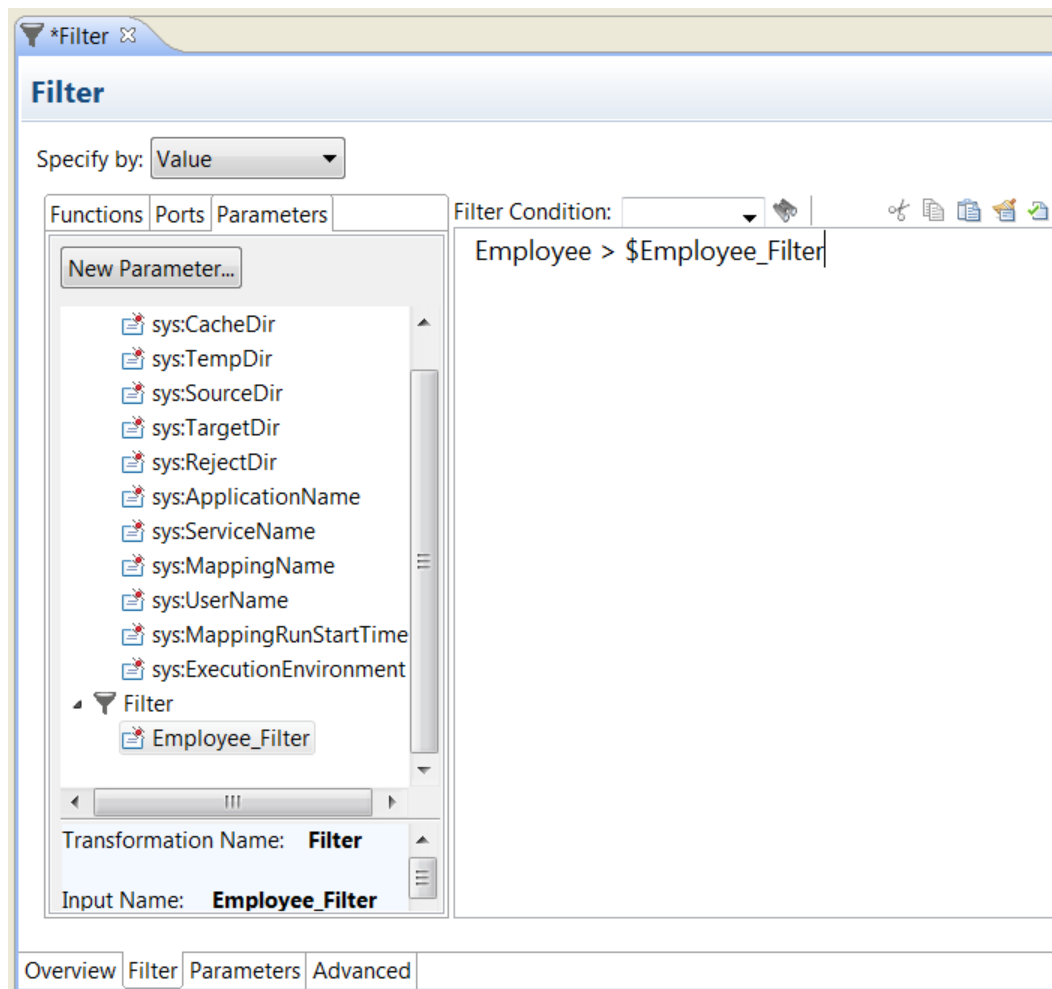
オブジェクト	フィールド
読み取りトランスフォーメーション[よみとりとらんすふぉーめーしょん]	接続 データオブジェクト 所有者名。リレーショナルのみ。 リソース/テーブル名。リレーショナルのみ。
リレーショナルデータオブジェクト	フィルタ条件の要素 結合条件の要素 PreSQL クエリの要素 PostSQL クエリの要素 SQL オーバーライドの要素
ルータートランスフォーメーション	グループフィルタ条件の要素。 グループフィルタ条件。完全な式。
ソータートランスフォーメーション	ソートキー グループ化 ワークディレクトリ
SQL トランスフォーメーション	接続
標準化トランスフォーメーション	参照テーブル
トークンパーサートランスフォーメーション	参照テーブル
アップデイトストラテジトランスフォーメーション	アップデイトストラテジ式の要素。 アップデイトストラテジ式。完全な式。
書き込みトランスフォーメーション[かきこみとらんすふぉーめーしょん]	データオブジェクト リンクの解決順序 拒否ディレクトリ 拒否ファイル名

式のパラメータ

パラメータは、アグリゲータトランスフォーメーション、ルックアップトランスフォーメーション、式トランスフォーメーション、フィルタトランスフォーメーションなどのトランスフォーメーションの、式または条件内で設定できます。

例えば、フィルタトランスフォーメーションのフィルタ条件を設定します。条件に含めるポートとパラメータを選択します。フィルタ条件に含めるシステムパラメータまたはユーザー定義のパラメータを選択します。

次の図は、Employee ポートと Employee_Filter パラメータを含むフィルタ条件を示しています。



ポート名を引数として受け付ける、同じ引数で式のパラメータを使用できます。パラメータを使用して式内の定数引数を置き換えることはできません。

例えば、string を decimal 値に変換する TO_DECIMAL 式を考えてみます。

```
TO_DECIMAL( value [, scale] )
```

式内の scale 引数は定数値である必要があります。

次の有効な式には、scale の定数引数が含まれます。

```
TO_DECIMAL( Input_Port,10 )
```

次の式は有効ではありません。scale 引数に対してユーザー定義のパラメータが含まれているためです。

```
TO_DECIMAL( Input_Port,$Scale_Param )
```

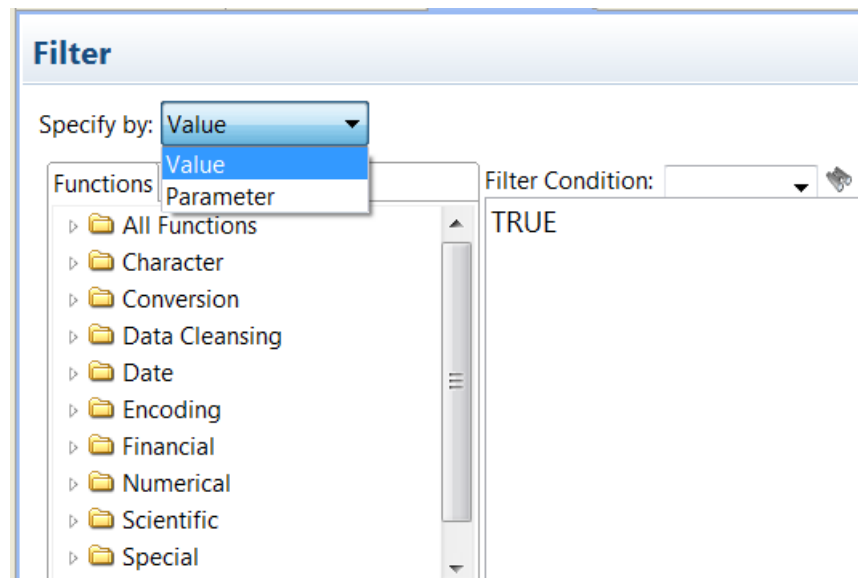
パラメータに別のパラメータを含めることはできません。例えば、トランスフォーメーション内で Parameter1 と Parameter2 を設定した場合、Parameter1 のデフォルト値を \$Parameter2 に設定することはできません。パラメータをネストした場合、マッピングは実行時に検証エラーで失敗します。

式パラメータ

式のパラメータタイプを設定することができます。式パラメータとは、完全な式が含まれるパラメータです。式パラメータは、フィルタトランスフォーメーションとルックアップトランスフォーメーションで使用できます。

式エディタで式パラメータを定義します。[指定元:] で **パラメータ** を選択し、完全な式をパラメータ化することを示します。

次の図は、[指定元:] の **パラメータ** オプションをフィルタ条件として示しています。



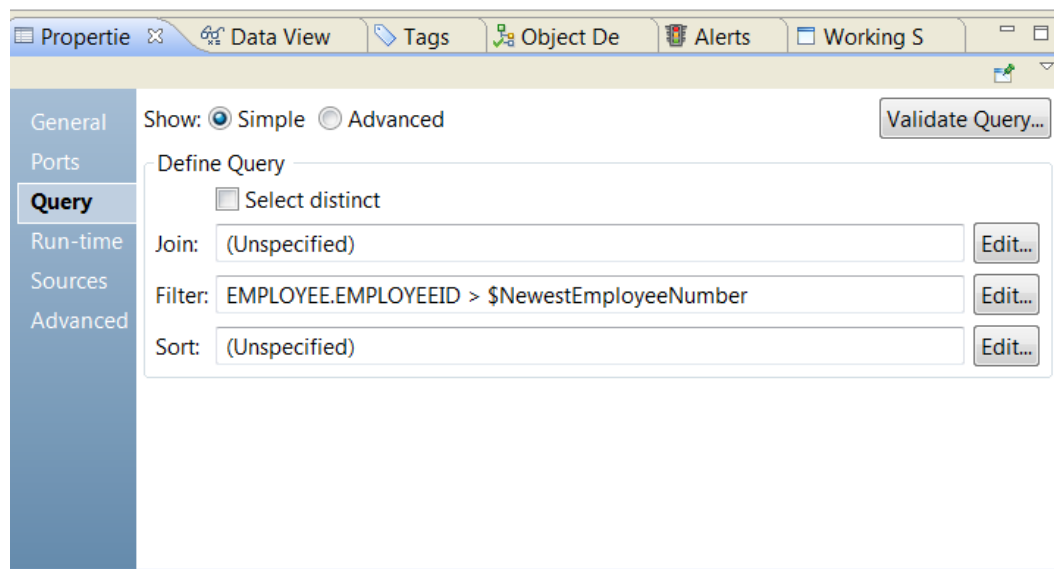
式パラメータを使用する場合、式パラメータを作成するか、既存の式パラメータを選択してトランスフォーメーションで使用することができます。式パラメータには、ポート、演算子、および定数を含めることができます。式パラメータに他のパラメータを含めることはできません。

例えば、フィルタトランスフォーメーション内で、デフォルト値 `EmployeeID > 100` を含むフィルタ式パラメータを作成することができます。マッピング内では、デフォルト値 `Dept < 2000` を含む別の式パラメータを作成できます。マッピングパラメータをトランスフォーメーションパラメータにバインドした場合、マッピングの式パラメータを実行時に上書きできます。動的マッピング用の別のポート名と演算子を持つ式パラメータを作成することができます。

SQL 文のパラメータ

リレーショナルデータオブジェクトまたはルックアップトランスフォーメーションに追加する SQL 文にパラメータを含めることができます。

次の図に、リレーショナルソースを読み取る SQL クエリをパラメータ化する方法を示します。



データ統合サービスは、クエリを展開するときに、一重引用符 (') を持つ各パラメータ内のデータをエスケープします。例えば、\$date_parm という日付パラメータを持つ SQL 文を、次のように指定できます。

```
select * from <table_name> where <date_port> >$date_parm
```

このクエリは展開されると次のようになります。select * from <table_name> where <date_port> > '01/31/2000 00:00:00'

パラメータのデフォルト値で一重引用符を使用しないでください。予期しない結果になる可能性もあります。

パラメータ名にピリオド (.) を含めることはできません。ピリオドを含むパラメータを持つ SQL クエリは無効です。例えば、以下の SQL 文のパラメータ名には、ピリオドが含まれます。

```
SELECT $tname.ID, "MY_SOURCE"."NAME" FROM "MY_SOURCE" where FIELDX=1
```

このクエリを検証すると、データ統合サービスは、tname.ID パラメータを見つけることができないというエラーを返します。

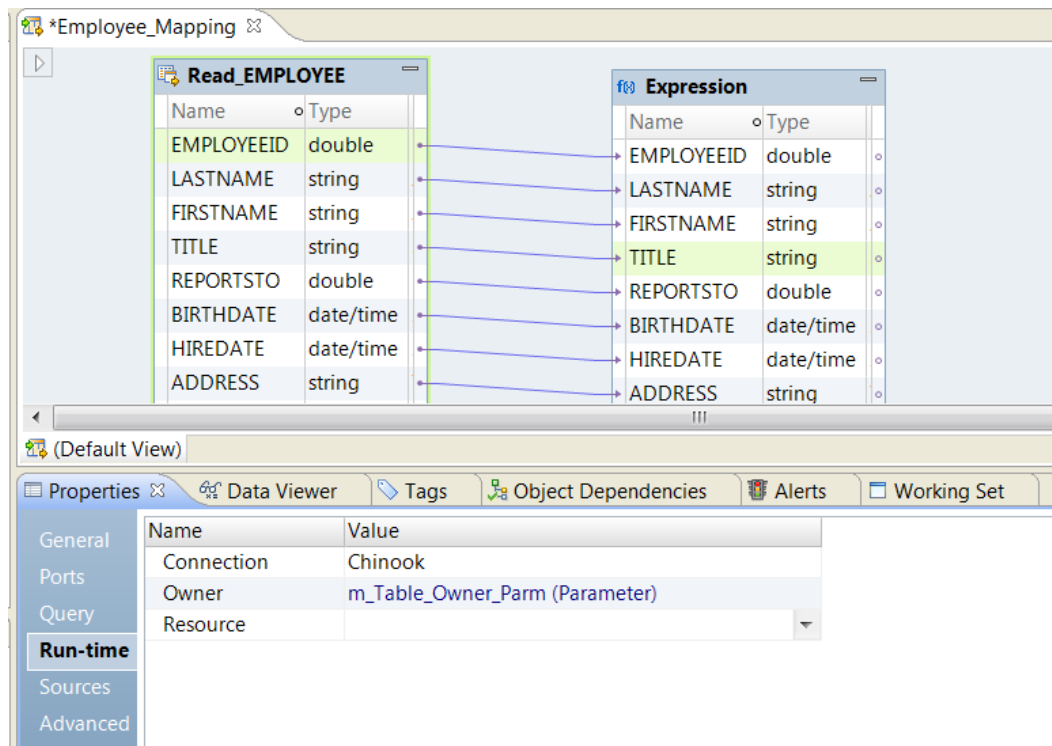
リレーショナルテーブルリソースのパラメータ

読み取りトランスフォーメーションのリソース名、テーブル所有者、および接続をパラメータ化できます。リソースは、リレーショナルデータオブジェクトのテーブル、ビュー、またはシノニムの名前です。

動的マッピングで同じデータベースの複数のテーブルを処理する必要がある場合、リソース名をパラメータ化できます。

マッピングで読み取り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【ランタイム】タブで、【値】カラムをクリックし、接続、テーブル所有者、またはリソースのパラメータを割り当てます。

次の図は、読み取りトランスフォーメーションの接続、リソース名、およびテーブル所有者のパラメータを割り当てるところを示しています。



フィールドおよびプロパティ値のパラメータ

トランスフォーメーションおよび物理データオブジェクトで、いくつかのフィールドおよびプロパティの値のパラメータを設定できます。

リレーショナルデータオブジェクト、カスタマイズデータオブジェクト、およびルックアップトランスフォーメーションの接続名を設定できます。フラットファイルデータオブジェクトでは、入力ファイルディレクトリ、出力ファイルディレクトリ、および拒否ファイルディレクトリのパラメータを設定できます。フラットファイル区切り文字のタイプを変更するように、パラメータを設定することもできます。

次の図は、物理データオブジェクトの【詳細】タブにあるフラットファイル区切り文字のパラメータを示しています。

Advanced

Format

Code page: MS Windows Latin 1 (ANSI), superset of Latin1

Format

☒ Delimited (fields separated by delimiters)

☐ Fixed-width (fields aligned in columns)

Delimiters

☐ Use Fixed Values

☐ Tab ☐ Semicolon ☐ Comma ☐ Space ☐ Other: ...

☒ Assign Parameter

...

Text qualifier

☒ No quotes ☐ Single quotes ☐ Double quotes

ポートリストのパラメータ

ポートのリストを含むパラメータを作成できます。このパラメータを、ソータートランスフォーメーション、ランクトランスフォーメーション、ジョイナートランスフォーメーション、式トランスフォーメーションなどの各種トランスフォーメーションで参照できます。

複数のポート名が含まれる以下のタイプのパラメータを設定できます。

ポートリスト

カンマ区切りのポート名のリスト。ポートリストパラメータの構文は次のとおりです。Port1,Port2,Port3

ソートリスト

ポート名のリスト、および各ポートのソートタイプ。ソートリストパラメータの構文は次のとおりです。
Port1:A,Port2:A,Port3:D

入力リンクセット

実行時にリンクするポートのセット。リンクセットのパラメータには名前と値のペアが含まれます。このペアの構文は次のとおりです。Port1>:=Port2, Port3>:=Port4

マッピングのパラメータ

再利用可能なトランスフォーメーションまたはデータオブジェクトでパラメータを定義した場合、マッピングにトランスフォーメーションを追加するときにパラメータのデフォルト値を上書きできます。特定のマッピング内のパラメータ値を上書きするように、マッピングパラメータを設定します。

マッピングパラメータを定義する場合、マッピングパラメータを特定のトランスフォーメーションパラメータにバインドできます。マッピングパラメータの値が、トランスフォーメーション内のデフォルトのパラメータ値を上書きします。

マッピングパラメータをトランスフォーメーションパラメータにバインドする場合、それらのパラメータのタイプが同じである必要があります。マッピングパラメータ名は、トランスフォーメーションパラメータ名と同じである必要はありません。

パラメータセットまたはパラメータファイルを使用して、マッピングパラメータの値を実行時に設定することができます。パラメータセットまたはパラメータファイルを使用して、トランスフォーメーションパラメータの値を設定することはできません。実行時にパラメータ値を変更する場合は、マッピングパラメータを設定する必要があります。

以下のいずれかの方法を使用して、マッピングパラメータを定義します。

マッピングの [プロパティ] ビューの [パラメータ] タブで、マッピングパラメータを定義する

マッピングの [パラメータ] タブで、各パラメータ名、パラメータ属性、およびデフォルト値を手動で入力できます。マッピングにトランスフォーメーションを追加するときには常に、これらのパラメータをトランスフォーメーションパラメータにバインドできます。マッピングの [パラメータ] タブで、マッピングパラメータを更新できます。

トランスフォーメーションパラメータからマッピングパラメータを追加する

マッピングにトランスフォーメーションを追加したら、マッピングパラメータをトランスフォーメーションの [パラメータ] タブから直接作成できます。トランスフォーメーションパラメータをマッピングパラメータとして公開できます。Developer tool は、トランスフォーメーションパラメータと同じプロパティを持つマッピングパラメータを作成します。

再利用不可能なトランスフォーメーションにパラメータを追加する

マッピング内でトランスフォーメーションを作成した場合、そのトランスフォーメーションは再利用不可能なトランスフォーメーションです。トランスフォーメーションのいずれかのプロパティをパラメータ化する場合、トランスフォーメーションパラメータではなくマッピングパラメータを作成します。

パラメータインスタンス値

パラメータ付きの再利用可能なトランスフォーメーションをマッピングに追加する場合、トランスフォーメーションの各パラメータのインスタンス値を設定できます。

このインスタンス値は、特定のマッピングのパラメータ値です。デフォルト値、特定の値、またはマッピングパラメータの値にインスタンス値を設定できます。

マッピングパラメータまたはマッピングパラメータは、トランスフォーメーションパラメータのデフォルト値を上書きできます。マッピングパラメータまたはマッピングパラメータを選択し、そのパラメータをトランスフォーメーションパラメータにバインドします。

[プロパティ] ビューの、トランスフォーメーションの [パラメータ] タブで、インスタンス値を設定します。

[インスタンス値] に対して、次のいずれかのオプションを選択します。

マッピングパラメータとして公開

トランスフォーメーションパラメータと同じ属性のマッピングパラメータを作成します。同じ手順で、マッピングパラメータをトランスフォーメーションパラメータにバインドします。

パラメータ

トランスフォーメーションパラメータにバインドするマッピングパラメータを参照し、選択します。マッピングパラメータを作成してトランスフォーメーションパラメータにバインドすることもできます。マッピングパラメータを作成してバインドする場合は、[マッピングパラメータとして公開] オプションと同じタスクを実行します。ただし、マッピングパラメータを手動で作成する場合、トランスフォーメーションパラメータとは別の名前を設定できます。

デフォルトを使用

トランスフォーメーションパラメータのデフォルト値を使用します。トランスフォーメーションパラメータへのマッピングパラメータのバインドをスキップします。

値

マッピングで使用するデフォルトのパラメータ値を入力します。トランスフォーメーションパラメータへのマッピングパラメータのバインドをスキップします。

マップレットのパラメータ

マップレットパラメータは、データオブジェクトのパラメータにバインドしたり、マップレット内にあるトランスフォーメーションのパラメータにバインドしたりできます。

マップレットパラメータを定義する場合、マップレットパラメータを特定のトランスフォーメーションパラメータにバインドできます。マップレットパラメータの値が、トランスフォーメーション内のデフォルトのパラメータ値を上書きします。マップレットパラメータをトランスフォーメーションパラメータにバインドする場合、それらのパラメータのタイプが同じである必要があります。マップレットパラメータ名は、トランスフォーメーションパラメータ名と同じである必要はありません。マップレットパラメータを複数のトランスフォーメーションパラメータにバインドすることができます。

以下のいずれかの方法を使用して、マップレットパラメータを定義します。

マップレットの【プロパティ】ビューの【パラメータ】タブで、マップレットパラメータを定義する

マップレットの【パラメータ】タブで、各パラメータ名、パラメータ属性、およびデフォルト値を手動で入力できます。

トランスフォーメーションパラメータからマップレットパラメータを追加する

マップレットにトランスフォーメーションを追加したら、マップレットパラメータをトランスフォーメーションの【パラメータ】タブから直接作成できます。

マップレットのパラメータインスタンス値

トランスフォーメーションパラメータ付きの再利用可能なトランスフォーメーションをマップレットに追加する場合、各パラメータのインスタンス値を設定できます。パラメータのインスタンス値は、特定のマップレットのパラメータ値です。

マップレットにトランスフォーメーションを追加したら、トランスフォーメーションの【パラメータ】タブでインスタンス値を設定します。

【インスタンス値】に対して、次のいずれかのオプションを選択します。

マップレットパラメータとして公開

トランスフォーメーションパラメータと同じ属性のマップレットパラメータを作成します。同じ手順で、マッピングパラメータをトランスフォーメーションパラメータにバインドします。

パラメータ

トランスフォーメーションパラメータにマップレットパラメータをバインドします。トランスフォーメーションパラメータにバインドするマップレットパラメータを参照し、選択することができます。マップレットパラメータを作成し、そのパラメータをトランスフォーメーションパラメータにバインドすることもできます。マップレットパラメータを作成してバインドする場合は、【マップレットパラメータとして公開】オプションと同じタスクを実行します。ただし、マップレットパラメータを手動で作成する場合、トランスフォーメーションパラメータとは別の名前と別のデフォルト値を設定できます。

デフォルトを使用

トランスフォーメーションパラメータのデフォルト値を使用します。トランスフォーメーションパラメータへのマップレットパラメータのバインドをスキップします。

値

マップレットで使用する別のデフォルトのパラメータ値を入力します。トランスフォーメーションパラメータへのマップレットパラメータのバインドをスキップします。

マッピングのマップレットパラメータ

マップレットパラメータ付きのマップレットをマッピングに追加する場合、マップレットパラメータのインスタンス値を設定できます。マップレットパラメータのインスタンス値は、特定のマッピングのパラメータ値です。

【プロパティ】ビューの、マップレットの【パラメータ】タブで、インスタンス値を設定します。

【インスタンス値】に対して、次のいずれかのオプションを選択します。

マッピングパラメータとして公開

マップレットパラメータと同じ属性のマッピングパラメータを作成します。同じ手順で、マッピングパラメータをマップレットパラメータにバインドします。

パラメータ

マップレットパラメータにマッピングパラメータをバインドします。マップレットパラメータにバインドするマッピングパラメータを参照し、選択することができます。マッピングパラメータを作成してマップレットパラメータにバインドすることもできます。マッピングパラメータを作成してバインドする場合は、【マッピングパラメータとして公開】オプションと同じタスクを実行します。ただし、マッピングパラメータを手動で作成する場合、マップレットパラメータとは別の名前とデフォルト値でマッピングパラメータを設定できます。

デフォルトを使用

マップレットパラメータのデフォルト値を使用します。マップレットパラメータへのマッピングパラメータのバインドをスキップします。

値

マッピングで使用するデフォルトのパラメータ値を入力します。マップレットパラメータへのマッピングパラメータのバインドをスキップします。

マップレットのパラメータの例

マップレットパラメータを定義し、マッピングパラメータでマップレットパラメータを上書きできます。

顧客テーブルから顧客データを返す SQL トランスフォーメーションを定義することができます。SQL トランスフォーメーションをマップレットに追加し、ランタイム接続をパラメータ化します。

次に、異なるデータベースから顧客データを取得するマッピングにマップレットを追加します。マップレットパラメータのデフォルト接続をオーバーライドするように、各マッピングのマッピングパラメータを定義します。

次の表に、マップレットとマッピングに対して作成できる接続パラメータのリストを示します。

オブジェクト名	オブジェクトタイプ	パラメータ名	パラメータのデフォルト値
mp_Get_Customer	マップレット	mp_cust_connection	Oracle_Default
m_billing_mapping	マッピング	m_acctg_connection	Oracle_AcctDB
m_order_fill_mapping	マッピング	m_shipping_connection	Oracle_Warehouse
m_cust_main_mapping	マッピング	m_master_connection	Oracle_Cust_Mast

mp_Get_Customer のマップレットには、mp_cust_connection という接続パラメータがあります。このパラメータのデフォルトの接続名は Oracle_Default です。この接続では、例えばテストデータベースを参照します。

各マッピングに、mp_cust_connection パラメータをオーバーライドする接続パラメータがあります。各マッピングは、会計データベース、倉庫データベース、または顧客マスタデータベースに接続されます。

各マッピングパラメータをマップレットパラメータにバインドし、デフォルト値をオーバーライドする必要があります。マッピングパラメータの値を実行時に変更するには、パラメータセットまたはパラメータファイルを設定できます。

論理データオブジェクトのパラメータ

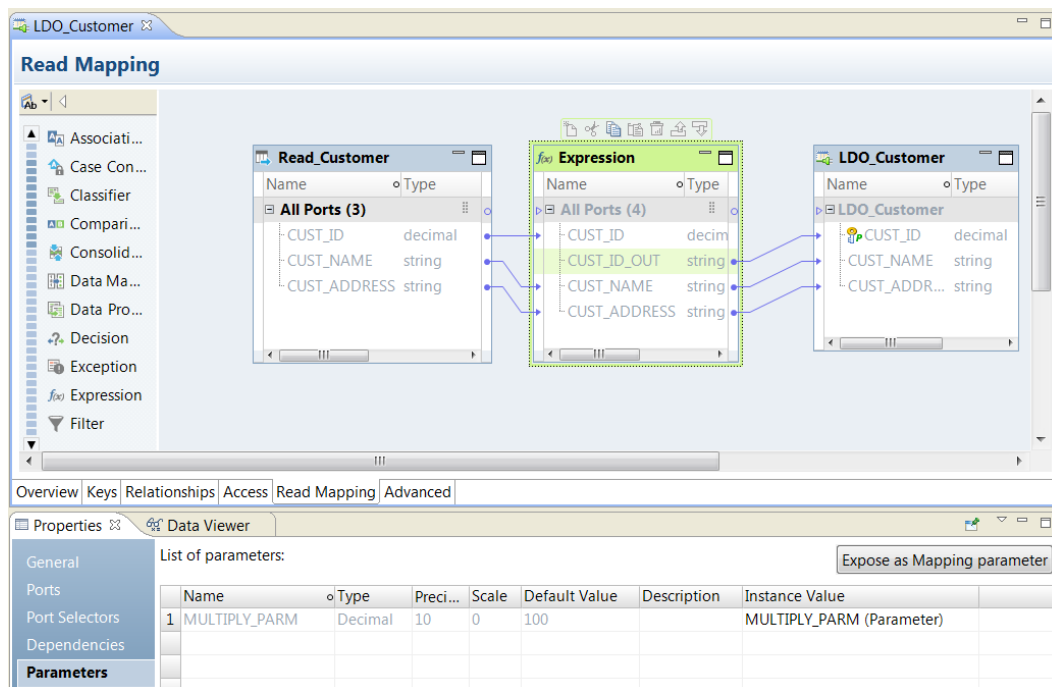
論理データオブジェクトにパラメータを含めることができます。パラメータは、トランスフォーメーションや読み取りおよび書き込みマッピングで使用できます。

論理データオブジェクトは、読み取りマッピングと書き込みマッピングを持つことができます。読み取りまたは書き込みマッピングには、パラメータを使用するトランスフォーメーションを含めることができます。再利用可能なトランスフォーメーションパラメータを読み取りまたは書き込みマッピングのパラメータにバインドできます。

例えば、式トランスフォーメーションが含まれる読み取りマッピングを持っている論理データオブジェクトがあります。式トランスフォーメーションには、式内の decimal 値を定義するパラメータがあります。デフォルト値は 100 です。

式トランスフォーメーションを読み取りマッピングに追加する場合、異なるパラメータ値を使用することが必要になる場合があります。読み取りマッピングレベルでパラメータを作成し、トランスフォーメーションパラメータをオーバーライドすることができます。[マッピングパラメータとして公開] をクリックし、重複するパラメータを読み取りマッピングで作成します。Developer tool は、重複するパラメータをトランスフォーメーションパラメータにバインドします。

次の図は、読み取りマッピング内の式トランスフォーメーションの [パラメータ] タブを示しています。



重複したパラメータを表示するには、エディタをクリックして、マッピングの【プロパティ】ビューを表示します。【パラメータ】タブをクリックします。読み取りマッピングレベルでパラメータのデフォルト値を変更できます。

論理データオブジェクトをマップレットまたはマッピングに追加する場合、読み取りマッピングパラメータをオーバーライドできます。マップレットまたはマッピングで、重複するパラメータを作成します。重複するパラメータのデフォルト値を変更します。

仮想テーブルマッピングのパラメータ

仮想テーブルマッピングは、SQL データサービスのソースと仮想テーブル間のデータフローを定義します。仮想テーブルマッピングにはパラメータを含めることができますが、パラメータファイルまたはパラメータセットを使用してパラメータのデフォルト値をオーバーライドすることはできません。

仮想テーブルマッピングには、パラメータが含まれる再利用可能なトランスフォーメーションまたはマップレットを含めることができます。マッピングパラメータを、仮想テーブルマッピングのトランスフォーメーションパラメータまたはマップレットパラメータにバインドできます。

ただし、仮想テーブルマッピングにパラメータが含まれる場合、データ統合サービスは、マッピングレベルのデフォルトのパラメータ値を適用します。データ統合サービスは、パラメータファイルまたはパラメータセットから仮想テーブルマッピングのパラメータに値をバインドできません。

仮想テーブルマッピングに接続された、パラメータ化されたソースを使用できます。マッピングはデフォルトのパラメータ値を使用します。

パラメータセット

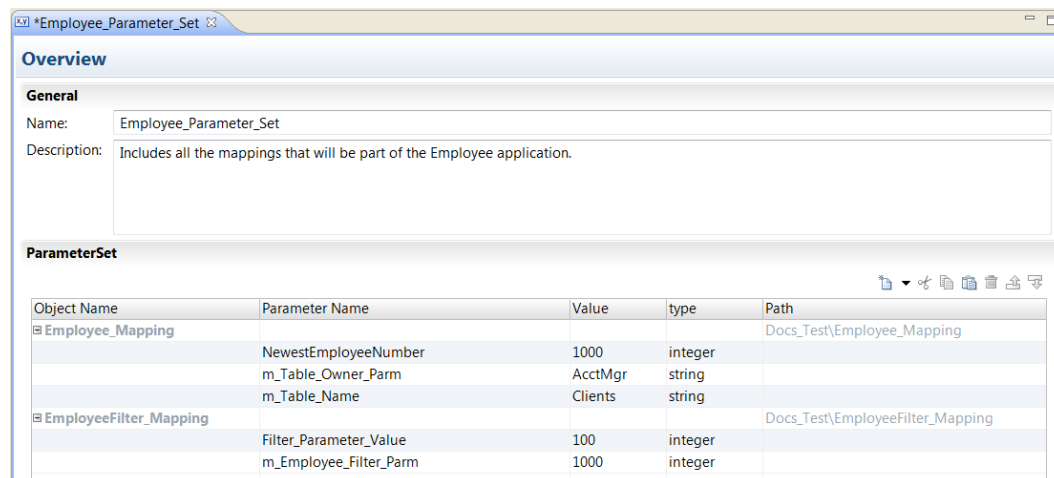
パラメータセットはモデルリポジトリ内のオブジェクトであり、マッピングおよびワークフローの実行に使用する一連のパラメータとその値を含みます。

パラメータセットを作成する場合、パラメータを使用するマッピングまたはワークフローを選択します。マッピングまたはワークフローを選択後、パラメータセットのパラメータを手動で入力するか、マッピングまたはワークフローのリポジトリにすでにあるパラメータを選択できます。

さまざまな状況でパラメータセットを使用できます。例えば、テスト環境でワークフローを実行する場合に特定のパラメータセットを使用することがあります。

マッピング、マッピングタスク、またはワークフローでパラメータセットを使用します。アプリケーションをデプロイする場合、アプリケーションに1つ以上のパラメータセットを追加できます。複数のアプリケーションに1つのパラメータセットを追加し、デプロイすることができます。ワークフローまたはマッピングでパラメータセットを使用するには、ワークフローまたはマッピングのデプロイ時にアプリケーションにパラメータセットを追加する必要があります。

次の画像は、2つのマッピングのパラメータを含むパラメータセットを示します。



Object Name	Parameter Name	Value	type	Path
Employee_Mapping	NewestEmployeeNumber	1000	integer	Docs_Test\Employee_Mapping
	m_Table_Owner_Parm	AcctMgr	string	
	m_Table_Name	Clients	string	
EmployeeFilter_Mapping	Filter_Parameter_Value	100	integer	Docs_Test\EmployeeFilter_Mapping
	m_Employee_Filter_Parm	1000	integer	

パラメータセットには次の情報が含まれます。

オブジェクト名

パラメータの定義を含むマッピング、マップレット、またはワークフローの名前

パラメータ名

マッピング、マップレット、またはワークフローのパラメータ名

値

ランタイムに使用するパラメータ値パラメータセットのパラメータ値は、マッピングまたはワークフローのパラメータ値をオーバーライドします。

タイプ

パラメータのタイプ。パラメータタイプの例には、文字列、数値タイプ、接続、ポートリスト、ソートリスト、および日付/時刻パラメータが含まれます。

Infacmd からパラメータセットを使用してマッピングを実行する

マッピングをアプリケーションとしてデプロイし、そのアプリケーションにパラメータセットを含めることができます。これにより、デプロイされたアプリケーションを実行して、パラメータセットを使用できます。

マッピングをデプロイしたら、デプロイ済みマッピングをパラメータセットを使用してコマンドラインから実行できます。異なるパラメータセットを使用する必要がある場合は、アプリケーションに複数のパラメータセットをデプロイできます。マッピングを実行する場合に、使用するパラメータセットを指定できます。

アプリケーションがデプロイされたら、`infacmd addParameterSetEntries` コマンドを使用してパラメータセットのエントリを追加できます。パラメータセットのエントリを更新するには、`infacmd updateParameterSetEntries` コマンドを使用します。

`infacmd` を使用してパラメータセットを使用する方法の詳細については、『*Informatica コマンドリファレンス*』を参照してください。

パラメータファイル

パラメータファイルは、ユーザー定義パラメータおよびその割り当て値を列挙した XML ファイルです。パラメータファイルを使用すると、コマンドラインから `mapping` を実行するたびにパラメータ値の変更を柔軟に行えます。

パラメータ値を使用して、ワークフローの、またはワークフローを実行するマッピングタスクに含まれるマッピングまたはマップレットのプロパティを定義します。データ統合サービスでこれらの値を適用するには、コマンドラインからワークフローを実行するときにパラメータファイルを指定します。

パラメータ値を使用して、マッピングまたはマップレットのプロパティを定義します。データ統合サービスでこれらの値を適用するには、コマンドラインからマッピングを実行するときにパラメータファイルを指定します。

パラメータファイルでマッピングパラメータおよびワークフローパラメータを定義できます。パラメータファイルでシステムパラメータの値を定義することはできません。

1 つのパラメータファイルで複数の `mapping` のパラメータを定義できます。複数のパラメータファイルを作成し、`mapping` を実行するたびに異なるファイルを使用することもできます。データ統合サービスは、`mapping` 実行の開始時にパラメータファイルを読み込み、パラメータを解決します。

`ms ListMappingParams` コマンドは、`mapping` で使用されるパラメータをそのデフォルト値とともに一覧表示します。このコマンドの出力は、パラメータファイルのテンプレートとして使用できます。

パラメータファイルを指定して `ms RunMapping` を実行するには、`mapping` コマンドを使用します。

注: パラメータファイルの構造は、マッピングでもワークフローでも同じです。1 つのパラメータファイルでデプロイ済みのマッピングとデプロイ済みのワークフローのパラメータを定義できます。

パラメータファイルの構造

パラメータファイルは、少なくとも 1 つのパラメータとその割り当て値を含む XML ファイルです。

Data Integration Service はパラメータファイルで定義された階層を使用して、パラメータおよびその定義値を識別します。この階層によって、パラメータを使用したワークフロー、マッピング、またはマップレットが識別されます。

Data Integration Service はパラメータファイルで定義された階層を使用して、パラメータおよびその定義値を識別します。この階層によって、パラメータを使用したマッピング、マップレット、またはワークフローが識別されます。

パラメータ値は、最上位の要素である project 要素または application 要素で定義します。project 要素では、任意のデプロイ済みのアプリケーションでプロジェクト内の特定の mapping を実行する場合に使用するパラメータ値を定義します。また、プロジェクト内のオブジェクトを使用する任意の mapping を実行する場合に使用するパラメータ値も定義します。application 要素では、特定のデプロイ済みのアプリケーションで特定の mapping を実行する場合に使用するパラメータ値を定義します。同じパラメータファイルで最上位の project 要素と application 要素の両方に同じパラメータが定義されている場合、application 要素で定義されているパラメータ値が優先されます。

Data Integration Service は、以下の順でパラメータ値を探します。

1. application 要素内に指定された値。
2. project 要素内に指定された値。
3. パラメータのデフォルト値。

パラメータファイルは、パラメータファイルの XML スキーマ定義（XSD）の構造に従う必要があります。パラメータファイルがスキーマ定義に従っていない場合、mapping の実行は失敗します。

Developer ツールをホストするマシンでは、パラメータファイル XML スキーマ定義は次のディレクトリにあります。

```
<Informatica Installation Directory>\clients\DeveloperClient\infacmd\plugins\ms\parameter_file_schema_1_0.xsd
```

Informatica Services をホストするマシンでは、パラメータファイル XML スキーマ定義は次のディレクトリにあります。

```
<Informatica Installation Directory>\isp\bin\plugins\ms\parameter_file_schema_1_0.xsd
```

project 要素

project 要素は、任意のデプロイ済みのアプリケーションのプロジェクト内で特定の mapping を実行する場合に使用するパラメータ値を定義します。project 要素はまた、プロジェクト内のオブジェクトを使用する任意の mapping を実行する場合に使用するパラメータ値も定義します。

project 要素では、パラメータを使用するオブジェクトを含むモデルリポジトリ内のプロジェクトを定義します。project 要素には、ワークフローまたはマッピングを含めることができます。プロジェクトにトランスフォーメーションやデータソースを含めることはできません。

次の表に、project 要素に含めることができる要素を示します。

エレメント名	説明
folder	プロジェクト内のフォルダを定義します。folder 要素は、オブジェクトをプロジェクト内の複数のフォルダにまとめる場合に使用します。 folder 要素には、dataSource、mapping、mapplet、transformation、workflow の各要素を含めることができます。 folder 要素には、dataSource、mapping、mapplet、transformation の各要素を含めることができます。
mapping	パラメータを使用するプロジェクト内のマッピングを定義します。マッピング要素には、マッピング、パラメータを受け付けるマッピング内の再利用不可能なデータオブジェクト、再利用不可能なトランスフォーメーション、再利用可能なルックアップトランスフォーメーションのパラメータ値を定義する 1 つ以上のパラメータ要素が含まれています。
workflow	パラメータを使用するプロジェクト内のワークフローを定義します。workflow 要素には、ワークフローのパラメータ値を定義する 1 つ以上のパラメータ要素が含まれています。

最上位の project 要素でパラメータ値が定義されているパラメータファイルを使用してワークフローを実行すると、指定されたワークフローにそのパラメータ値が適用されます。また、ワークフローのマッピングタスク

で実行されるマッピングに指定されたオブジェクトが含まれている場合は、それらのオブジェクトにもパラメータ値が適用されます。

例えば、「MyWorkflow」というワークフローを実行するときにパラメータ値が適用されるようにするとします。このワークフローには「MyMapping」というマッピングを実行するマッピングタスクがあり、このマッピングには「MyDataObject」というデータオブジェクトと「MyTransformation」という再利用可能なトランスフォーメーションが含まれています。「MyWorkflow」を実行するときは、どのデプロイ済みのアプリケーションでもパラメータ値を使用します。さらに、これらのオブジェクトをプロジェクト「MyProject」で使用するマッピングを実行するマッピングタスクが含まれる他のワークフローを実行するときにも、パラメータ値を使用するとします。以下の要素内にパラメータを定義します。

```
<project name="MyProject">

  <!-- Apply this parameter value to workflow "MyWorkflow" in project "MyProject". -->
  <workflow name="MyWorkflow">
    <parameter name="MyWorkflow_Param">Param_value</parameter>
  </workflow>

  <!-- Apply this parameter value when you run any workflow that runs mapping "MyMapping"
        in project "MyProject". -->
  <mapping name="MyMapping">
    <parameter name="MyMapping_Param">Param_value</parameter>
  </mapping>
</project>
```

最上位の project 要素でパラメータ値が定義されているパラメータファイルを使用してマッピングを実行すると、指定されたマッピングにそのパラメータ値が適用されます。

例えば、「MyMapping」というマッピングを実行するときにパラメータ値が適用されるようにとします。

```
<project name="MyProject">

  <!-- Apply this parameter value to mapping "MyMapping" in project "MyProject". -->
  <mapping name="MyMapping">
    <parameter name="MyMapping_Param">Param_value</parameter>
  </mapping>

</project>
```

application 要素

application 要素は、project 要素の実行時の範囲を指定する要素です。application 要素では、特定のデプロイ済みのアプリケーションで特定の mapping を実行する場合に使用するパラメータ値を定義します。

application 要素では、パラメータを使用するオブジェクトを含むデプロイ済みのアプリケーションを定義します。application 要素には、パラメータを使用するデプロイ済みのアプリケーションの mapping を定義する mapping 要素を含めることができます。mapping 要素には project 要素が含まれます。

例えば、「MyApp」というデプロイ済みのアプリケーションで「MyWorkflow」というワークフローを実行するときにパラメータ値が適用されるようにとします。他のアプリケーションでワークフローを実行する場合や、プロジェクト「MyProject」内の別のワークフローを実行する場合は、パラメータ値を使用しません。以下の要素内にパラメータを定義します。

```
<application name="MyApp">
  <workflow name="MyWorkflow">
    <project name="MyProject">
      <workflow name="MyWorkflow">
        <parameter name="MyWorkflow_Param">Param_value</parameter>
      </workflow>

      <mapping name="MyMapping">
        <parameter name="MyMapping_Param">Param_value</parameter>
      </mapping>
    </project>
  </workflow>
```

```
</application>
```

例えば、「MyApp」というデプロイ済みのアプリケーションで「MyMapping」というマッピングを実行するときにパラメータ値が適用されるようにします。他のアプリケーションでマッピングを実行する場合や、プロジェクト「MyProject」内の別のマッピングを実行する場合は、パラメータ値を使用しません。以下の要素内にパラメータを定義します。

```
<application name="MyApp">
  <mapping name="MyMapping">
    <project name="MyProject">
      <mapping name="MyMapping">
        <parameter name="MyMapping_Param">Param_value</parameter>
      </mapping>
    </project>
  </mapping>
</application>
```

パラメータファイルに関するルールとガイドライン

パラメータファイルを作成するときは、特定のルールとガイドラインが適用されます。

パラメータファイルを作成する際のルールは次のとおりです。

- パラメータファイルのマッピングレベルパラメータを参照できます。トランスフォーメーションレベルパラメータは参照できません。
- パラメータ値は空にできません。例えば、パラメータファイルに以下のエントリがある場合、mapping は失敗します。

```
<parameter name="Param1"> </parameter>
```

- 要素内のアーティファクト名は大文字と小文字が区別されません。したがって、データ統合サービスは <parameter name="SrcDir"> と <parameter name="Srcdir"> を同じアプリケーションとして解釈します。
- 参照テーブルを特定するパラメータは、リポジトリフォルダパスでフォルダ名を区切るのにフォワードスラッシュ (/) を使用する必要があります。

パラメータファイルの例

以下に、mapping の実行に使用されるパラメータファイルの例を示します。

```
<?xml version="1.0"?>
<root description="Sample Parameter File"
  xmlns="http://www.informatica.com/Parameterization/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!--
    The Data Integration Service uses this section only when you run workflow "Workflow1" or
    "Workflow2" in project "Project1" in deployed application "App1."

    This section assigns values to parameters created in workflows "Workflow1" and "Workflow2."
  -->
  <application name="App1">
    <workflow name="Workflow1">
      <project name="Project1">
        <workflow name="Workflow1">
          <parameter name="WORKFLOW1_PARAM1">WORKFLOW1_PARAM1_VAL</parameter>
          <parameter name="WORKFLOW1_PARAM2">WORKFLOW1_PARAM2_VAL</parameter>
        </workflow>
      </project>
    </workflow>
    <workflow name="Workflow2">
      <project name="Project1">
        <workflow name="Workflow2">
          <parameter name="WORKFLOW2_PARAM1">WORKFLOW2_PARAM1_VAL</parameter>
```

```

        <parameter name="WORKFLOW2_PARAM2">WORKFLOW2_PARAM2_VAL</parameter>
    </workflow>
</project>
</workflow>
</application>

<!--
The Data Integration Service uses this section only when you run workflow "Workflow1"
in project "Project1" in deployed application "App2." "Workflow1" includes a Mapping
task that runs mapping "Map1".

This section assigns values to parameters created in the following objects:
* Workflow "Workflow1"
* Mapping "Map1"
-->
<application name="App2">
  <workflow name="Workflow1">
    <project name="Project1">
      <workflow name="Workflow1">
        <parameter name="WORKFLOW1_PARAM1">WORKFLOW1_PARAM1_VAL</parameter>
        <parameter name="WORKFLOW1_PARAM2">WORKFLOW1_PARAM2_VAL</parameter>
      </workflow>

      <mapping name="Map1">
        <parameter name="MAP1_PARAM2">MAP1_PARAM2_VAL</parameter>
      </mapping>
    </project>
  </workflow>
</application>

<!--
The Data Integration Service uses this section when you run any workflow that
includes a Mapping task that runs a mapping that includes data source "DS1" or
mapplet "DS1" in project "Project1".

This section assigns values to parameters created in the following objects:
* Mapplet "DS1"
-->
<project name="Project1">
  <mapplet name="DS1">
    <parameter name="PROJ1_DS1">PROJ1_DS1_VAL</parameter>
    <parameter name="PROJ1_DS1_PARAM1">PROJ1_DS1_PARAM1_VAL</parameter>
  </mapplet>
</project>

<!--
The Data Integration Service uses this section when you run any workflow that
includes a Mapping task that runs a mapping that includes reusable transformation
"TX2", mapplet "MPLT1" in folder "Folder2", or Mapplet "RULE1" in nested folder
"Folder2_1_1" in project "Project2".

This section assigns values to parameters created in the following objects:
* Mapplet "MPLT1" in folder "Folder2"
* Mapplet "RULE1" in nested folder "Folder2_1_1"
-->
<project name="Project2">
  <folder name="Folder2">
    <mapplet name="MPLT1">
      <parameter name="PROJ2_FOLD2_MPLT1">PROJ2_FOLD2_MPLT1_VAL</parameter>
    </mapplet>
    <folder name="Folder2_1">
      <folder name="Folder2_1_1">
        <mapplet name="RULE1">
          <parameter name="PROJ2_RULE1">PROJ2_RULE1_VAL</parameter>
        </mapplet>
      </folder>
    </folder>
  </folder>

```

```

        </folder>
    </project>
</root>

<?xml version="1.0"?>
<root description="Sample Parameter File"
    xmlns="http://www.informatica.com/Parameterization/1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <!--
        The Data Integration Service uses this section only when you run mapping "Map1" or "Map2"
        in project "Project1" in deployed application "App1."

        This section assigns values to parameters created in mappings "Map1" and "Map2."
    -->
    <application name="App1">
        <mapping name="Map1">
            <project name="Project1">
                <mapping name="Map1">
                    <parameter name="MAP1_PARAM1">MAP1_PARAM1_VAL</parameter>
                    <parameter name="MAP1_PARAM2">MAP1_PARAM2_VAL</parameter>
                </mapping>
            </project>
        </mapping>
        <mapping name="Map2">
            <project name="Project1">
                <mapping name="Map2">
                    <parameter name="MAP2_PARAM1">MAP2_PARAM1_VAL</parameter>
                    <parameter name="MAP2_PARAM2">MAP2_PARAM2_VAL</parameter>
                </mapping>
            </project>
        </mapping>
    </application>

    <!--
        The Data Integration Service uses this section only when you run mapping "Map1" in
        project "Project1" in deployed application "App2."

        This section assigns values to parameters created in the following objects:

        * Mapping "Map1"
    -->
    <application name="App2">
        <mapping name="Map1">
            <project name="Project1">
                <mapping name="Map1">
                    <parameter name="MAP1_PARAM2">MAP1_PARAM2_VAL</parameter>
                </mapping>
            </project>
        </mapping>
    </application>

    <!--
        The Data Integration Service uses this section when you run any mapping that
        includes mapplet "DS1" in project "Project1."

        This section assigns values to parameters created in the following objects:

        * Mapplet "DS1"
    -->
    <project name="Project1">
        <mapplet name="DS1">
            <parameter name="PROJ1_DS1">PROJ1_DS1_VAL</parameter>
            <parameter name="PROJ1_DS1_PARAM1">PROJ1_DS1_PARAM1_VAL</parameter>
        </mapplet>
    </project>

    <!--
        The Data Integration Service uses this section when you run any mapping that
        includes reusable transformation "TX2", mapplet "MPLT1" in folder "Folder2",

```

or Maplet "RULE1" in nested folder "Folder2_1_1" in project "Project2".

This section assigns values to parameters created in the following objects:

- * Reusable transformation "TX2"
- * Maplet "MPLT1" in folder "Folder2"
- * Maplet "RULE1" in nested folder "Folder2_1_1"

```
-->
<project name="Project2">
  <transformation name="TX2">
    <parameter name="RTM_PATH">Project1\Folder1\RTM1</parameter>
  </transformation>
  <folder name="Folder2">
    <maplet name="MPLT1">
      <parameter name="PROJ2_FOLD2_MPLT1">PROJ2_FOLD2_MPLT1_VAL</parameter>
    </maplet>
    <folder name="Folder2_1">
      <folder name="Folder2_1_1">
        <maplet name="RULE1">
          <parameter name="PROJ2_RULE1">PROJ2_RULE1_VAL</parameter>
        </maplet>
      </folder>
    </folder>
  </folder>
</project>
</root>
```

パラメータファイルの作成

infacmd ms ListMappingParams コマンドを実行して、デプロイ済みアプリケーションの mapping で使用されるパラメータと各パラメータのデフォルト値をリスト表示します。このコマンドの出力を使用して、パラメータファイルを作成します。

1. infacmd ms ListMappingParams コマンドを実行して、mapping で使用されるパラメータと各パラメータのデフォルト値をリスト表示します。

-o 引数は、コマンド出力を XML ファイルに出力します。

例えば、次のコマンドでは、ワークフロー「MyWorkflow」のパラメータがファイル「MyOutputFile.xml」にリストされます。

```
infacmd wfs ListWorkflowParams -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -wf MyWorkflow -o MyOutputFile.xml
```

例えば、次のコマンドでは、マッピング「MyMapping」のパラメータがファイル「MyOutputFile.xml」にリストされます。

```
infacmd ms ListMappingParams -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -m MyMapping -o MyOutputFile.xml
```

データ統合サービスは、すべてのマッピングパラメータをデフォルト値とともにリスト表示します。

2. -o 引数を指定しない場合、コマンド出力を XML ファイルにコピーしてファイルを保存できます。
3. XML ファイルを編集し、パラメータのデフォルト値を、mapping の実行時に使用する値に置き換えます。
4. XML ファイルを保存します。

パラメータファイルを使用した Mapping の実行

infacmd ms RunMapping コマンドを使用して、パラメータファイルを指定して mapping を実行します。-pf 引数でパラメータファイル名を指定します。

例えば、次のコマンドでは、ワークフロー「MyWorkflow」がパラメータファイル「MyParamFile.xml」を使用して実行されます。

```
infacmd wfs StartWorkflow -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -wf MyWorkflow -pf MyParamFile.xml
```

例えば、次のコマンドでは、マッピング「MyMapping」がパラメータファイル「MyParamFile.xml」を使用して実行されます。

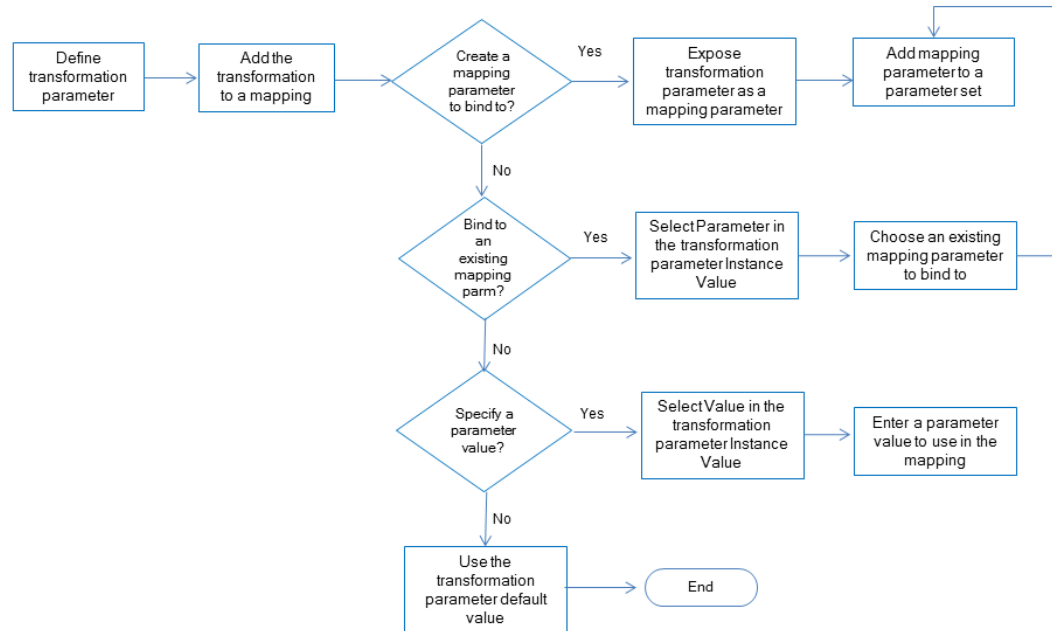
```
infacmd ms RunMapping -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -m MyMapping -pf MyParamFile.xml
```

パラメータファイルを使用してマッピングを実行し、そのファイルが無効であった場合、データ統合サービスは、mapping を失敗にします。パラメータファイルが見つからないまたはアクセスできない場合、データ統合サービスは、mapping を失敗にします。

パラメータの設定方法

トランスフォーメーション、マッピング、マップレット、またはワークフローでパラメータを定義します。

次の図は、マッピング内の再利用可能なトランスフォーメーションでパラメータを使用するためのプロセスを示しています。



1. 再利用可能なトランスフォーメーションで、トランスフォーメーション内のプロパティのパラメータ、または式エディタ内の変数のパラメータを作成します。
2. トランスフォーメーションをマッピングまたはマップレットに追加します。
3. トランスフォーメーションの **【パラメータ】** タブで、マッピングまたはマップレットでパラメータ値を設定する方法を選択します。
 - トランスフォーメーションパラメータをマッピングパラメータとして公開します。マッピングレベルで、トランスフォーメーションパラメータの複製を作成します。
 - トランスフォーメーションパラメータをマッピングパラメータにバインドします。マッピングパラメータを参照するか手動で作成して、トランスフォーメーションパラメータにバインドします。
 - 特定のパラメータ値を入力します。マッピング実行で使用するデフォルト値を入力します。
 - トランスフォーメーションパラメータのデフォルト値を使用します。マッピング内で元のパラメータ値を使用します。

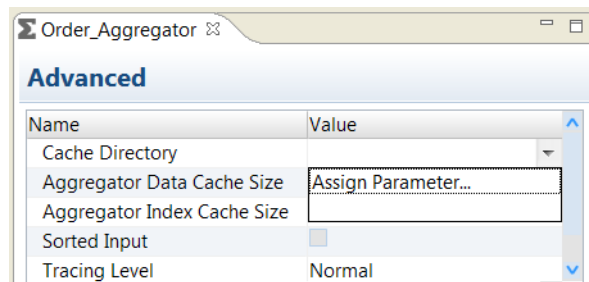
マッピングパラメータをトランスフォーメーションパラメータにバインドしたら、パラメータセットを作成してマッピングパラメータの値を実行時にオーバーライドできます。 コマンドラインからマッピングを実行し、そのマッピング実行で使用するパラメータセットを指定します。

トランスフォーメーションプロパティのパラメータの作成

フィールドまたはトランスフォーメーションプロパティにパラメータを割り当てる場合、使用するパラメータを参照するか、該当のフィールド専用のパラメータを作成することができます。

1. 更新するフィールドまたはプロパティに移動します。
2. **【値】** カラムで選択矢印をクリックします。

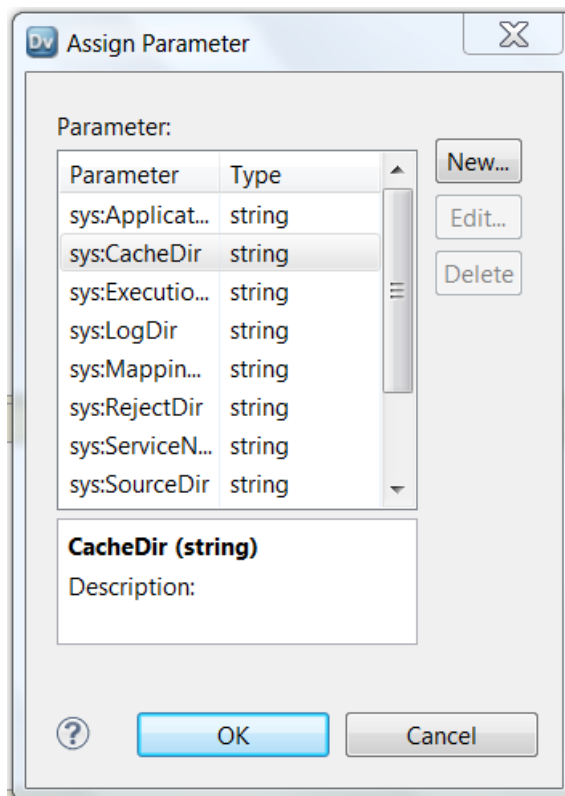
プロパティをパラメータ化できる場合、**【パラメータの割り当て】** オプションが表示されます。次の図は、キャッシュディレクトリの **【パラメータの割り当て】** オプションを示しています。



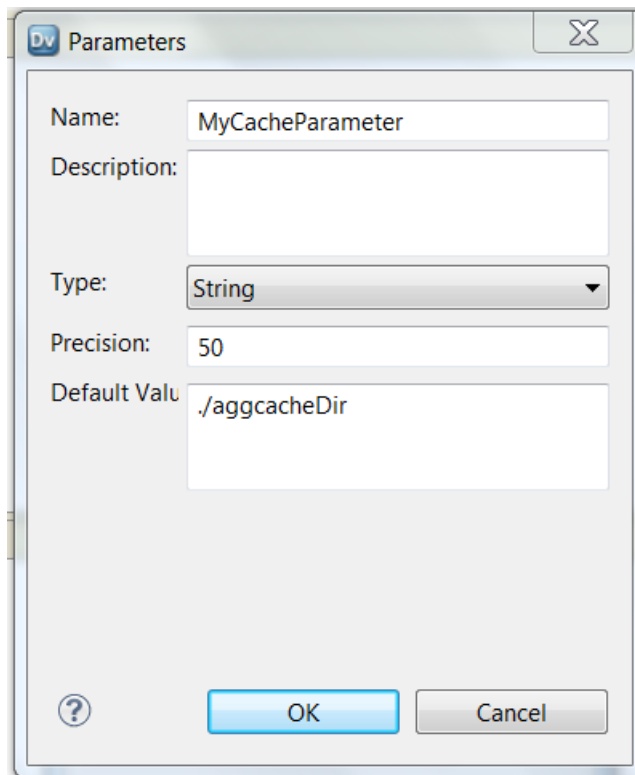
3. **【パラメータの割り当て】** をクリックします。

【パラメータの割り当て】 ダイアログボックスが表示されます。ダイアログボックスには、システムパラメータと、トランスフォーメーション内で作成したユーザー定義のパラメータが表示されます。

次の図は、**【パラメータの割り当て】** ダイアログボックスを示しています。



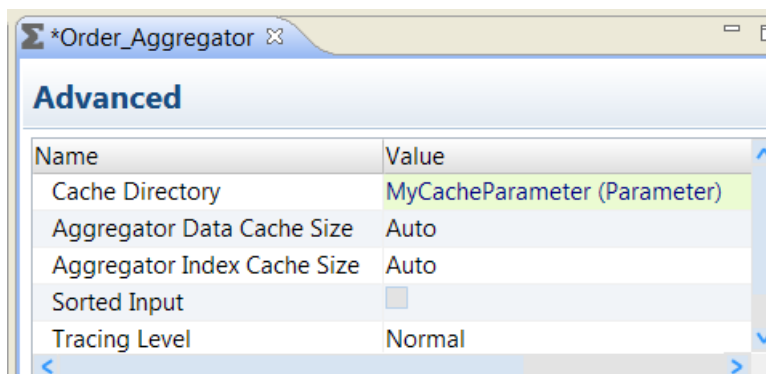
4. パラメータを作成するには、**【新規】** をクリックします。
 5. パラメータ名、タイプ、精度、およびデフォルト値を入力します。
- 次の図は、**【パラメータ】** ダイアログボックス内の MyCacheParameter というパラメータを示しています。



6. **[OK]** をクリックします。

トランスフォーメーションプロパティにパラメータ名が表示されます。

次の図は、アグリゲータトランスフォーメーションのキャッシュディレクトリ内の MyCacheParameter を示しています。



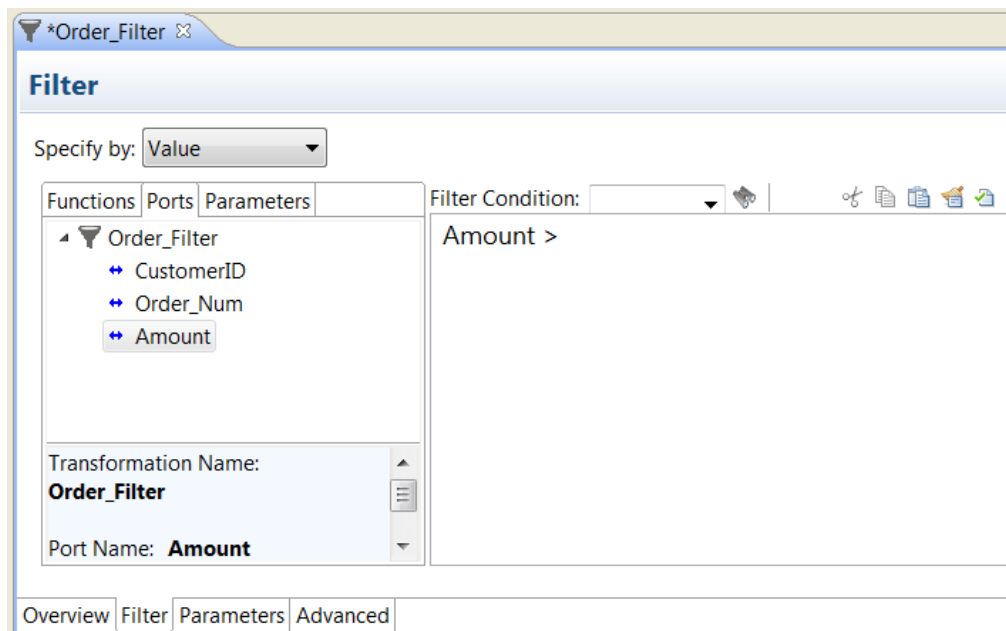
トランスフォーメーションの **[パラメータ]** タブでは、パラメータの追加、変更、削除ができます。

式内のパラメータの作成

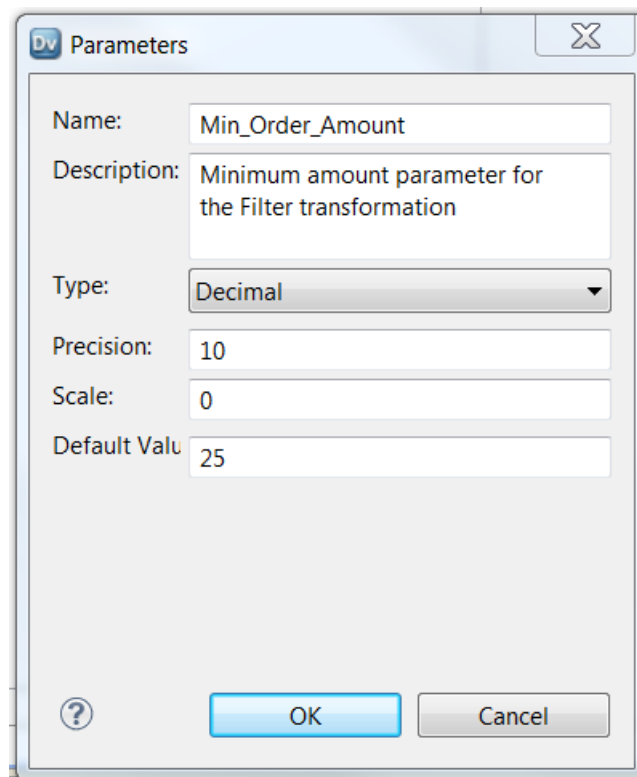
パラメータを定義したら、式内でパラメータを参照できます。次の例は、フィルタ式のコンポーネントにパラメータを割り当てる方法を示しています。

1. フィルタトランスフォーメーションで、**[フィルタ]** タブをクリックします。
式エディタが表示されます。関数、ポート、およびパラメータを選択して式を作成できます。
2. 式パラメータを使用する代わりに式を定義するには、**[指定元:]** で **[値]** を選択します。

3. [フィルタ] タブで、[ポート] タブをクリックします。
 4. Amount ポートを選択します。 [関数] タブで、「より大きい (>)」関数を選択します。
- 次の図は、Amount ポートと「>」演算子を含む式を示しています。



5. 式エディタの [パラメータ] タブをクリックします。
式エディタには、システムパラメータとユーザー定義のパラメータのリストが表示されます。
 6. [パラメータの管理] をクリックしてパラメータを追加します。
[パラメータ] ダイアログボックスが表示されます。
 7. [新規] をクリックします。
デフォルトのパラメータ値でダイアログボックスが表示されます。
 8. パラメータ名、パラメータタイプ、精度、およびデフォルト値を入力します。
- 次の図は、[パラメータ] ダイアログボックスを示しています。

A screenshot of a 'Parameters' dialog box. The dialog has a title bar with a 'Dv' icon and a close button. It contains several input fields: 'Name' with the value 'Min_Order_Amount', 'Description' with the text 'Minimum amount parameter for the Filter transformation', 'Type' with a dropdown menu set to 'Decimal', 'Precision' with the value '10', 'Scale' with the value '0', and 'Default Value' with the value '25'. At the bottom, there is a help icon (question mark), an 'OK' button, and a 'Cancel' button.

Parameters

Name: Min_Order_Amount

Description: Minimum amount parameter for the Filter transformation

Type: Decimal

Precision: 10

Scale: 0

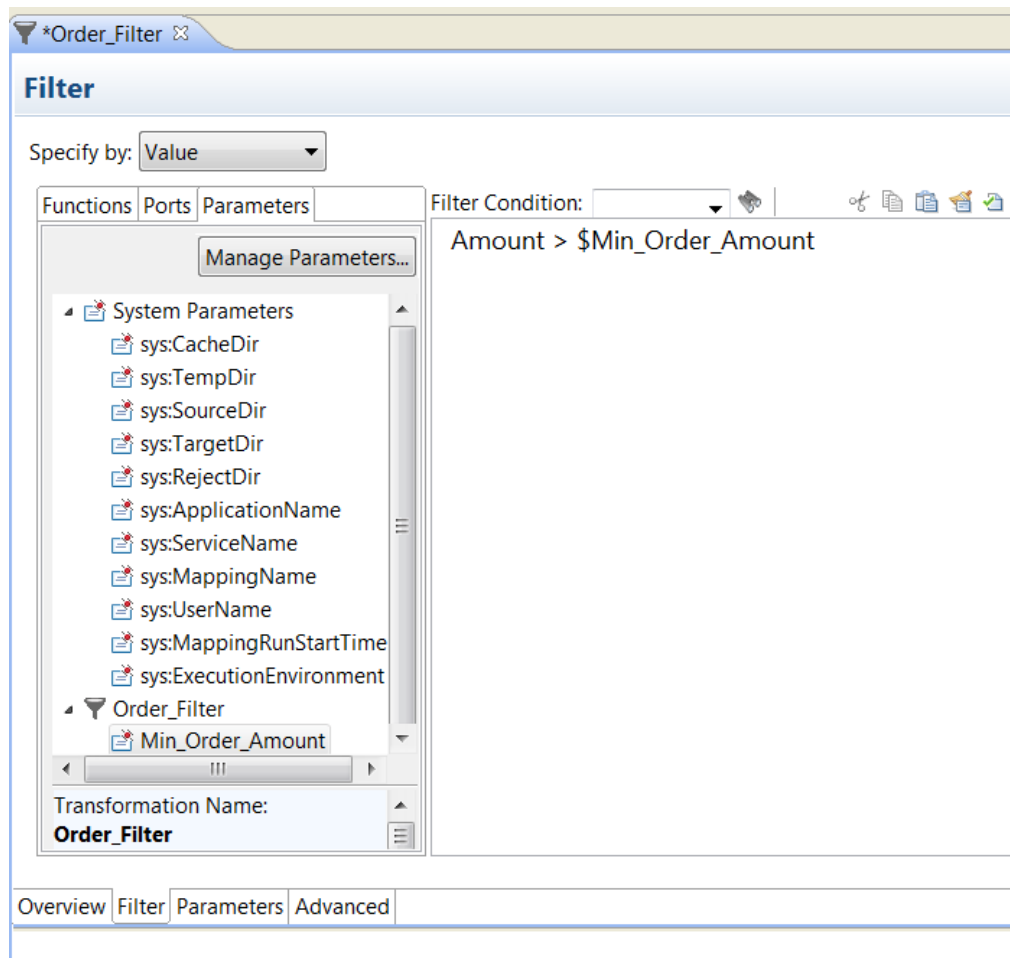
Default Value: 25

? OK Cancel

9. 式エディタで、**[OK]** をクリックします。
作成したパラメータがパラメータリストに表示されます。

10. Min_Order_Amount パラメータを選択して式に追加します。

Min_Order_Amount パラメータが式に表示されます。



パラメータは、式内ではドル記号（\$）識別子付きで表示されます。Min_Order_Amount のデフォルト値は 50 です。Min_Order_Parameter をオーバーライドせずにトランスフォーメーションをマッピングに追加した場合、フィルタトランスフォーメーションは、50 より大きい金額を持つ行を返します。

マッピングパラメータとしてトランスフォーメーションパラメータを公開する

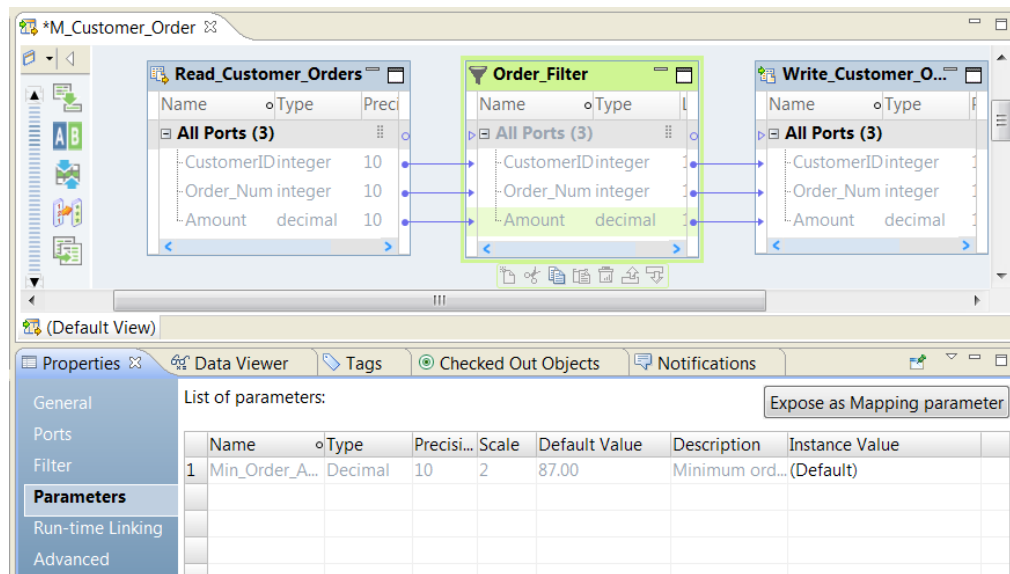
マッピングにトランスフォーメーションを追加したら、トランスフォーメーションパラメータをマッピングパラメータとして公開できます。トランスフォーメーションパラメータをマッピングパラメータとして公開する場合は、トランスフォーメーションパラメータの複製であるマッピングパラメータを作成します。

パラメータのインスタンス値は、特定のマッピングまたはマップレットで使用するパラメータ値です。トランスフォーメーションパラメータをマッピングパラメータとして公開する場合、マッピングパラメータを作成してトランスフォーメーションパラメータにバインドします。これを 1 回の手順で行うことができます。パラメータセットまたはパラメータファイルを使用して、マッピングパラメータの値を実行時に設定することができます。

1. マッピングを開きます。パラメータが含まれるトランスフォーメーションをクリックします。
トランスフォーメーションの【プロパティ】ビューが開きます。

2. **【パラメータ】** タブをクリックします。

次の図は、フィルタトランスフォーメーションの **【パラメータ】** タブを示しています。



3. パラメータのマッピングパラメータを作成するには、パラメータを選択し、**【マッピングパラメータとして公開】** をクリックします。

Developer tool は、同じ名前で作成してトランスフォーメーションパラメータにバインドします。

4. マッピングパラメータを更新するには、マッピングの **【プロパティ】** ビューの **【パラメータ】** タブを開きます。

デフォルトのマッピングパラメータ値を変更できます。 **【パラメータ】** タブでは、マッピングパラメータを追加することもできます。

パラメータインスタンス値の設定

トランスフォーメーションの **【パラメータ】** タブの **【インスタンス値】** カラムで、パラメータのインスタンス値を設定できます。重複したマッピングパラメータを作成しない場合は、このカラムでインスタンス値を設定します。

トランスフォーメーションパラメータをデフォルト値に設定したり、既存のマッピングパラメータをトランスフォーメーションパラメータにバインドしたりできます。

1. トランスフォーメーションをマッピングに追加したら、トランスフォーメーションの **【プロパティ】** ビューで **【パラメータ】** タブをクリックします。
2. マッピングパラメータをトランスフォーメーションパラメータにバインドするには、以下の手順を実行します。
 - a. トランスフォーメーションパラメータの **【インスタンス値】** カラムをクリックします。
【指定元】 ダイアログボックスが表示されます。
 - b. **【指定元:】** で **【パラメータ】** をクリックします。
 - c. **【パラメータの割り当て】** ダイアログボックスで、マッピングパラメータまたはシステム定義のパラメータを参照して選択し、トランスフォーメーションパラメータにバインドします。
 - d. **【OK】** をクリックします。

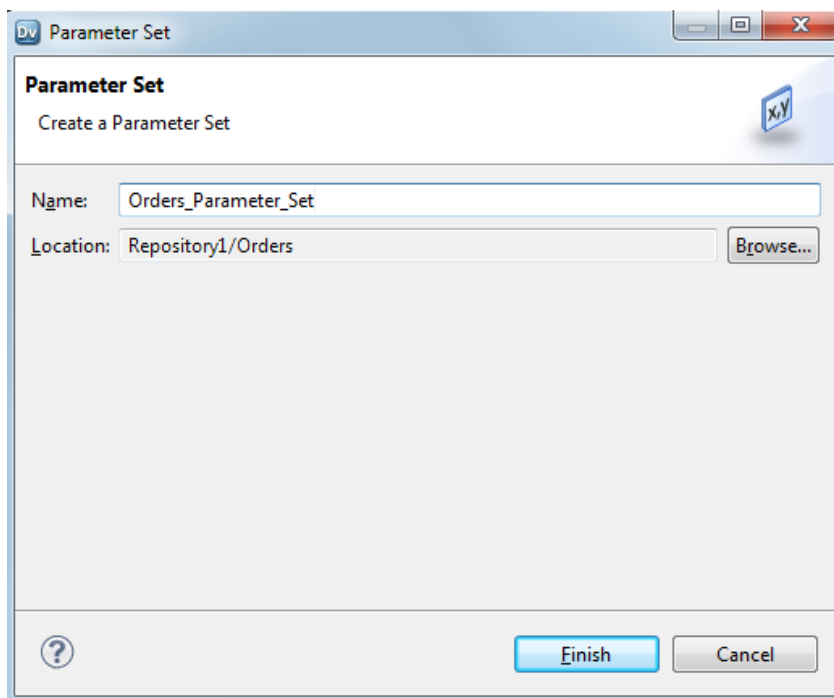
- 【指定元】** ダイアログボックスに、マッピングパラメータ名がパラメータ値として表示されます。
- e. **【指定元】** ダイアログボックスで **【OK】** をクリックします。
- 【インスタンス値】** カラムにマッピングパラメータ名が表示されます。
3. トランスフォーメーションパラメータインスタンスのデフォルト値を設定するには、以下の手順を実行します。
- トランスフォーメーションパラメータの **【インスタンス値】** カラムをクリックします。
【指定元】 ダイアログボックスが表示されます。
 - デフォルト値を入力するには、**【指定元:】** で **【値】** をクリックし、インスタンスのデフォルト値を入力します。
 - トランスフォーメーションパラメータのデフォルト値を使用するには、**【デフォルトを使用】** をクリックします。

パラメータセットの作成

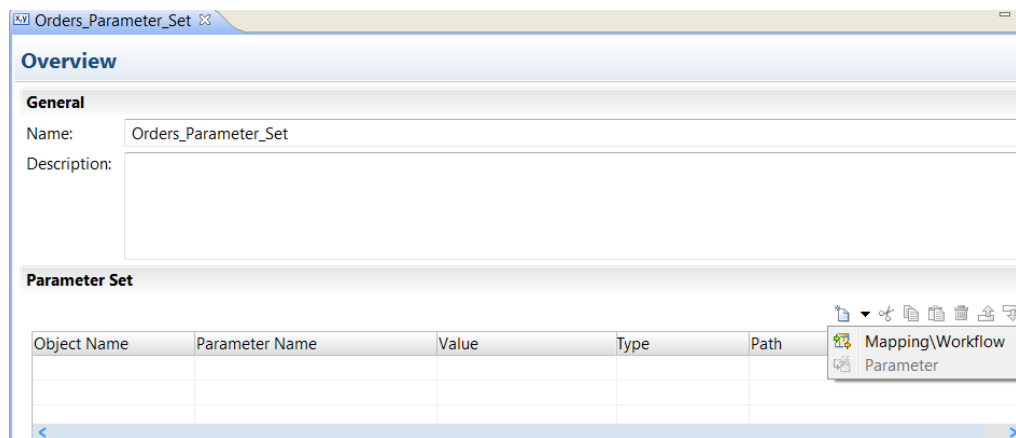
マッピングおよびワークフローのランタイムコンテキスト変更に使用できるパラメータセットを作成します。

パラメータセットを作成する場合、パラメータを含めるためのマッピングまたはワークフローを選択します。マッピングまたはワークフローを選択後、パラメータセットに手でパラメータを入力するか、パラメータを選択できます。

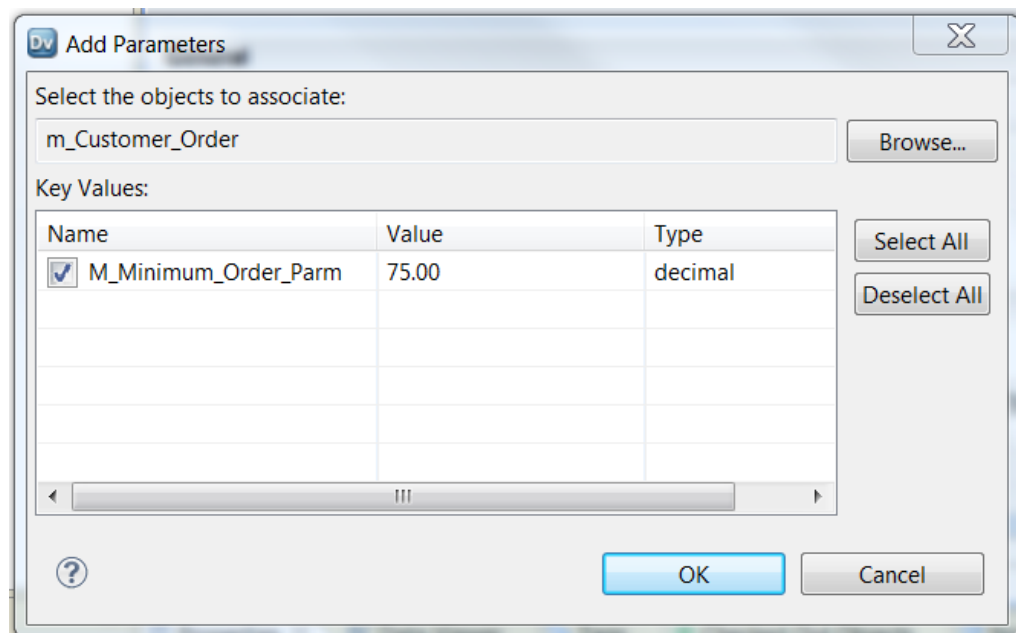
- 【オブジェクトエクスプローラ】** ビューで、プロジェクトを右クリックし、**【新規】** > **【パラメータセット】** をクリックします。
- パラメータセット名を入力して、**【完了】** をクリックします。



- 【プロパティ】** パネルをドラッグしてグリッドを表示し、パラメータをパラメータセットに追加します。
- 【新規】** > **【マッピング/ワークフロー】** をクリックします。



5. **パラメータの追加**ダイアログボックスで、**[参照]**をクリックし、セットに含める必要があるパラメータを含むマッピングまたはワークフローを見つけます。
マッピングおよびワークフローのリストが表示されます。
6. マッピングまたはワークフローを選択し、**[OK]**をクリックします。
マッピングまたはワークフローのパラメータのリストが表示されます。



7. パラメータセットに含めるパラメータを選択し、**[OK]**をクリックします。
パラメータセットにマッピングまたはワークフロー名とパスが表示されます。選択した各パラメータがオブジェクトの下に表示されます。

Overview

General

Name: Orders_Parameter_Set

Description:

Parameter Set

Object Name	Parameter Name	Value	type	Path
m_Customer_Order	M_Minimum_Order_Parm	75.00	decimal	Orders\m_Customer_Order

8. ワークフローまたはマッピングにないパラメータを追加するには、マッピングまたはオブジェクト名を右クリックして【パラメータ】挿入を選択します。

Developer tool によりワークフローまたはマッピングの下にパラメータが作成されます。パラメータ名、値、およびタイプを変更します。

注: パラメータセットを使用する前にマッピングまたはワークフローにパラメータを追加する必要があります。

第 4 章

マッピング出力

この章では、以下の項目について説明します。

- [マッピング出力の概要, 82 ページ](#)
- [ユーザー定義マッピング出力, 83 ページ](#)
- [システム定義のマッピング出力, 86 ページ](#)
- [保持されたマッピング出力, 87 ページ](#)
- [マッピング出力のワークフロー変数へのバインド, 89 ページ](#)
- [マップレットのマッピング出力, 90 ページ](#)
- [論理データオブジェクトのマッピング出力, 93 ページ](#)
- [マッピング出力の設定方法, 93 ページ](#)
- [マップレット出力をマッピング出力にバインドする方法, 102 ページ](#)

マッピング出力の概要

マッピングはマッピング出力を返すことができます。マッピング出力は、マッピングで処理する各行のフィールドまたは式を集計した結果である単一値です。

マッピング出力は、マッピングの実行に関する情報を提供する値を返します。例えば、マッピング出力ではマッピングで見つかったエラー行の数を返すことができます。マッピング出力では、マッピングで処理された最後の注文日、およびすべての注文の合計数を返すことができます。

トランスフォーメーションはマッピング出力値を受け取りません。マッピングは、マッピングが完了すると各マッピング値を返します。マッピング出力はワークフローの他のタスクに渡すことができます。マッピングが次回実行されるときに入力パラメータとして使用するために値を保存できます。複数のマッピング出力を同一のマッピングに定義することができます。

マッピングでは、ユーザー定義マッピング出力またはシステム定義マッピング出力を返すことができます。

ユーザー定義マッピング出力

ユーザー定義マッピング出力は、マッピングの各行のフィールドまたは式を集計することによってマッピングが返す数値または日付です。例えば、注文が特定のしきい値に達したときを認識する必要がある場合があります。マッピングが処理した注文の合計数を返すように、マッピングを設定できます。TotalOrderAmt というマッピング出力を定義して、すべての行の Order_Amount フィールドを集計するようにマッピングを設定します。式またはポート名を定義して、式トランスフォーメーションで集計します。

システム定義のマッピング出力

システム定義のマッピング出力は、マッピングの完了時にマッピングが必ず返す組み込み値です。マッピングは、マッピングが処理したソース行数、ターゲット行数、およびエラー行数を返します。ワークフロー変数のこれらの値はワークフローの別のタスク（通知タスク、排他的なゲートウェイタスクなど）に渡すことができます。システム定義のマッピング出力を定義する必要はありません。

マッピング出力に関する次のタスクを実行します。

マッピング出力のリポジトリへの保存

マッピング出力値をリポジトリ内に保持するように、マッピングタスクを設定できます。保持されたマッピングの出力値を、マッピングタスクの入力に割り当てることができます。例えば、マッピングで生成された最新のシーケンス番号を返すようにマッピングを設定できます。Last_Seq_Num マッピング出力をリポジトリ内に保持します。マッピングが次回実行されるときに、Last_Seq_Num を開始シーケンス番号として使用できます。

出力のワークフロー変数へのバインド

マッピング出力をワークフロー変数にバインドして、値をワークフローの他のタスクに渡すことができます。現在のマッピングタスク実行のマッピング出力を、ワークフロー変数にバインドできます。以前のマッピングタスク実行で保持されたマッピング出力を、現在の実行のワークフロー変数にバインドすることもできます。

ユーザー定義マッピング出力

ユーザー定義マッピング出力は、マッピングの各行のフィールドまたは式を集計することによってマッピングが返す数値または日付です。集計する式、および結果値のデータ型を定義します。

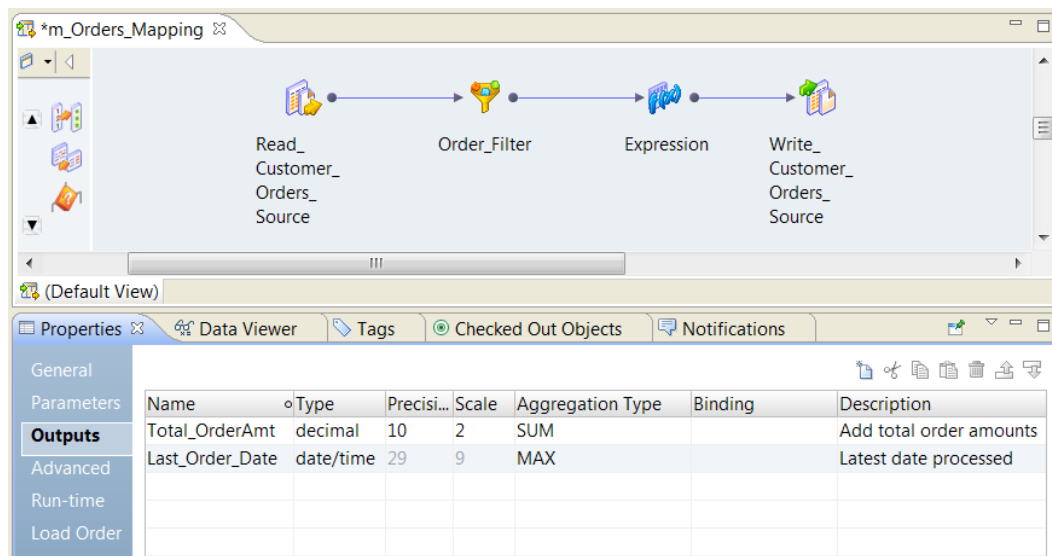
[プロパティ] ビューの **[出力]** タブでマッピング出力を定義します。マッピング出力の名前および結果のデータ型を設定し、結果を返すために実行する集計のタイプを指定します。

マッピング出力名および出力タイプを定義したら、マッピングに式トランスフォーメーションを設定します。式トランスフォーメーションで、集計する出力式を定義します。式にはポート名を含めるか、ポート、関数、およびパラメータを持つ式を含めることができます。

[出力] ビュー

マッピング出力は、マッピングの **[プロパティ]** の **[出力]** ビューで定義します。各マッピング出力を定義するときに、マッピング出力名、マッピング出力タイプ、および実行する集計のタイプを入力します。

次の図は、マッピングの **[プロパティ]** ビューの **[出力]** タブのマッピング出力を示しています。



【出力】ビューには以下のフィールドがあります。

名前

出力の名前。デフォルトは「Output」です。

タイプ

マッピング出力のタイプ。numeric タイプまたは date/time タイプを選択できます。デフォルトは Integer です。

精度

マッピング出力フィールドの長さ。デフォルトは 10 です。

スケール

マッピング出力フィールドの小数点の右側の桁数。デフォルトはゼロです。

集計タイプ

以下のいずれかの集計タイプを選択できます。

SUM

データ統合サービスが処理した各入力行のフィールドまたは式の合計を返します。

MIN

データ統合サービスが各入力行の特定のフィールドまたは式から処理した最小の数値または日付を返します。

MAX

データ統合サービスが各入力行の特定のフィールドまたは式から処理した最大の数値または日付を返します。

バインディング

マッピング出力にバインドするマプレットまたは論理データオブジェクトの出力の名前。マッピング出力がマッピングの式トランスフォーメーションからではなくマプレットから返される場合を除き、このフィールドは空白です。

説明

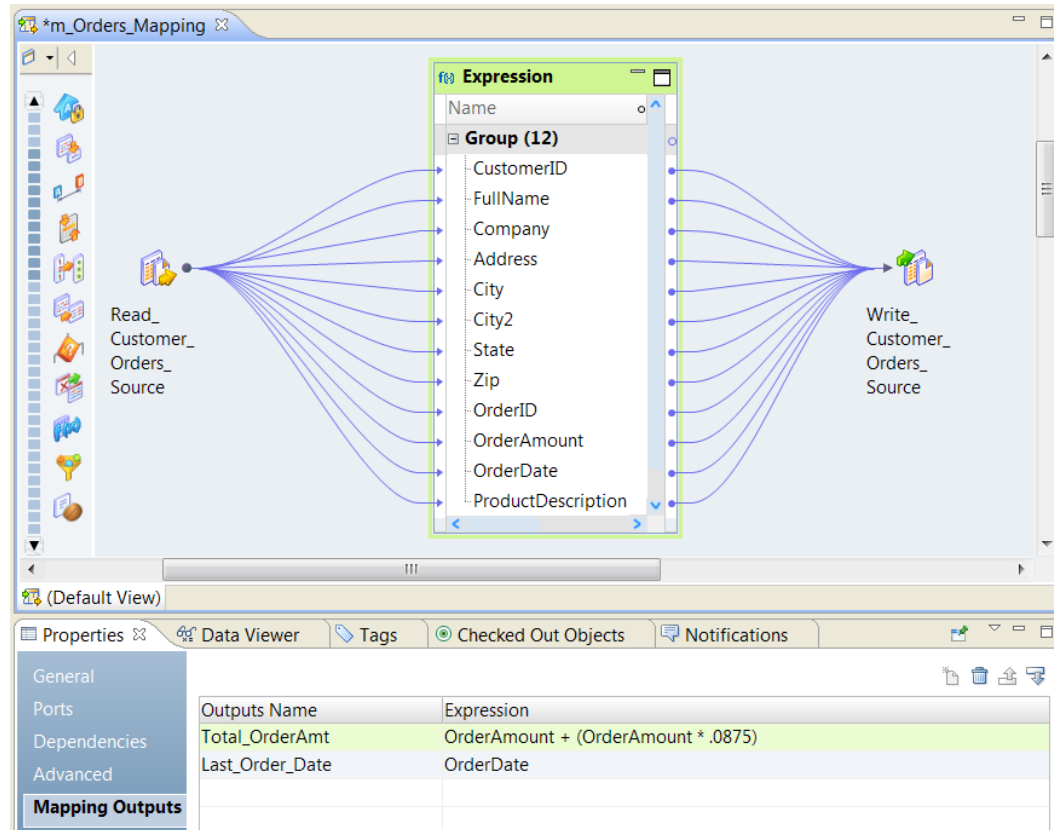
マッピング出力の説明。

マッピング出力式

マッピング出力式は、式トランスフォーメーションの【マッピング出力】ビューで設定します。マッピング出力式は、式トランスフォーメーションが受け取る行から集計するフィールドまたは式です。

マッピングに式トランスフォーメーションを設定して、集計する出力式を含めます。マッピングにフィルタまたはアクティブなトランスフォーメーションが含まれているかどうかによって、パイプラインでの式トランスフォーメーションの位置がマッピング出力の結果に影響するかどうかが決まります。異なるパイプラインの行を集計する必要がある場合は、複数の式トランスフォーメーションをマッピングに追加できます。

次の図は、式トランスフォーメーションの【マッピング出力】ビューの式を示しています。



【マッピング出力】ビューには以下のフィールドがあります。

出力名

マッピングレベルで作成したマッピング出力の名前。マッピング出力は最初にマッピングレベルで作成する必要があります。マッピング出力を式トランスフォーメーションに追加する場合は、すでに作成した出力のリストから出力名を選択します。

式

マッピングの各行を集計するための式。式エディタにポート名または式を入力します。式の結果は数値または日付である必要があります。式にパラメータを使用できます。データ統合サービスは、式トランスフォーメーションが受け取る各行に式を適用します。各マッピング出力は、マッピングが完了すると1つの値を返します。

注: 式トランスフォーメーションには、実行する集計のタイプは指定しません。マッピングが各行を処理する際に集計するフィールドまたは式を示します。

システム定義のマッピング出力

システム定義のマッピング出力は、各マッピングで生成されるマッピング出力です。システム定義のマッピング出力には集計を設定する必要はありません。システム定義のマッピング出力はワークフロー変数に渡すことができます。

マッピングは、次のタイプのシステム定義マッピング出力を返します。

numberOfTargetRows

マッピングがターゲットに書き込んだ行の数。

numberOfSourceRows

マッピングがソースから読み取った行の数。

numberOfErrorRows

マッピングによって生成されたエラー行の数。

ワークフロー変数を設定し、マッピングタスクの【プロパティ】ビューの【出力】タブでシステム定義マッピング出力を割り当てます。

次の図は、【出力】タブのシステム定義マッピング出力を示しています。

The screenshot shows the SAP Workflow Designer interface. The top pane displays the workflow diagram with a task named 'm_Orders_Mapping' between 'Start_Event' and 'End_Event'. The bottom pane shows the 'Properties' view for the 'm_Orders_Mapping' task, with the 'Output' tab selected. The 'Output' tab displays a table titled 'Assign task output to workflow variables:'.

Output	Variable
System Defined Mapping Outputs	
Number of Target Rows Processed	
Number of Source Rows Processed	
Number of Error Rows	
User Defined Mapping Outputs	
Persisted Value	
Last_Order_Date	
Current Value	
Total_OrderAmt	wf_Variable_Total_OrderAmt (Variable)
Last_Order_Date	

保持されたマッピング出力

マッピングをワークフローで実行する場合は、マッピング出力をリポジトリに保存できます。マッピング出力を、同じマッピングタスクの後続の実行で使用できます。以前のマッピングタスク実行で保持されたマッピング出力を、現在のマッピングタスク実行のワークフロー変数に割り当てすることもできます。

マッピングタスクのマッピング出力を保持します。マッピングタスクは、マッピングの設定およびパラメータのバインドを含む、ワークフロー内のマッピングのインスタンスです。マッピングタスクの詳細については、『*Informatica Developer ワークフローガイド*』を参照してください。

マッピングタスクの【プロパティ】ビューの【持続】タブをクリックすると、Developer tool にマッピングのすべてのマッピング出力が表示されます。マッピング出力を保持するには、マッピング出力の【持続】を有効にして、保持された値を返すために実行する集計のタイプを選択します。

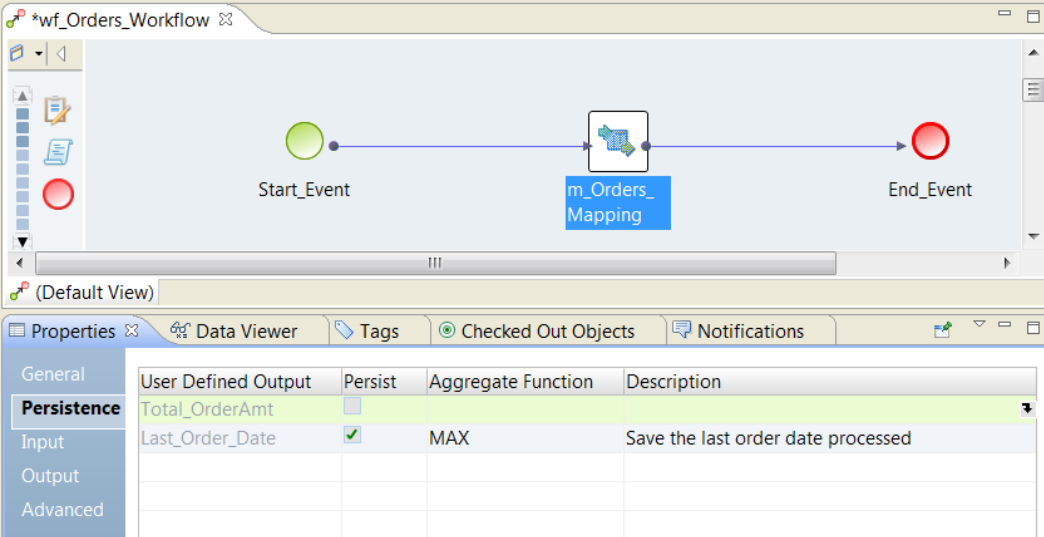
データ統合サービスがモデルリポジトリでマッピング出力を保持する場合、データ統合サービスはマッピングタスク名をキーとしてマッピング出力を保存します。例えば、ワークフローに4つのマッピングタスクが含まれ、それぞれが同じマッピングを実行している場合、データ統合サービスはモデルリポジトリ内に4つの出力を保存します。

マッピング出力を保持すると、マッピングレベルで定義した集計関数とは異なる集計関数を保持された値について設定できます。データ統合サービスは複数のマッピング出力値を生成します。例えば、OrderDate マッピング出力には MIN OrderDate が含まれていることがあります。保持された OrderDate マッピング出力には、MAX OrderDate が含まれていることがあります。

マッピングタスクのマッピング出力は、マッピングタスクが次回実行される時に入力パラメータにバインドできます。フィードバックバインディングは、1つのマッピング実行の結果を同じマッピングの次回の実行の入力として設定する方法です。フィードバックバインディングに使用するには、マッピング出力をマッピングタスクに保持する必要があります。

フィードバックバインディングの1つの例は、マッピングが処理した最後の注文日を保持する場合です。マッピングタスクが次回実行される時、マッピングの入力パラメータは最後に処理された日付になります。マッピングでは、最後に処理された注文日より後の注文日を持つ行を含めるように、パラメータソース行をフィルタ処理できます。

次の図は、マッピングタスクの【プロパティ】ビューの【持続】タブを示しています。



User Defined Output	Persist	Aggregate Function	Description
Total_OrderAmt	<input checked="" type="checkbox"/>		
Last_Order_Date	<input checked="" type="checkbox"/>	MAX	Save the last order date processed

【持続】タブには以下のフィールドがあります。

ユーザー定義の出力

マッピングが返すマッピング出力の名前。

持続

データ統合サービスがマッピング出力をリポジトリに保持するようにします。

集計関数

保持するマッピング出力で実行される集計のタイプ。MIN、MAX、または SUM を選択します。デフォルトは、マッピングプロパティに定義したマッピング出力の値です。保持されたマッピング出力の集計関数のタイプは変更できます。ワークフロー変数に渡すマッピング出力値とは異なる値をリポジトリに保持できます。

説明

リポジトリに保持するマッピング出力について説明します。

保持された値の保守

リポジトリ内の保持されたマッピング出力を一覧表示、更新、およびリセットできます。

保持されているマッピングタスク値に対して、次の `infacmd` コマンドを実行できます。

`listMappingPersistedOutputs`

ワークフロー内のマッピングタスクインスタンスについて、保持されたマッピング出力およびそれらの値を一覧表示します。

`setMappingPersistedOutputs`

ワークフロー内の特定のマッピングタスクインスタンスについて、保持されたマッピング出力を更新またはリセットします。値をリセットすると、保持された値がリポジトリから削除されます。マッピング出力を設定するには、コマンドラインで、マッピング出力の名前と値のペアをスペースで区切って入力します。マッピング出力をリセットするには、マッピング出力のスペース区切りのリストとともに `-reset` オプションを使用します。

`infacmd` の詳細については、『*Informatica コマンドリファレンス*』を参照してください。

保持されたマッピング出力およびデプロイメント

ワークフローを再デプロイするか、マッピング出力を変更すると、保持されたマッピング出力の状態に影響することがあります。

保持されたマッピング出力のルールおよびガイドラインは、次のとおりです。

- ワークフローをアプリケーションとして初めてデプロイする際に、マッピングタスクに保持されたマッピング出力がある場合は、追加のタスクを実行する必要はありません。
- アプリケーションを再デプロイする場合は、状態情報の保持または破棄を選択できます。状態情報を保持することを選択した場合は、アプリケーションを再デプロイするときに、マッピング出力値はリポジトリで変更されません。それ以外の場合、マッピング出力の状態は持続から削除されます。
- マッピングまたはワークフローをバックアップおよびリストアするときに、マッピング出力の状態はバックアップされません。
- マッピング出力を名前変更または再作成した場合は、以前のワークフローの実行で保持されたマッピング出力値を使用できません。

マッピング出力のワークフロー変数へのバインド

マッピングをワークフローに追加したら、マッピング出力をワークフロー変数にバインドできます。値はワークフローの他のタスクに渡すことができます。

例えば、データ統合サービスがマッピング出力値を評価してから、次に実行するオブジェクトを決定することができます。または、データ統合サービスが次のタスクのフィールドでマッピング出力値を使用することができます。

マッピング出力を別のタスクで使用するには、マッピングタスクの【出力】ビューでマッピング出力をワークフロー変数にバインドします。

注: マッピング出力をワークフロー変数に割り当てたが、マッピングが行を処理しない場合、出力は NULL になります。マッピングタスクによってワークフロー変数の値は変更されることはありません。変数は、マッピングタスクが実行される前と同じ値を保持します。

次の図は、マッピングタスクの【出力】ビューを示しています。

Output	Variable
System Defined Mapping Outputs	
Number of Target Rows Processed	
Number of Source Rows Processed	
Number of Error Rows	
User Defined Mapping Outputs	
Persisted Value	
Last_Order_Date	
Current Value	
Total_OrderAmt	wf_Variable_Total_OrderAmt (Variable)
Last_Order_Date	

【出力】カラムには、次のタイプのマッピング出力が含まれています。

システム定義のマッピング出力

トランスフォーマーがマッピングに返す組み込みのマッピング出力。システム定義マッピング出力には、マッピングが処理したソース行数、ターゲット行数、およびエラー行数が含まれています。

ユーザー定義マッピング出力

保持されたマッピング出力値と現在のマッピング出力値を、ワークフロー変数にバインドできます。

保持された値

以前のワークフローの実行のユーザー定義マッピング出力値。保持された値は、最後のマッピングタスク実行でリポジトリに保持された値です。保持された値は、現在のマッピングで集計された値ではありません。

現在の値

現在のマッピングタスクのユーザー定義マッピング出力値。

ワークフロー変数の詳細については、『*Informatica Developer ワークフローガイド*』を参照してください。

マップレットのマッピング出力

マッピング出力を返すようにマップレットを設定できます。マップレットのマッピング出力を、マッピング出力にマッピングレベルでバインドできます。

マップレットをマッピングに含めると、マップレットは出力の値を計算して、出力値をマッピングに渡します。マップレットの複数の出力を、同一の出力にマッピングレベルでバインドできます。マップレットのシステム定義の出力をマッピング出力にバインドすることもできます。マップレット出力とマッピング出力は、同じタイプである必要があります。

例えば、マップレットは、Salary ポート、Bonus ポート、および Commission ポートの最大値を 3 つのマッピング出力として返すことができます。

次の図は、【出力】ビューの Out_Salary、Out_Bonus、および Out_Commission マッピング出力を示しています。

The screenshot displays the Informatica Developer interface for a workflow named 'Test_Compensation'. The main workspace shows three tables: 'NEWINPUT', 'Expression', and 'NEWOUTPUT'. Each table has columns for 'Name' and 'Type'. 'NEWINPUT' and 'NEWOUTPUT' both have a 'Group (5)' containing 'Employee' (decimal), 'Salary' (decimal), 'Bonus' (decimal), 'Commission' (decimal), and 'Date' (date/time). The 'Expression' table also has a 'Group (5)' with the same fields. Arrows indicate data flow from 'NEWINPUT' to 'Expression' and from 'Expression' to 'NEWOUTPUT'. Below the workspace, the 'Properties' pane is open to the 'Outputs' tab, showing a table of output fields.

Name	Type	Precisi...	Scale	Aggregation Type	Binding	Description
Out_Salary	decimal	10	0	MAX		Maximum salary paid
Out_Bonus	decimal	10	0	MAX		Maximum bonus paid
Out_Commission	decimal	10	0	MAX		Maximum commission paid

【出力】ビューには以下のフィールドがあります。

名前

出力の名前。デフォルトは「Output」です。

タイプ

マッピング出力のタイプ。numeric タイプまたは date/time タイプを選択できます。デフォルトは Integer です。

精度

マッピング出力フィールドの長さ。

スケール

マッピング出力フィールドの小数点の右側の桁数。

集計タイプ

出力式で実行される集計のタイプ。SUM、MIN、または MAX を選択します。デフォルトは SUM です。

バインディング

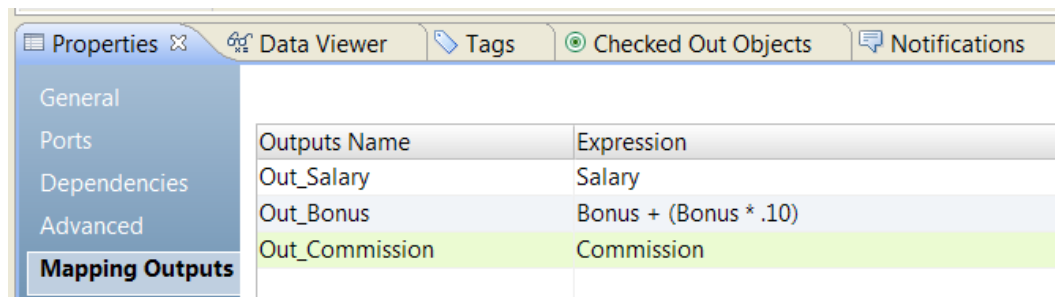
このマッピング出力にバインドする別のマップレットの出力の名前。マップレット内にマッピング出力を返す別のマップレットが含まれている場合を除き、【バインディング】フィールドは空です。

説明

マッピング出力の説明。

マップレット内の各マッピング出力について、関連付けられる出力式を式トランスフォーメーションに作成します。各式では集計するフィールドを指定します。

次の図は、式トランスフォーメーションのマッピング出力式を示しています。



Outputs Name	Expression
Out_Salary	Salary
Out_Bonus	Bonus + (Bonus * .10)
Out_Commission	Commission

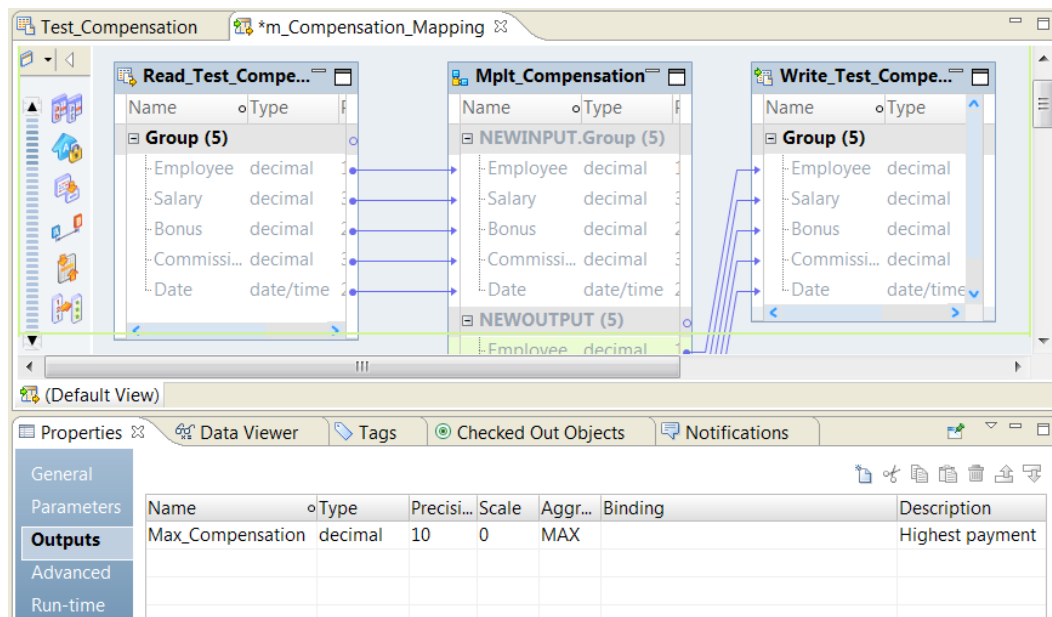
この例の場合、式トランスフォーメーションは Salary ポート値および Commission ポート値を集計します。Out_Bonus マッピング出力は、Bonus ポート値に Bonus の 10%を加えている式です。

マップレット出力のマッピング出力へのバインド

マップレットがマッピング出力を計算する場合は、マップレットの出力値をマッピングに渡す必要があります。

マッピングの【出力】ビューで、マップレット出力をマッピング出力にバインドします。

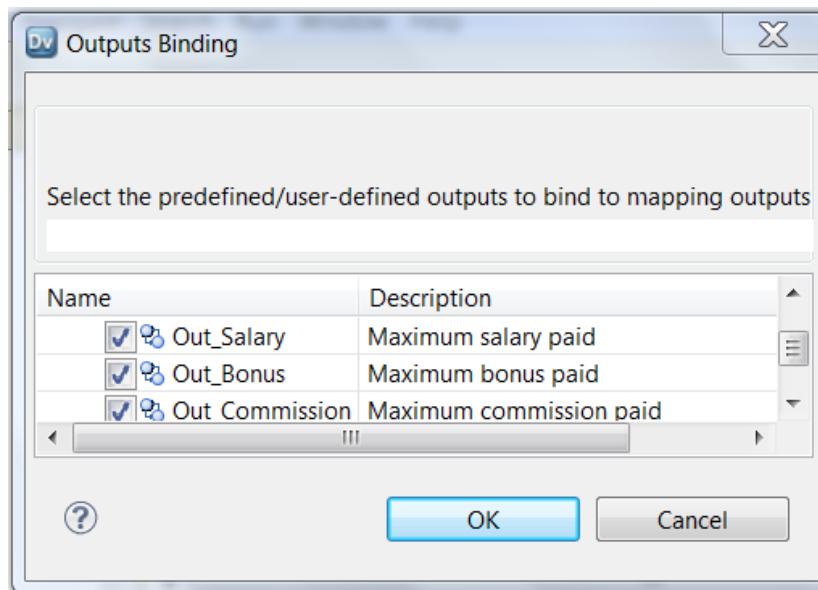
次の図は、マッピングレベルの Max_Compensation マッピング出力を示しています。



マッピングレベルで、Salary マップレット出力、Bonus マップレット出力、および Commission マップレット出力を、Max_Compensation という同一のマッピング出力にバインドできます。

マップレット出力をマッピング出力にバインドするには、マッピング出力の【バインディング】カラムをクリックします。使用可能なマップレット出力のリストが表示されます。リストには、マッピング出力と同じタイプおよび集計のマップレット出力が表示されます。マッピング出力に割り当てるマップレット出力を選択します。

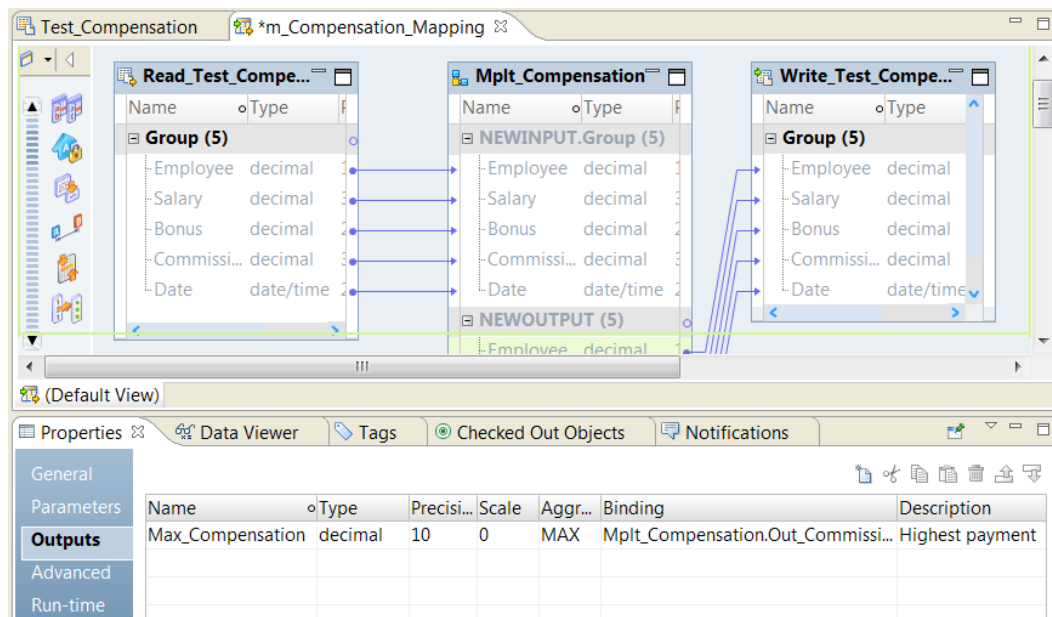
次の図は、【出力バインディング】ダイアログボックスを示しています。



Max_Compensation にバインドする出力を選択したら、【バインディング】フィールドに次のテキストが表示されます。

Mplt_Compensation.Out_Salary, Mplt_Compensation.Out_Bonus, Mplt_Compensation.Out_Commission

次の図は、【バインディング】フィールドのマッピング出力を示しています。



マッピング出力の集計タイプは MAX に設定します。データ統合サービスは、Salary、Bonus、または Commission ポートから見つかった最も高い報酬値を返します。

論理データオブジェクトのマッピング出力

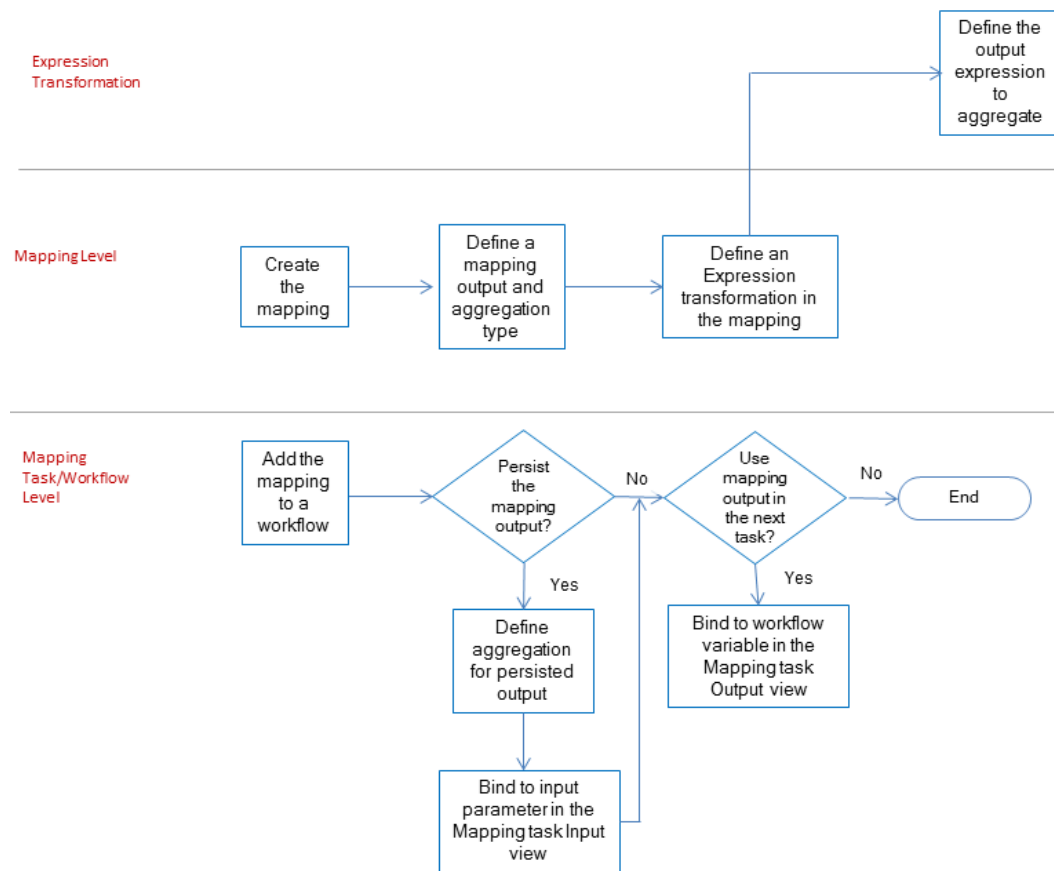
論理データオブジェクトには、読み取りマッピングまたは書き込みマッピングを含めることができます。マッピング出力を返すようにそれらのマッピングを設定できます。論理データオブジェクトのマッピング出力を、マッピングのマッピング出力にバインドできます。

論理データオブジェクトをマッピングに含めると、読み取りマッピングまたは書き込みマッピングはマッピング出力の値を計算します。論理データオブジェクトは出力値をマッピングに渡します。論理データオブジェクトのマッピングの複数の出力を、同一の出力にマッピングレベルでバインドできます。論理データオブジェクトのシステム定義の出力をマッピング出力にバインドすることもできます。論理データオブジェクトのマッピング出力とマッピング出力は、同じタイプである必要があります。

マッピング出力の設定方法

マッピング出力を設定する場合は、マッピングレベルでマッピング出力を定義し、トランスフォーメーションレベルで集計する式を設定して、マッピングタスクレベルで結果を保持します。

次の図は、マッピング出力を設定するプロセスを示しています。



マッピング出力を設定するには、以下の手順を実行します。

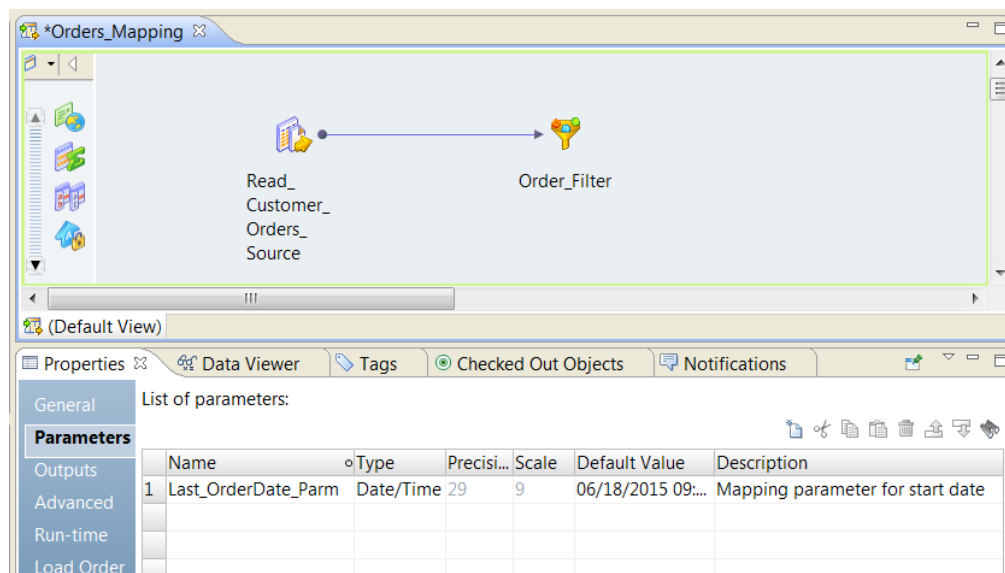
1. マッピングを作成します。
2. マッピングの【出力】ビューで、マッピング出力名および集計のタイプを定義します。
3. マッピングに式トランスフォーメーションを追加して、式の【マッピング出力】ビューでマッピング出力式を設定します。
4. マッピングタスクを作成するには、マッピングをワークフローに追加します。
5. マッピングタスクの【持続】ビューでマッピング出力を保持して、保持された値の集計関数タイプを設定します。
6. 保持されたマッピング出力を、マッピングタスクの入力パラメータに割り当てます。
7. マッピング出力を別のワークフロータスクで使用する場合は、マッピング出力をワークフロー変数に割り当てます。

マッピングの作成

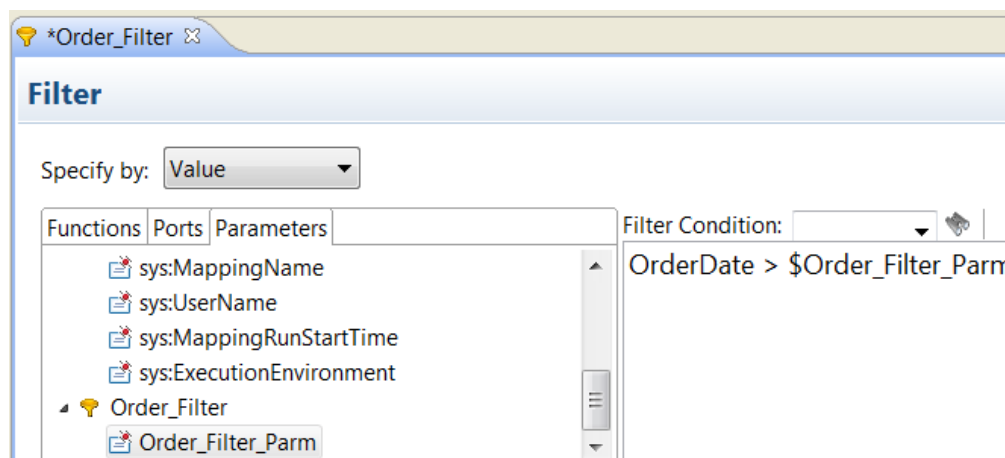
再利用可能なフィルタトランスフォーメーションを含むマッピングを作成します。フィルタトランスフォーメーションは、特定の日付より前の注文日を持つ行をフィルタ処理します。フィルタ式には、Last_Order_Date_Parm というパラメータが含まれています。

1. Customer_Order ファイルの注文データを処理するマッピングを作成します。
2. マッピングの【プロパティ】ビューで、【パラメータ】タブをクリックします。
3. Last_Order_Date_Parm という日付/時刻マッピングパラメータを追加します。

開始パラメータのデフォルト日付を入力します。
次の図は、マッピングパラメータを示しています。



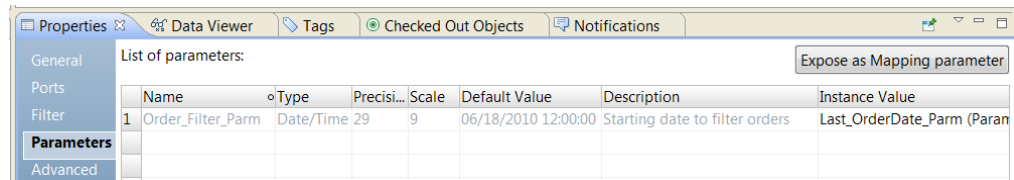
4. 再利用可能なフィルタトランスフォーメーションを作成して、Customer_Order の行をフィルタ処理します。
5. Order_Filter というフィルタトランスフォーメーションにパラメータを定義します。
開始パラメータのデフォルト日付を入力します。
6. パラメータより後の注文日を見つけるためのフィルタ式を追加します。



7. フィルタトランスフォーメーションをマッピングに追加します。
8. フィルタトランスフォーメーションをクリックして、トランスフォーメーションの【プロパティ】ビューを表示します。
9. 【パラメータ】タブをクリックします。
10. Order_Filter_Parm トランスフォーメーションパラメータを Last_Order_Date マッピングパラメータにバインドするには、Order_Filter_Parm の【インスタンス値】カラムをクリックします。

11. Last_Order_Date を選択します。

次の図は、マッピングパラメータがトランスフォーメーションパラメータにバインドされていることを示しています。



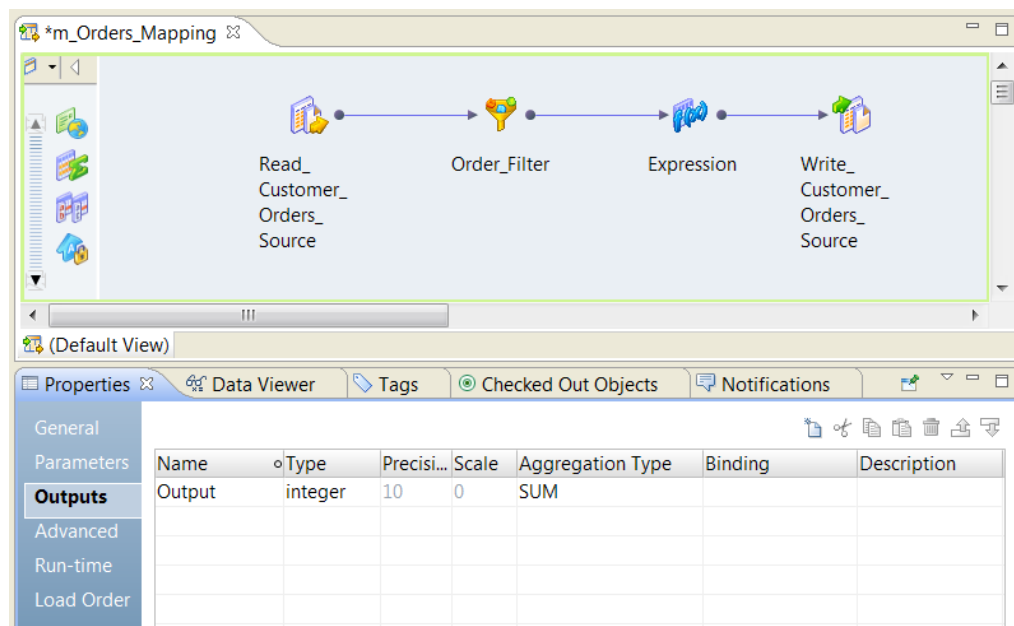
マッピング出力の定義

マッピングを作成して、マッピングの【プロパティ】でマッピング出力を定義します。各マッピング出力の定義では、実行する集計のタイプ、および結果のデータ型を記述します。

1. マッピングを作成したら、エディタをクリックして、マッピングの【プロパティ】にアクセスします。
2. 【出力】ビューをクリックします。
3. 【新規】をクリックして、マッピング出力を作成します。

Developer tool がマッピング出力を作成して、デフォルトのフィールドの値を設定します。

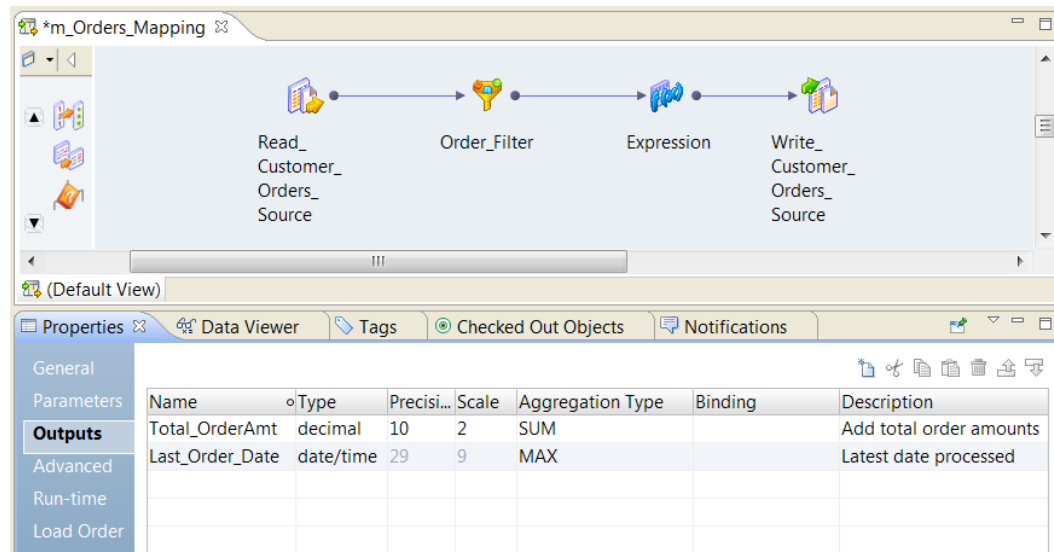
次の図は、【出力】ビューのマッピング出力のデフォルト値を示しています。



4. マッピング出力を識別する名前を変更します。
5. 数値または日付のマッピング出力タイプを選択します。精度およびスケールを入力します。
6. マッピング出力の集計タイプを選択します。
出力式を集計するか、マッピングが処理した最小式値または最大式値を見つけることができます。デフォルトは SUM です。
7. 【ファイル】 > 【保存】 をクリックして、マッピング出力を保存します。

式トランスフォーメーションにマッピング出力式を作成する前に、マッピング出力を保存する必要があります。

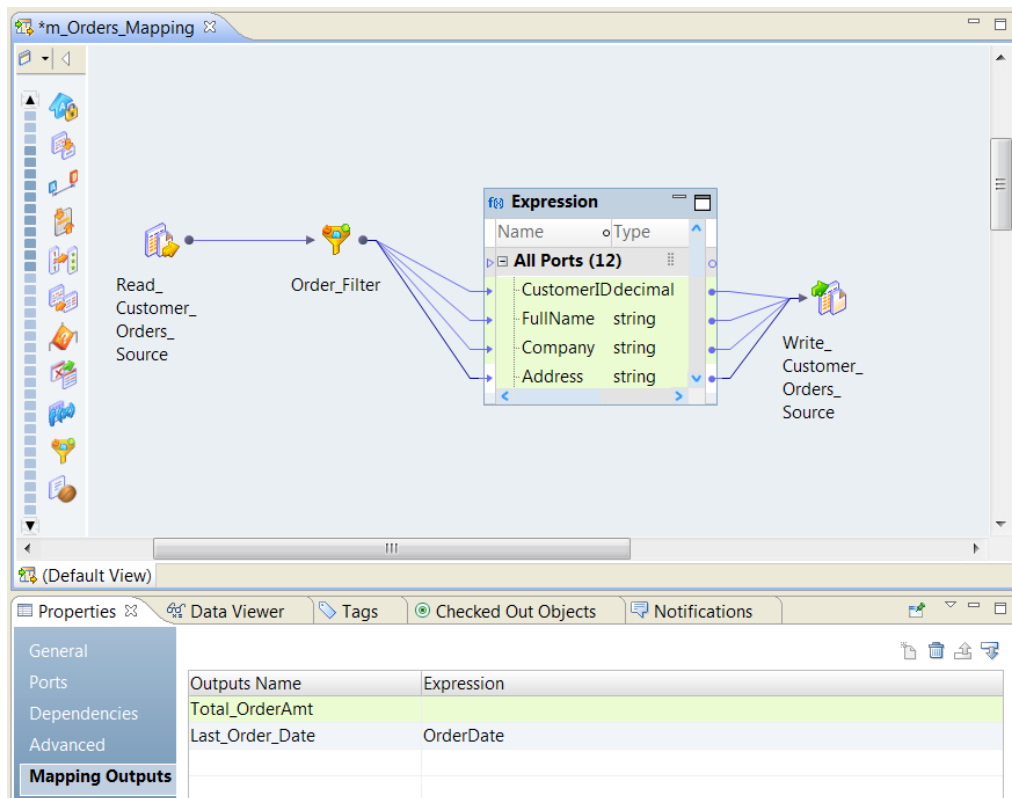
次の図は、decimal フィールドの合計、および最大日付値を含むマッピング出力が含まれるマッピング出力を示しています。



マッピング出力式の設定

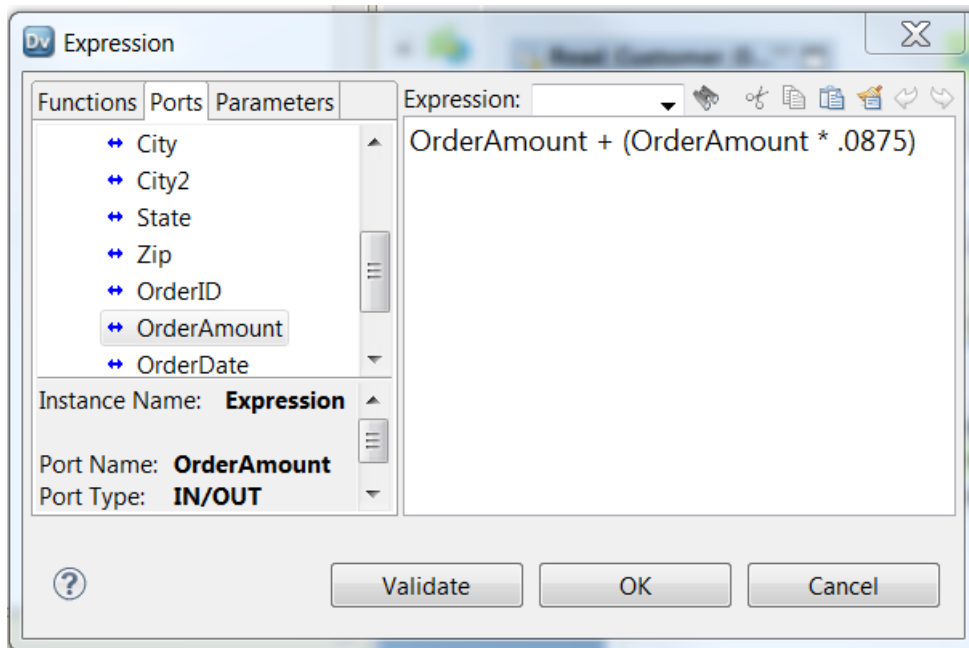
式トランスフォーメーションで、マッピングが処理する各行を集計する式を設定します。

1. 式トランスフォーメーションをマッピングに追加します。
トランスフォーメーションを配置する場所を決定する前に、マッピングロジックを検討します。マッピング出力には、式トランスフォーメーションが受け取る行の集計が含まれています。
2. 式トランスフォーメーションで、**【マッピング出力】** ビューをクリックします。
3. **【新規】** をクリックして、マッピング出力式を追加します。
Developer tool が、マッピングレベルで作成したいいずれかのマッピング出力と一致する出力名でマッピング出力を作成します。マッピングの**【プロパティ】** に複数のマッピング出力がある場合は、使用する適切なマッピング出力名を選択します。
次の図は、式トランスフォーメーションの**【マッピング出力】** ビューを示しています。



4. **【式】** カラムをクリックして、式エディタで式を入力します。

式には、ポート名のみを含めるか、関数、ポート、およびパラメータを含めることができます。次の図は、式エディタで Total_OrderAmt を計算する式を示しています。



5. **【検証】** をクリックして式が有効であることを確認します。

6. **【OK】** をクリックして式を保存します。

式がマッピング出力の【式】 カラムに表示されます。

7. 【ファイル】 > 【保存】 をクリックして、式トランスフォーメーションを保存します。

マッピング出力の保持

マッピングをワークフローに追加したら、マッピングタスクのマッピング出力を保持できます。保持されたマッピング出力を、マッピングタスクが次回実行されるときの入力として使用できます。

1. マッピングをワークフローに追加して、マッピングタスクを作成します。
2. ワークフローの [マッピングタスク] アイコンをクリックして、マッピングタスクの【プロパティ】を表示します。
3. 【持続】 ビューをクリックします。

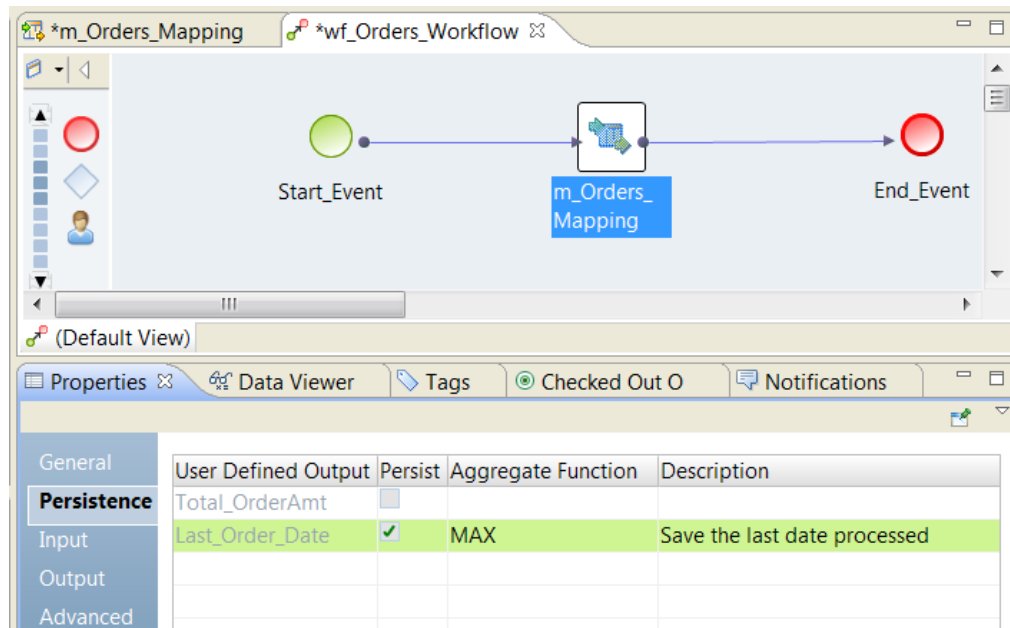
ユーザー定義のマッピング出力のリストが【持続】 ビューに表示されます。

The screenshot shows a workflow editor with a task named 'm_Orders_Mapping' between 'Start_Event' and 'End_Event'. Below the workflow, the 'Properties' panel is open, showing the 'Persistence' tab. The table below represents the data shown in this tab.

General	User Defined Output	Persist	Aggregate Function	Description
Persistence	Total_OrderAmt	<input type="checkbox"/>		
Input	Last_Order_Date	<input type="checkbox"/>		
Output				
Advanced				

4. 【持続】 を有効にして、マッピングタスクが実行された後にマッピング出力を保存します。

5. 必要に応じて、集計タイプを変更して説明を入力します。
次の図は、マッピングタスクの「持続」ビューを示しています。



Last_Order_Date マッピング出力が保持されています。集計関数は MAX であるため、データ統合サービスは最大注文日の値をリポジトリに保存します。

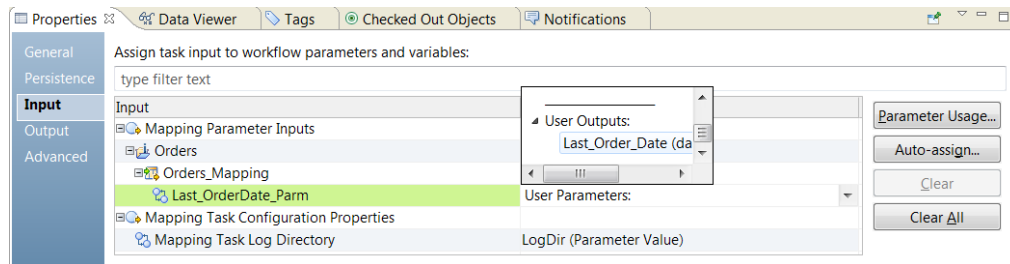
マッピングタスクの入力への保持された出力の割り当て

マッピングタスクの保持されたマッピング出力は、ワークフローが次回実行されるときに同じマッピングタスクの入力パラメータにバインドできます。

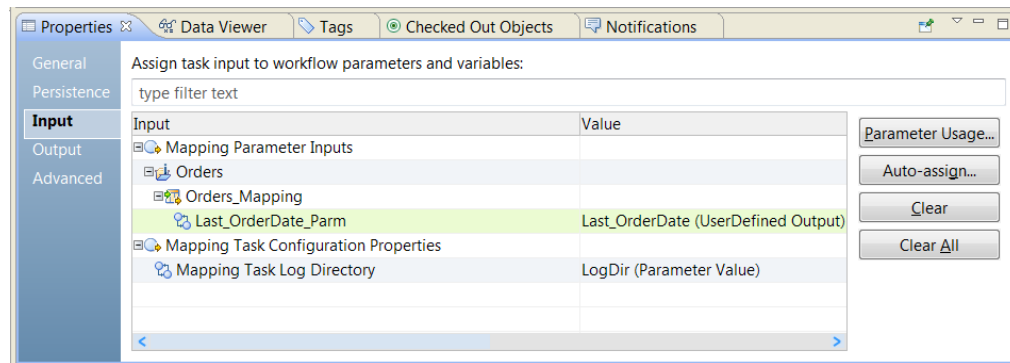
マッピングタスクの保持された最後の注文日の値を、同じマッピングタスクの入力パラメータとして割り当てます。Last_OrderDate_Parm パラメータを使用して処理する注文を選択するように、フィルタトランスフォーメーションを設定します。入力行を選択するフィルタ式は、Order_Date > Last_OrderDate_Parm です。

1. ワークフローの「マッピングタスク」アイコンをクリックして、マッピングタスクの「プロパティ」ビューを表示します。
マッピングタスクの入力パラメータのリスト、およびパラメータ化されたマッピングタスク設定プロパティのリストが表示されます。マッピングには、マッピング出力の割り当て先のマッピングパラメータが含まれる必要があります。
2. マッピング出力のバインド先となるマッピング入力パラメータを見つけます。「値」列をダブルクリックして、選択矢印を表示します。
3. 選択矢印をクリックして、入力パラメータに割り当てることができるパラメータおよび変数のリストを表示します。
4. リストの「ユーザー出力」セクションをスクロールして、使用する保持されたマッピング出力を選択します。

次の図は、マッピングタスクの「入力」ビューの Last_OrderDate_Parm マッピングパラメータを示しています。



5. マッピング出力を選択して、パラメータに割り当てます。
入力パラメータの値カラムにマッピング出力名が表示されます。



6. **【ファイル】 > 【保存】** をクリックして、マッピングタスクを保存します。
Last_OrderDate_Parm が、リポジトリの保持された注文日の値にバインドされます。

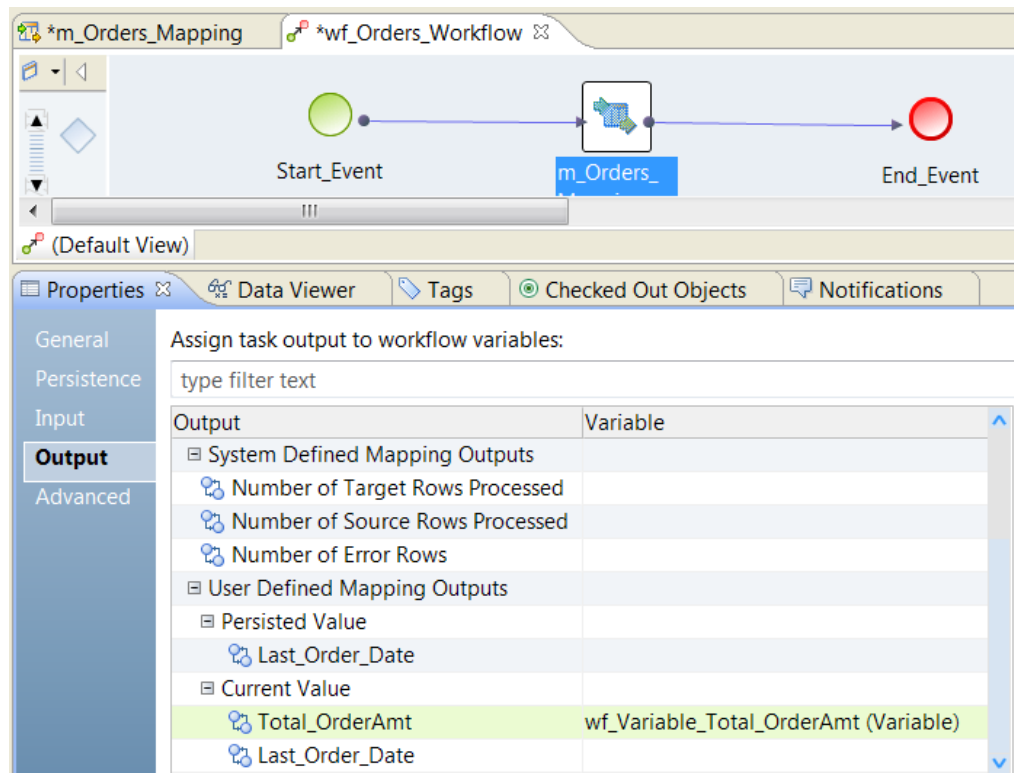
マッピング出力のワークフロー変数へのバインド

マッピング出力をワークフロー変数にバインドして、値をワークフローの他のタスクに渡すことができます。

マッピング出力値を別のタスクに渡すには、マッピングタスクの**【出力】**ビューでマッピング出力をワークフロー変数にバインドします。現在のマッピングタスクのマッピング出力、または以前のマッピングタスク実行の保持されたマッピング出力をバインドできます。

1. マッピング出力を含むマッピングをワークフローに追加します。
2. ワークフローの**【マッピングタスク】**アイコンをクリックして、マッピングタスクの**【プロパティ】**を表示します。
3. マッピングタスクの**【プロパティ】**で、**【出力】**ビューをクリックします。
マッピングタスクの**【出力】**ビューに、タスクからワークフロー変数に渡すことができるデータが表示されます。
4. 変数にバインドするマッピング出力を見つけます。
5. **【変数】**カラムをダブルクリックして選択矢印にアクセスし、ワークフロー変数のリストを表示します。

次の図は、マッピングタスクの**【出力】**ビューで、Total_OrderAmt マッピング出力を wf_Variable_Total_OrderAmt ワークフロー変数にバインドする場所を示しています。



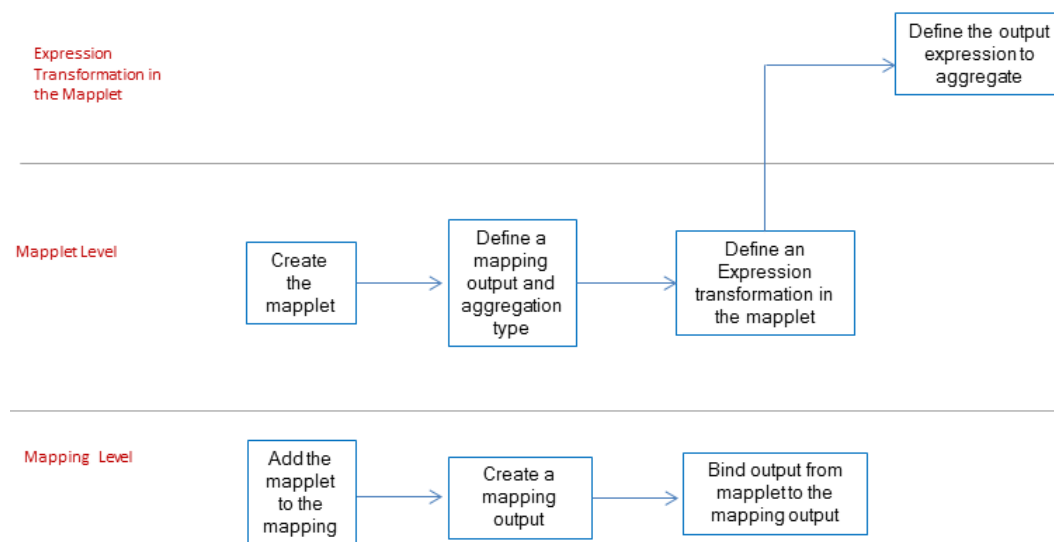
6. ワークフロー変数を作成するには、**【値】** カラムのワークフロー変数のリストから **【新規変数】** オプションをクリックします。
- 変数名、タイプ、およびデフォルト値を入力します。

マップレット出力をマッピング出力にバインドする方法

マッピング出力を返すようにマップレットを設定できます。マップレットのマッピング出力を、マッピング出力にマッピングレベルでバインドできます。

マップレットをマッピングに含めると、マップレットは出力の値を計算して、出力値をマッピングに渡します。マップレットの複数の出力を、同一の出力にマッピングレベルでバインドできます。マップレットのシステム定義の出力をマッピング出力にバインドすることもできます。

次の図は、マップレット出力を設定してそれらをマッピング出力にバインドするプロセスを示しています。



マップレットの出力をマッピング出力にバインドするには、以下の手順を実行します。

1. マップレットを作成する。
2. マップレットの【出力】ビューで、マップレット出力名および集計のタイプを定義します。
3. マップレットに式トランスフォーメーションを追加して、式の【マッピング出力】ビューでマッピング出力式を設定します。
4. マップレットをマッピングに追加します。
5. マッピングにマッピング出力を作成します。
6. マップレットの出力をマッピング出力にバインドします。

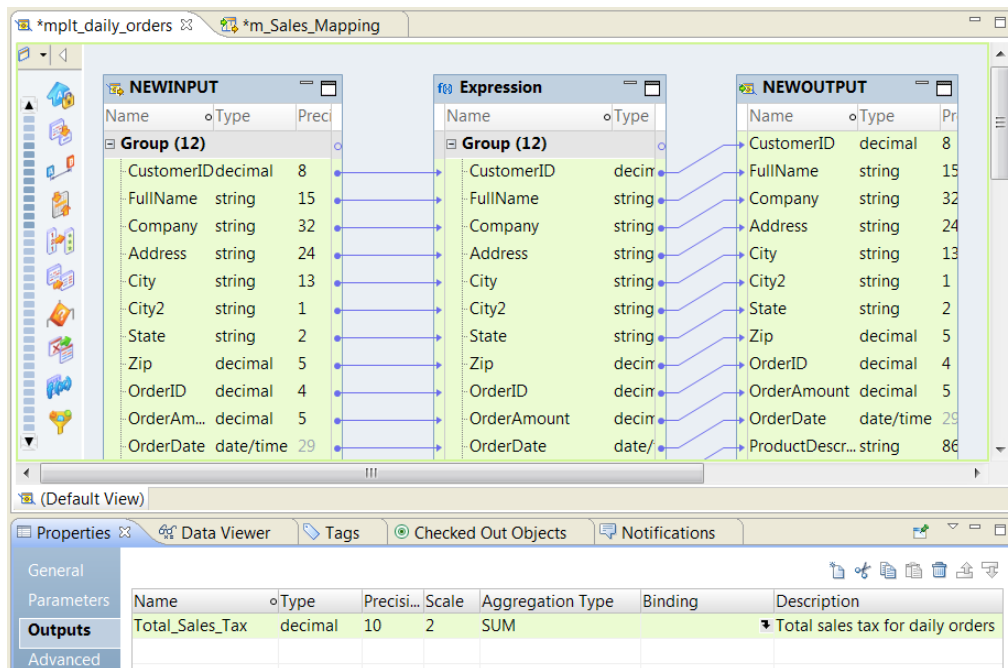
マップレット出力の定義

マップレットを作成して、マップレットの【プロパティ】ビューの【出力】タブでマッピング出力を定義します。各マッピング出力の定義では、実行する集計のタイプ、および結果のデータ型を記述します。

1. マップレットを作成したら、マッピングキャンバスの内側をクリックして、マップレットのプロパティにアクセスします。
2. 【出力】ビューをクリックします。
3. 【新規】をクリックして、マッピング出力を作成します。
Developer tool がマッピング出力を作成して、デフォルトのフィールドの値を設定します。
4. マッピング出力を識別する名前を変更します。
5. 数値または日付のマッピング出力タイプを選択します。数値タイプを作成する場合は、精度とスケールを入力します。
6. マッピング出力の集計タイプを選択します。

出力式を集計するか、マッピングが処理した最小式値または最大式値を見つけることができます。デフォルトは SUM です。

次の図は、集計タイプ SUM の Total_Sales_Tax というマップレット出力を示しています。



7. **【ファイル】 > 【保存】** をクリックして、マッピング出力を保存します。

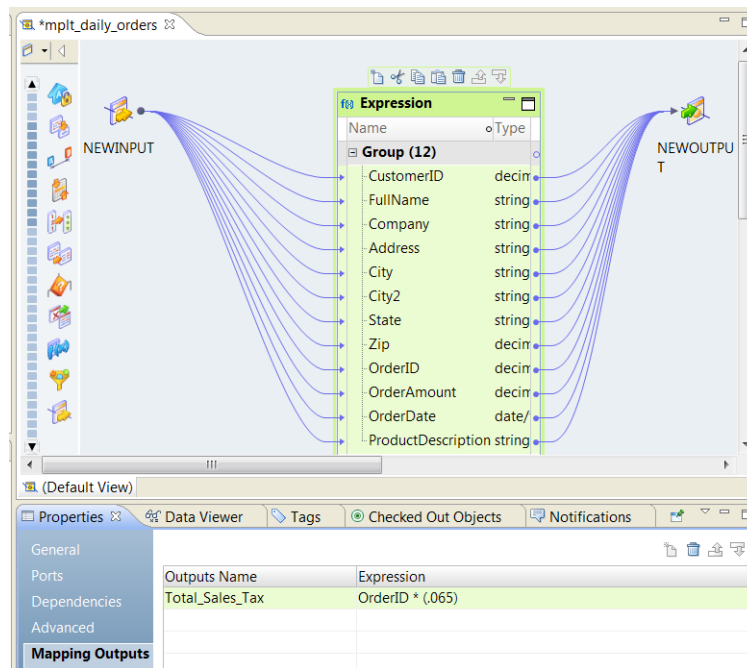
式トランスフォーメーションにマッピング出力式を作成する前に、マッピング出力を保存する必要があります。

マップレットでのマッピング出力式の設定

マップレットが処理する各行を集計する式を設定します。

1. 式トランスフォーメーションをマップレットに追加します。
トランスフォーメーションを配置する場所を決定する前に、マップレットロジックを検討します。
2. 式トランスフォーメーションで、**【マッピング出力】** ビューをクリックします。
3. **【新規】** をクリックして、出力式を追加します。
Developer tool が、マップレットレベルで作成したいいずれかのマッピング出力と一致する出力名でマッピング出力を作成します。複数の出力を用意して、そこから選択するようにできます。
4. 式エディタで式を入力します。
式には、ポート名を含めるか、関数、ポート、およびパラメータを含めることができます。
5. **【検証】** をクリックして式が有効であることを確認します。
6. **【OK】** をクリックして式を保存します。

次の図は、消費税を計算するマッピング出力式が表示された **【マッピング出力】** ビューを示しています。



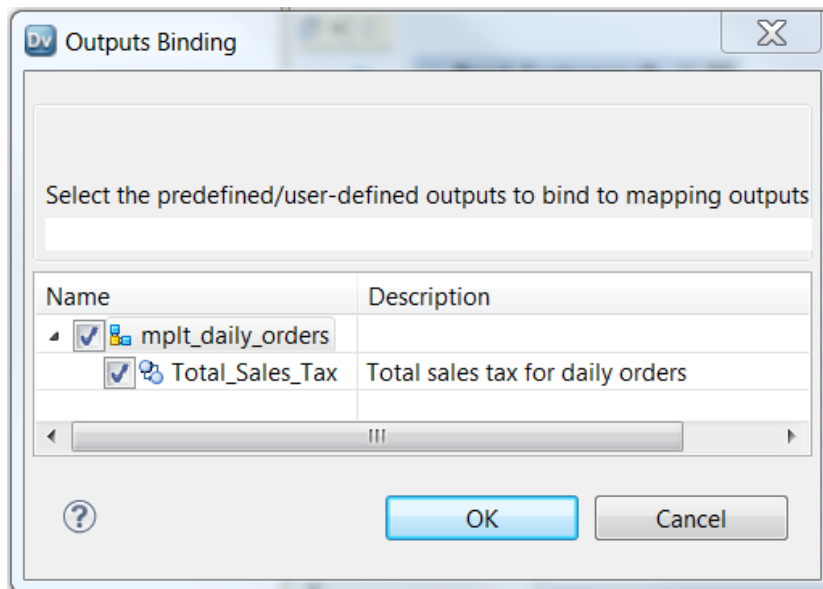
7. **【ファイル】** > **【保存】** をクリックして、式トランスフォーメーションを保存します。

マップレットからの出力のマッピング出力へのバインド

マップレットをマッピングに含めた場合、マッピングレベルで定義したマッピング出力にマップレットの出力をバインドできます。

1. マッピングを定義して、マップレットをマッピングに追加します。
2. マッピングキャンバスをクリックして、マッピングの**【プロパティ】**を表示します。
3. **【出力】** ビューをクリックします。
4. **【新規】** をクリックして、マッピング出力を作成します。
Developer tool がマッピング出力を作成して、デフォルトのフィールドの値を設定します。
5. バインド先となるマップレットのフィールドと一致するように、マッピング出力タイプ、集計タイプ、精度、およびスケールを変更します。
6. 必要に応じて、名前を変更して説明を入力します。
7. **【バインディング】** フィールドの選択矢印をクリックして、出力のリストを表示します。

次の図は、**【出力バインディング】** ダイアログボックスを示しています。



8. マッピング出力にバインドするマププレット出力を選択します。
複数のマププレット出力を選択して、同一のマッピング出力にバインドできます。
9. [OK] をクリックします。
選択したマププレット出力が **【バインディング】** フィールドに表示されます。
次の図は、マッピング出力の **【バインディング】** フィールドのマププレット出力名を示しています。

Name	Type	Precisi...	Scale	Aggregation Type	Binding	Description
Total_Tax	decimal	10	2	SUM	mplt_daily_orders.Total_Sales_Tax	Sales tax

第 5 章

動的マッピング

この章では、以下の項目について説明します。

- [動的マッピングの概要, 107 ページ](#)
- [動的マッピングの設定, 108 ページ](#)
- [動的ソース, 111 ページ](#)
- [動的ターゲット, 115 ページ](#)
- [動的ポートおよび生成されたポート, 120 ページ](#)
- [動的式, 122 ページ](#)
- [入力ルール, 123 ページ](#)
- [選択ルールとポートセレクト, 132 ページ](#)
- [設計時リンク, 134 ページ](#)
- [ランタイムリンク, 136 ページ](#)
- [動的マッピングのトラブルシューティング, 139 ページ](#)

動的マッピングの概要

動的マッピングとは、実行時にソース、ターゲット、およびトランスフォーメーションロジックに対する変更に対応できるマッピングです。動的マッピングは、スキーマまたはメタデータへの頻繁な変更を管理する、あるいは異なるスキーマを持つデータソースにマッピングロジックを再利用する場合に使用します。動的マッピングを作成するには、ルール、パラメータ、全般的なトランスフォーメーションパラメータを設定します。

ソース、ターゲット、またはルックアップでデータソースが変更される場合、マッピングを設定すれば実行時にメタデータの変更を動的に取得できます。マッピングのすべてのステージで変更を受信してプロパゲートするには、マッピング内でパラメータ、ルール、ポート、リンクを設定します。マッピングを再び実行する前に手動でデータオブジェクトを同期して各トランスフォーメーションを更新する必要はありません。データ統合サービスは、トランスフォーメーションポート、ポート内のトランスフォーメーションロジック、マッピング内のポートリンクを動的に決定できます。

動的マッピングの例

毎週、さまざまな部門から顧客データを受信し、結合して集計する必要があるとします。部門は、部門分析用の追加のカラムを含めるためにソーススキーマを定期的に変更する可能性があります。

データソースへのこのような変更に対応するために、動的マッピングを作成します。読み取り時にデータオブジェクトカラムを取得するように、読み取りトランスフォーメーションを設定します。必要なカラムを含めて、その他のすべてのカラムを除外するように入力ルールを作成します。

動的マッピングの設定

ソースが変更された場合、変更に対応するために読み取りトランスフォーメーションを設定できます。例えば、異なるデータソースを使用するか、データソースに基づいてデータオブジェクトを更新するためにトランスフォーメーションを設定できます。ターゲットが変更された場合、ターゲットの変更に対応するために書き込みトランスフォーメーションを設定できます。例えば、関連付けられているデータオブジェクトまたはマッピングフローに基づいてカラムを生成するために書き込みトランスフォーメーションを設定できます。ターゲットがリレーショナルの場合、実行時にテーブルを作成または置換できます。

マッピング全体を通して変更を受信してプロパゲートするために、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を設定します。データフローに基づいて新しいカラムまたは変更されたカラムを受信するために動的ポートを作成します。動的カラムは各入力カラムのポートを生成します。動的ポートが受信するカラムを決定し、生成されたポートの名前または順序を変更するように入力ルールを設定します。

式で動的ポートまたは選択ルールを使用して動的式を作成します。動的ポートを含めた場合、式は動的ポートが生成する各ポートに対して実行されます。選択ルールを含めた場合、式はルールの各ポートに対して実行されます。

式、ジョイナ、またはルックアップトランスフォーメーションに生成されたポートが含まれる場合、マッピングが実行されるときに生成されたポートに対する変更に対応するポート選択ルールを設定できます。例えば、販売データに基づいて計算を実行する必要があるが、販売カラム名が各ソースで異なっているとします。計算するには正しいカラムを選択するルールを作成します。

パラメータを使用して実行時に値を変更できます。パラメータを使用して、マッピング内のソース、ターゲット、接続、ルールなどの値を変更します。

トランスフォーメーションは、マッピングを設計するときに直接リンクを作成できないように変更される場合があります。設計時にリンクを作成できない場合、ランタイムリンクを設定できます。ランタイムリンクはポリシーまたはパラメータを使用して実行時にトランスフォーメーショングループ間をリンクするポートを決定します。

動的データソース

実行時にソースおよびターゲットに対する変更に対応するためにマッピングを設定できます。動的マッピングにはフラットファイルおよびリレーショナルデータソースを含めることができます。パラメータを使用して、予測される変更のタイプに基づいてトランスフォーメーションのプロパティを設定できます。

次のデータソースに対するランタイム変更に対応するためにマッピングを設定できます。

ソース

動的ソースにはリレーショナルソースおよびフラットファイルソースを含めることができます。ランタイム変更に対応するために読み取りトランスフォーメーションおよび物理データオブジェクトを設定します。ファイルまたはソース接続の場所、入力ソースカラムに対する変更、またはデータオブジェクトに基づいてソースメタデータを変更できます。

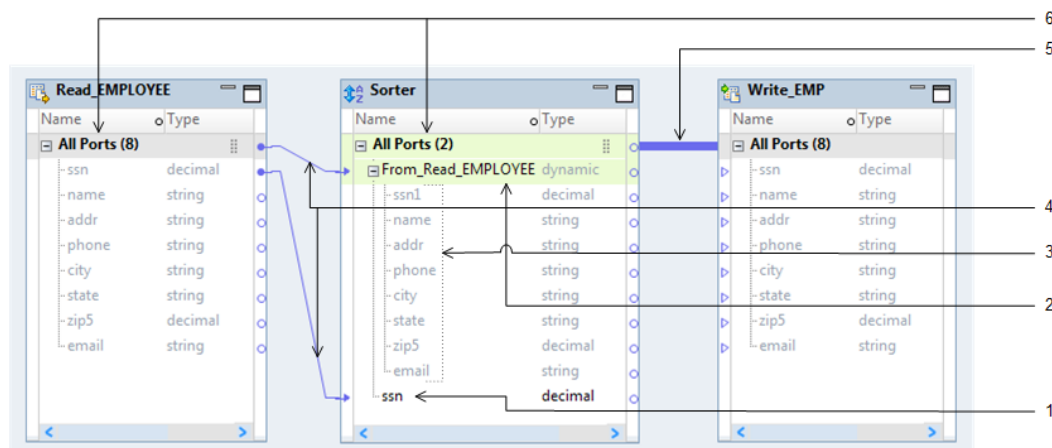
ターゲット

動的ターゲットにはリレーショナルターゲットおよびフラットファイルターゲットを含めることができます。マッピングフローまたは関連付けられているデータオブジェクトに基づいてターゲットを定義できます。データソースからデータオブジェクトのカラムを取得することも選択できます。パラメータを使用して、ターゲットデータオブジェクトやターゲット接続など、ランタイムプロパティを変更することができます。

動的マッピングのポートおよびリンク

メタデータの変更を処理するために、通常のマッピングには含まれないポートおよびリンクのタイプを設定できます。

次の画像は、動的マッピングで表示されるポートおよびリンクを示しています。



1. 静的ポート（ポート）
2. 動的ポート
3. 生成されたポート
4. 設計時リンク（リンク）
5. ランタイムリンク
6. ポートグループ

静的ポート（ポート）

マッピング、動的、非動的のどのタイプでも作成できるポート。データはポートの内外に渡すことができ、データには動的設定が一切含まれません。

動的ポート

アップストリームトランスフォーメーションから1つ以上のカラムを受信できるトランスフォーメーションのポート。動的ポートは、マッピングを使用して渡すメタデータに基づき、新規または変更されたカラムを受信できます。

生成されたポート

動的ポート内の1つのカラムを表すポート。動的ポートは動的ポートのルールに基づいて各カラムの生成されたポートを作成します。

設計時リンク（リンク）

データを1つのトランスフォーメーションから別のトランスフォーメーションにプロパゲートするポートを接続するために作成するリンク。これらのリンクは通常のマッピングで作成することもできます。

ランタイムリンク

ポリシー、パラメータ、またはその両方に基づいてデータ統合サービスが実行時に接続するポートを決定するために使用するトランスフォーメーショングループ間のリンク。

ポートグループ

データの行を表すマッピング内のポートのセット。動的マッピングでは、グループをダウンストリームトランスフォーメーションにドラッグして動的ポートを作成できます。

動的マッピングのルール

動的トランスフォーメーション内でルールを作成し、動的ポートが受信するポートと生成するポートを制御します。

次のタイプの動的マッピングのルールを設定できます。

入力ルール

入力ルールは動的ポートが生成するポートを定義します。ポートを含めることまたは除外することを選択できます。生成されたポートの名前や順序を変更することも選択できます。

選択ルールとポートセレクト

選択ルールを作成して、データ統合サービスが実行時に処理する生成されたポートを定義します。ポートセレクト内で選択ルールを作成します。ポートセレクトには、式または結合条件またはルックアップ条件で参照できるポートが含まれます。実行時に予測されるメタデータの変更に基いて、1つのトランスフォーメーションで複数のポートセレクトを設定できます。

動的マッピングのパラメータ

パラメータは、マッピングの実行間で変わる定数値です。フラットファイルまたはリレーショナルリソースのソースやターゲットを変更するには、動的マッピングのパラメータを使用します。パラメータを使用して、入力ルール、選択ルール、トランスフォーメーションプロパティ、ランタイムリンクを変更することもできます。

次の表に、動的マッピングコンポーネントについて作成できるパラメータの機能を示します。

動的マッピングコンポーネント	パラメータの機能
アグリゲータトランスフォーメーション	グループ化ポートを変更する。
ジョイナトランスフォーメーション	結合条件を変更する。
ルックアップトランスフォーメーション	ルックアップ条件を変更する。
ランクトランスフォーメーション	グループ化ポートを変更する。
読み取りトランスフォーメーション	次のタスクを実行するためにパラメータを作成します。 <ul style="list-style-type: none">- フラットファイルソースの入力ファイル名またはディレクトリを変更する。- リレーショナルソースの接続を変更する。- フラットファイルデータオブジェクト、カスタマイズされた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変更する。
ルール	次のタスクを実行するためにパラメータを作成します。 <ul style="list-style-type: none">- 名前またはパターン別に入力ルール条件を変更する。- 名前またはパターン別に選択ルール条件を変更する。
ランタイムリンク	トランスフォーメーショングループ間をリンクするためのポートのセットを変更する。

動的マッピングコンポーネント	パラメータの機能
ソートトランスフォーメーション	ソートキーを変更する。
書き込みトランスフォーメーション	次のタスクを実行するためにパラメータを作成します。 <ul style="list-style-type: none"> - フラットファイルターゲットの出力ファイル名またはディレクトリを変更する。 - リレーショナルターゲットの接続を変更する。 - フラットファイルデータオブジェクト、カスタマイズされた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変更する。

関連項目：

- [「マッピングパラメータの概要」 \(ページ 45\)](#)

動的ソース

動的ソースとは実行時に変更できるソースです。マッピングでフラットファイル動的ソースまたはリレーショナル動的ソースを設定できます。

ソースの動的ランタイム機能は、次の方法で設定できます。

データソースからカラムを取得する。

実行時にソースがわずかに変更されることが予想される場合、実行時にフラットファイルまたはリレーショナルオブジェクトカラムを取得するように読み取りトランスフォーメーションを設定できます。リレーショナルデータソースまたはフラットファイルデータソースの構造に基づいて、実行時に読み取りトランスフォーメーションのポートを更新できます。

パラメータを割り当ててソースのフラットファイル名およびディレクトリを決定する。

フラットファイルソースが類似している場合、パラメータをファイル名またはディレクトリに割り当てることができます。パラメータを使用する場合、各ソースのデータオブジェクトを作成する必要はありません。

パラメータを割り当ててリレーショナルデータオブジェクトのリソース、テーブル所有者、またはディレクトリを決定する。

リレーショナルソースが類似している場合、パラメータを割り当ててリソース、接続、テーブル所有者プロパティを取得できます。

パラメータを割り当ててファイルまたはリレーショナルソースに使用するデータオブジェクトを決定する。

ソースがわずかに変更されることが予想される場合、リレーショナルデータソースまたはフラットファイルデータソースの構造に基づいて、実行時に読み取りトランスフォーメーションのポートを更新できます。

次の表に、ソースの動的ランタイム機能を設定できるケースを示します。

動的ランタイムソース機能	設定
データソースからカラムを取得する。	次のソースタイプの読み取りトランスフォーメーションで【 データオブジェクト 】タブを設定する。 - フラットファイル - リレーショナル
パラメータを割り当ててフラットファイル名およびディレクトリを決定する。	次のソースタイプの物理データオブジェクトで【 詳細 】タブを設定する。 - フラットファイル
パラメータを割り当てて接続、所有者、またはリソースを決定する。	次のソースタイプの読み取りトランスフォーメーションで【 ランタイム 】タブを設定する。 - リレーショナル
パラメータを割り当ててデータオブジェクトを決定する。	次のソースタイプの読み取りトランスフォーメーションで【 データオブジェクト 】タブを設定する。 - フラットファイル - リレーショナル

データソースからカラムを取得する

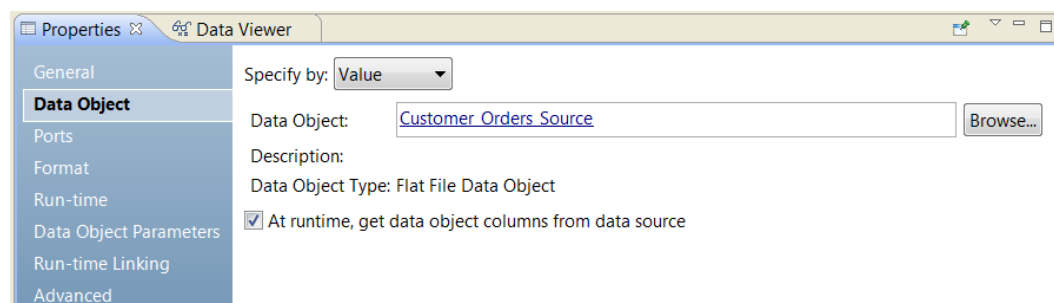
リレーショナルまたはフラットファイルデータソースの構造に基づいて、実行時に読み取りトランスフォーメーションのポートを更新できます。この方法を使用して、読み取りトランスフォーメーションのランタイムインスタンスを更新することもできます。

ソースがわずかに変更されることが予想される場合、実行時にカラムを更新します。例えば、別の組織からソースを処理する必要があるが、組織ではソースファイル内のカラムの順序については保証できないとします。実行時にデータオブジェクトカラムを更新するオプションを有効にした場合、データ統合サービスはソースデータの構造に基づいて読み取りトランスフォーメーションのポートを変更します。読み取りトランスフォーメーションは、処理を行う動的マッピングのダウンストリームトランスフォーメーションにデータを渡します。

実行時にデータオブジェクトカラムを更新した場合、データ統合サービスは読み取りトランスフォーメーションのランタイムインスタンスを更新します。モデルリポジトリのメタデータは更新しないため、Developer tool で変更を確認することはできません。モデルリポジトリの物理データオブジェクト定義を更新するには、Developer tool で同期オプションを使用します。Developer tool は物理データオブジェクトのメタデータを再度インポートし、メタデータを変更します。

注: 読み取りまたはルックアップトランスフォーメーションのカスタム SQL クエリを作成した場合、カスタムクエリのメタデータがデータソースカラムの更新されたメタデータと競合する場合があります。クエリが、デフォルトでデータ統合サービスが生成するクエリと一致しない場合、マッピングは失敗します。

次の画像は、【**データオブジェクト**】タブでオプションを有効にする箇所を示しています。



データ統合サービスはスキーマでリレーショナルソースの構造を決定します。【ランタイム】タブに表示されるリソースのスキーマを調査します。次に、スキーマに基づいてトランスフォーメーションデータオブジェクトのカラムを更新します。

データ統合サービスは、フラットファイルの物理データオブジェクトを設定する方法に基づいてフラットファイルソースの構造を決定します。実行時にカラム名を生成するために、データオブジェクトを設定します。

この機能は、フラットファイルまたはリレーショナルソースの読み取りトランスフォーメーションの【データオブジェクト】タブで設定します。

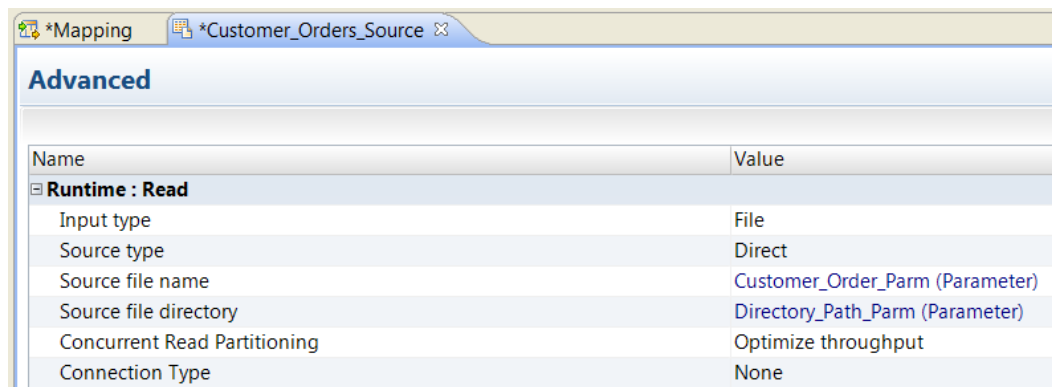
フラットファイル物理データオブジェクトのプロパティの詳細については、『*Informatica Developer Tool ガイド*』を参照してください。

フラットファイル名にパラメータを割り当てる

類似したフラットファイルソースを使用して動的マッピングを実行するために、ファイル名またはディレクトリにパラメータを割り当てることができます。パラメータを使用する場合、各ソースのデータオブジェクトを作成する必要はありません。

フラットファイル物理データオブジェクトのファイル名とディレクトリをパラメータ化できます。データオブジェクトからトランスフォーメーションを作成する前にプロパティをパラメータ化できます。物理データオブジェクトプロパティの【詳細】タブでパラメータを設定します。物理データオブジェクトからトランスフォーメーションを作成するときに、マッピングパラメータを使用してパラメータのデフォルト値をオーバーライドできます。

次の画像は、物理データオブジェクトの【詳細】タブを示しています。



この機能は、フラットファイルソースの物理データオブジェクトの【詳細】タブで設定します。

リレーショナルソースのプロパティにパラメータを割り当てる

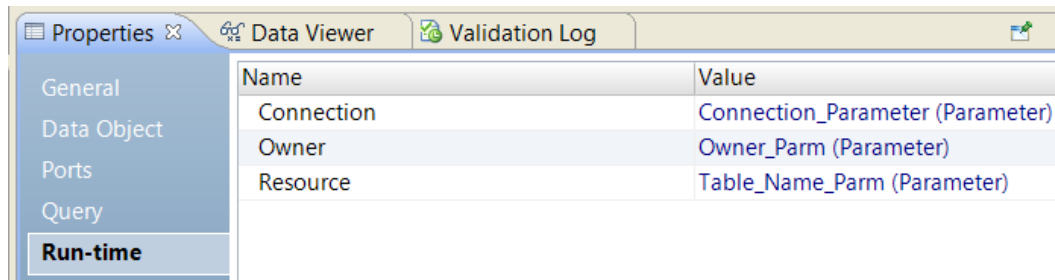
類似したリレーショナルソースを使用して動的マッピングを実行するために、読み取りトランスフォーメーションでリソース、接続、テーブル所有者プロパティにパラメータを割り当てることができます。

同じデータベースで異なるが類似しているテーブルを使用してマッピングを実行するには、リソースのパラメータを使用します。リソースのパラメータを使用する場合、各ソースのデータオブジェクトを作成する必要はありません。接続用のパラメータを使用して異なるデータベースにアクセスします。複数のリレーショナルソースに対して一意の SQL クエリを実行する必要がある場合があります。

トランスフォーメーションプロパティの【ランタイム】タブでリレーショナルテーブルのパラメータを設定します。リレーショナル物理データオブジェクトでこれらのプロパティはパラメータ化できません。読み取りトランスフォーメーションのプロパティのパラメータを作成する場合、マッピングパラメータを作成します。

デフォルトでは、接続の接続タイプパラメータを作成します。テーブル所有者のテーブル名と文字列パラメータにリソースタイプパラメータを設定します。

次の画像は、読み取りトランスフォーメーションの【ランタイム】タブを示しています。



この機能は、リレーショナルソースの読み取りトランスフォーメーションの【ランタイム】タブで設定します。

ソースデータオブジェクトにパラメータを割り当てる

データオブジェクトにパラメータを割り当てて、実行時に読み取りトランスフォーメーションのソースを変更することができます。

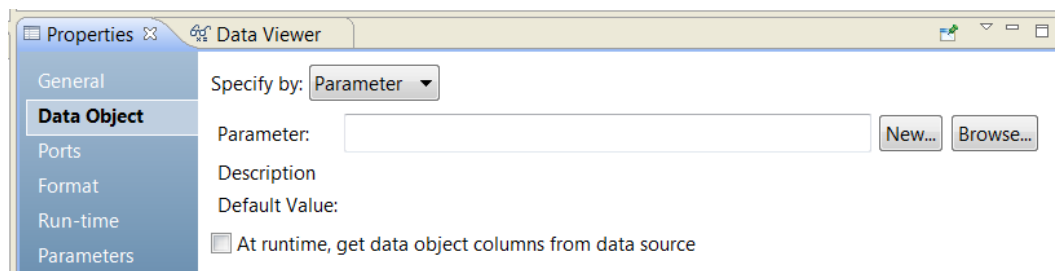
各データソースのモデルリポジトリに他の物理データオブジェクトがある場合、データオブジェクトをパラメータ化します。フラットファイルまたはデータベーステーブルの同じトランスフォーメーションを設定する必要がある場合にデータオブジェクトをパラメータ化することができます。データオブジェクトをパラメータ化すると、トランスフォーメーションで異なるプロパティまたは一意の SQL クエリがあるデータオブジェクトを使用できます。

物理データオブジェクトから読み取りトランスフォーメーションを作成すると、データオブジェクトの情報はトランスフォーメーションプロパティの【データオブジェクト】タブに表示されます。データオブジェクト名をクリックすると、モデルリポジトリにある物理データオブジェクトの定義を表示することができます。

データオブジェクトをパラメータ化するには、リソースタイプパラメータを作成するか、リソースパラメータを参照します。パラメータのデフォルト値は、モデルリポジトリ内の物理データオブジェクトの名前です。デフォルトのパラメータ値を作成する場合、リポジトリ内のデータオブジェクトのリストから物理データオブジェクト名を選択します。

データオブジェクトを変更すると、トランスフォーメーションポートが変更されます。トランスフォーメーションプロパティの【ポート】タブで、ポートを表示できます。

次の図は、パラメータでデータオブジェクトを指定した場合の【データオブジェクト】タブを示しています。



次の表に、[データオブジェクト] タブのパラメータオプションを示します。

パラメータオプション	説明
パラメータ	データオブジェクトとして設定したリソースパラメータの名前。読み取り専用。
説明	パラメータの説明。読み取り専用。
新規	リソースパラメータを作成します。パラメータのデフォルト値としてモデルリポジトリ内のデータオブジェクトを参照し、選択します。
参照	リソースパラメータを参照し、パラメータを選択します。
デフォルト値	データオブジェクトに設定したリソースパラメータのデフォルト値。デフォルト値は物理データオブジェクト名です。読み取り専用。

この機能は、フラットファイルまたはリレーショナルソースの読み取りトランスフォーメーションの [データオブジェクト] タブで設定します。

動的ターゲット

動的ターゲットでは、実行時にターゲットが変わります。フラットファイルおよびリレーショナルターゲットを動的に設定できます。

マッピングを実行すると、動的ターゲットはリレーショナルテーブル、フラットファイル、カスタマイズされたデータオブジェクトを含め、物理データターゲットからメタデータの変更を取得できます。また、アップストリームカラム定義に基づいてカラムを生成することもできます。

ターゲットの動的ランタイム機能は、次の方法で設定できます。

データソースからカラムを取得する。

ターゲットがわずかに変更されることが予想される場合、書き込みトランスフォーメーションを設定して、実行時にリレーショナルオブジェクトカラムを取得できます。書き込みトランスフォーメーションを設定して、ターゲットからメタデータを取得する場合、動的に更新してターゲットオブジェクトとの同期を維持するように書き込みトランスフォーメーションを設定できます。

マッピングフローに基づいてターゲットカラムを定義する。

マッピングフローに基づいてカラムを定義する場合、ターゲットカラムはアップストリームトランスフォーメーションによって決定されます。

データオブジェクトに基づいてターゲットカラムを定義する。

データオブジェクトに基づいてカラムを定義する場合、ターゲットカラムは関連付けられているデータオブジェクトによって決定されます。

実行時にリレーショナルターゲットテーブルを作成または置換する。

デフォルトでは、書き込みトランスフォーメーションを設定して実行時にターゲットを作成または置換する場合、データ統合サービスはデータオブジェクトに基づいてターゲットを作成します。マッピングフローに基づいてターゲットを作成することも、DDL クエリを定義してクエリに基づいてターゲットを作成することもできます。

パラメータを割り当ててリレーショナルデータオブジェクトのリソース、テーブル所有者、またはディレクトリを決定する。

リレーショナルターゲットが類似している場合、パラメータを割り当ててリソース、接続、テーブル所有者プロパティを取得できます。

パラメータを割り当ててファイルまたはリレーショナルターゲットに使用するデータオブジェクトを決定する。

カスタマイズされたデータオブジェクトを書き込みトランスフォーメーションとして作成し、パラメータ値をトランスフォーメーションのターゲットとして指定できます。パラメータの値を変更した場合、そのパラメータを使用するすべてのオブジェクトでターゲットが変更されます。

次の表に、ターゲットの動的ランタイム機能を設定できるケースを示します。

動的ランタイムターゲット機能	設定
データソースからカラムを取得する。	次のターゲットタイプの書き込みトランスフォーメーションで 【データオブジェクト】 タブを設定する。 - リレーショナル
データオブジェクトまたはマッピングフローに基づいてターゲットカラムを定義する。	次のターゲットタイプの書き込みトランスフォーメーションで 【ポート】 タブを設定する。 - フラットファイル - リレーショナル
実行時にテーブルを作成または置換する。	次のターゲットタイプの物理データオブジェクトで【詳細】 タブを設定する。 - リレーショナル
DDL クエリを定義して実行時にターゲットテーブルを作成します。	次のターゲットタイプの物理データオブジェクトで【詳細】 タブを設定する。 - リレーショナル - Hive
パラメータを割り当てて接続、所有者、またはリソースを決定する。	次のターゲットタイプの書き込みトランスフォーメーションで 【ランタイム】 タブを設定する。 - リレーショナル
パラメータを割り当ててデータオブジェクトを決定する。	次のターゲットタイプの書き込みトランスフォーメーションで 【データオブジェクト】 タブを設定する。 - フラットファイル - リレーショナル

データソースからカラムを取得する

リレーショナルデータソースの構造に基づいて、実行時に書き込みトランスフォーメーションのポートを更新できます。

ターゲットカラムがわずかに変更されることが予想される場合、実行時にカラムを更新します。実行時にデータソースからデータオブジェクトカラムを取得した場合、データ統合サービスはターゲットの構造に基づいてデータオブジェクトのランタイムインスタンスを作成します。モデルリポジトリ内のメタデータは更新しません。

注: データソースからカラムを取得してマッピングフローに基づいてターゲットを定義するために書き込みトランスフォーメーションを設定した場合、マッピングは失敗します。

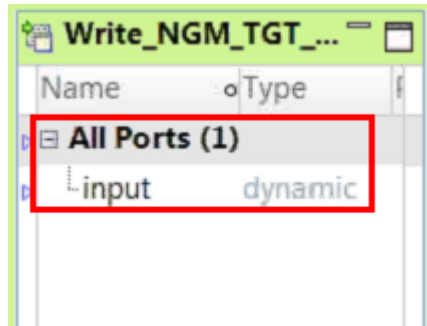
この機能は、リレーショナルターゲットの書き込みトランスフォーメーションの【データオブジェクト】タブで設定します。

マッピングフローに基づくターゲットの定義

マッピングフローによってカラムを定義する場合、ターゲットカラムはアップストリームトランスフォーメーションによって決定されます。アップストリームトランスフォーメーションによってポートの順序とメタデータが変更された場合、書き込みトランスフォーメーションはその変更を取得します。

アップストリームトランスフォーメーション内のキーカラムがターゲット内のキーカラム名に一致する場合、ターゲットを作成または置換するときにキーを維持できます。

次の画像は、マッピングフローに基づいてターゲットカラムを定義する場合に書き込みトランスフォーメーションがどのように表示されるかを示しています。



注: 予期しない結果にならないように、マッピングフローに基づいてターゲットを定義する書き込みトランスフォーメーションに対するランタイムリンクは設定しないでください。

この機能は、フラットファイルおよびリレーショナルターゲットの書き込みトランスフォーメーションの【ポート】タブで設定します。

データオブジェクトに基づくターゲットの定義

書き込みトランスフォーメーションを設定して、関連付けられているデータオブジェクトに基づいてターゲットカラムを定義できます。

データオブジェクトに基づいてターゲットカラムを定義する場合、書き込みトランスフォーメーションには動的ポートと生成されたポートが含まれます。

実行時にターゲットを作成または置換することもできます。カラム名が一致する場合、ターゲットを作成または置換するときにターゲットキーを維持できます。カラム名が一致するようにルールを設定できます。

この機能は、フラットファイルおよびリレーショナルターゲットの書き込みトランスフォーメーションの【ポート】タブで設定します。

実行時にターゲットを作成または置換する

実行時に、データ統合サービスはテーブルを作成したり、削除または置換したりすることができます。データ統合サービスは、マッピングフローまたは関連付けられているデータオブジェクトに基づいてテーブルを作成または置換します。

書き込みトランスフォーメーションを設定してターゲットを作成または置換する場合、データ統合サービスは書き込みオブジェクトに関連付けられている既存のターゲットテーブルをドロップし、データオブジェクトを使用するまたはマッピングフローを使用するための設定に基づいてテーブルを作成します。

データ統合サービスがデータオブジェクトに基づいてテーブルを作成した場合、テーブルにはデータオブジェクトのポートに一致するカラムが含まれます。カスタマイズされたデータオブジェクトを使用して実行時にターゲットを作成または置換する場合、データ統合サービスはデータオブジェクト接続で参照される名前を使用してテーブルを作成します。

データ統合サービスがマッピングフローに基づいてテーブルを作成する場合、テーブルには書き込みトランスフォーメーションの生成されたポートに一致するカラムが含まれます。

この機能は、データオブジェクトの【詳細】タブで設定します。

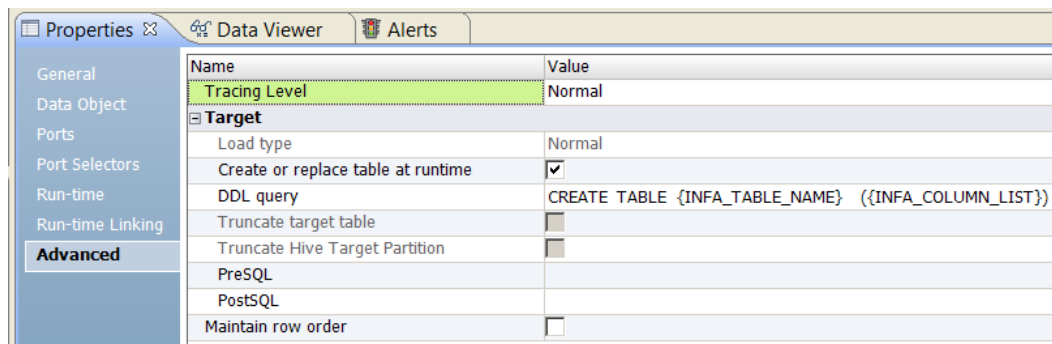
DDL クエリの定義

実行時に、データ統合サービスで定義した DDL クエリに基づいてリレーショナルターゲットテーブルおよび Hive ターゲットテーブルを作成または置換できます。

書き込みトランスフォーメーションを設定してターゲットを作成または置換すると、デフォルトでデータ統合サービスにより、書き込みオブジェクトに関連付けられた既存のターゲットテーブルがすべて削除されます。次に、マッピングフローまたはデータオブジェクトに基づいてテーブルが作成されます。

テーブルをカスタマイズするか、パーティションなどの追加パラメータを指定する場合、DDL クエリを定義することができます。データ統合サービスは、この DDL クエリに基づいてターゲットテーブルを作成または置換します。テーブルには、DDL クエリで定義するカラムが含まれます。

次の図に、【DDL クエリ】フィールドを示します。



DDL クエリにはプレースホルダを入力できます。データ統合サービスにより、実行時にプレースホルダが実際の値に置き換えられます。例えば、テーブルに 50 個のカラムが含まれている場合、DDL クエリにすべてのカラム名を入力する代わりに、プレースホルダを入力できます。

DDL クエリには、次のプレースホルダを入力できます。

INFA_TABLE_NAME

実行時にターゲットテーブル名を取得します。

INFA_COLUMN_LIST

実行時にターゲットテーブルのカラムのリストを取得します。

INFA_PORT_SELECTOR

ポートセレクタを追加します。

プレースホルダは 2 つの中括弧で囲む必要があります。例えば、{INFA_TABLE_NAME} のようになります。

この機能は、データオブジェクトの【詳細】タブで設定します。

実行時にターゲットを作成または置換するためのルールおよびガイドライン

実行時にターゲットを作成または置換する場合、次のルールおよびガイドラインを検討してください。

- データベースのテーブルの中でもターゲットテーブルに循環型の依存関係がある場合は、データベースがテーブルを削除または作成するコマンドを実行できず、マッピングが失敗する。
- データ統合サービスがターゲットを置換した場合、ターゲットテーブルのインデックスと権限が保持されない。

- 動的ポートを含むように書き込みトランスフォーメーションを設定していない場合、データ統合サービスはデータオブジェクトに基づいてリンクされたポートとリンクされていないポートを含むターゲットを作成する。データはリンクされたポートに書き込まれます。
- データオブジェクト内のリソースがシノニムまたはビューであったとしても、データ統合サービスはテーブルを作成する。各接続は別々のデータベースインスタンスを指すことができますが、動的マッピングのすべての接続は、同じデータベースタイプである必要があります。

リレーショナルターゲットのプロパティにパラメータを割り当てる

類似したリレーショナルターゲットを使用して動的マッピングを実行するために、書き込みトランスフォーメーションでリソース、接続、テーブル所有者プロパティにパラメータを割り当てることができます。

同じデータベースで異なるが類似しているテーブルを使用してマッピングを実行するには、リソースのパラメータを使用します。リソースのパラメータを使用する場合、各ターゲットのデータオブジェクトを作成する必要はありません。接続用のパラメータを使用して異なるデータベースにアクセスします。

物理データオブジェクトでこれらのプロパティはパラメータ化できません。書き込みトランスフォーメーションのプロパティのパラメータを作成する場合、マッピングパラメータを作成します。

デフォルトでは、接続の接続タイプパラメータを作成します。テーブル所有者のテーブル名と文字列パラメータにリソースタイプパラメータを設定します。

この機能は、リレーショナルターゲットの書き込みトランスフォーメーションの【ランタイム】タブで設定します。

ターゲットデータオブジェクトにパラメータを割り当てる

カスタマイズされたデータオブジェクトにパラメータを割り当てて、実行時に書き込みトランスフォーメーションのソースを変更することができます。

複数のターゲットデータソースでモデルリポジトリにカスタマイズされたデータオブジェクトがある場合、データオブジェクトをパラメータ化します。パラメータの値を変更した場合、そのパラメータを使用するすべてのオブジェクトでターゲットが変更されます。

カスタマイズされたデータオブジェクトから書き込みトランスフォーメーションを作成した場合、トランスフォーメーションプロパティの【データオブジェクト】タブに、データオブジェクトに関する情報が表示されます。データオブジェクト名をクリックすると、モデルリポジトリから定義を表示することができます。データオブジェクトをパラメータ化するには、リソースタイプパラメータを作成するか、リソースパラメータを参照します。パラメータのデフォルト値は、モデルリポジトリ内のカスタマイズされたデータオブジェクトの名前です。デフォルトのパラメータ値を作成する場合、リポジトリ内のデータオブジェクトのリストからカスタマイズされたデータオブジェクト名を選択します。

データオブジェクトを変更すると、トランスフォーメーションポートが変更されます。トランスフォーメーションプロパティの【ポート】タブで、ポートを表示できます。

次の表に、【データオブジェクト】タブのパラメータオプションを示します。

パラメータオプション	説明
パラメータ	データオブジェクトとして設定したリソースパラメータの名前。読み取り専用。
説明	パラメータの説明。読み取り専用。

パラメータオプション	説明
新規	リソースパラメータを作成します。パラメータのデフォルト値としてモデルリポジトリ内のデータオブジェクトを参照し、選択します。
参照	リソースパラメータを参照し、パラメータを選択します。
デフォルト値	データオブジェクトに設定したリソースパラメータのデフォルト値。デフォルト値はカスタマイズされたデータオブジェクト名です。読み取り専用。

この機能は、リレーショナルターゲットの書き込みトランスフォーメーションの【データオブジェクト】タブで設定します。

動的ターゲットのルールおよびガイドライン

動的ターゲットに関する作業を行うときは、次のルールおよびガイドラインを検討してください。

- 動的ターゲットをプレビューする場合、Developer tool はスキーマ定義を更新しない。データソースからカラムを取得する設定や実行時にターゲットを置換する設定など、スキーマの変更に起因する不一致がある場合、データのプレビューは失敗します。読み取りまたは書き込みトランスフォーメーションを手動で同期します。エラーが解消されない場合、マッピングを実行して結果を確認します。
- 動的ターゲットが入力データに対して小さすぎる場合、マッピングは失敗し、値がカラムに対して大きすぎることを示すメッセージが表示される。
- ターゲットテーブルのデータ型は、書き込みトランスフォーメーションのデータ型と異なる場合がある。データ統合サービスがマッピングを実行すると、アップストリームトランスフォーメーションとターゲットテーブル間でデータ型が変更される場合があります。

動的ポートおよび生成されたポート

トランスフォーメーションで動的ポートを作成して、アップストリームトランスフォーメーションから新規または変更されたカラムを受信することができます。動的ポートは 1 つ以上のカラムを受信し、入力ルールに基づいてポートを生成します。入力ルールにより、動的ポートが受信して生成するカラムが決定されます。

動的ポートは、次のタスクの実行に使用します。

新規または変更されたカラムを受信する。

動的ソースまたはパラメータ化されたソースからデータを取得するために、ダウンストリームトランスフォーメーションに動的ポートを作成して、新規または変更されたカラムを受信します。マッピングに動的ソースが含まれている場合、ダウンストリームトランスフォーメーションの動的ポートは、自動的に新規または変更されたカラムを取得します。例えば、新規カラム「title」を動的ソースに追加すると、読み取りトランスフォーメーションはこの新規カラムを動的ポートに渡して、動的ポートは「title」カラムに対して生成されたポートを作成します。

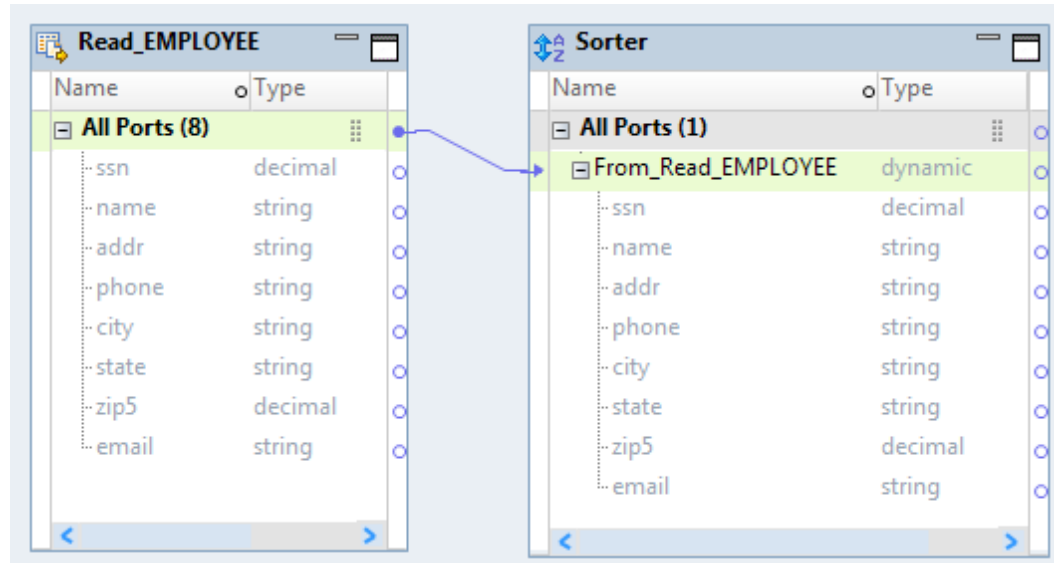
入力ルールに基づいてカラムをフィルタリングする。

トランスフォーメーションの特定タイプのカラムのみを処理するために、動的ポートを作成し、カラムをフィルタリングする入力ルールを定義します。例えば、マッピングソースに decimal、string、および date/time データ型のカラムがあるとします。decimal データ型のカラムのみでデータを処理する必要があります。この場合に、動的ポートを作成して、decimal 型のカラムのみを含める入力ルールを定義します。

式トランスフォーメーションの複数ポートで同じ計算を繰り返す。

複数のポートに対して同じ計算を実行するために、動的式で動的ポートを使用します。動的ポートの各ポートに対して動的式は1回実行され、動的出力ポートに結果が返されます。

次の画像は、[From_Read_Employee] という動的ポートと生成されたポートを示しています。



動的ポートおよび生成されたポートの設定

読み取りトランスフォーメーションの [すべてのポート] グループ、アップストリームトランスフォーメーションのグループ、またはアップストリームトランスフォーメーションの動的ポートから動的ポートを作成できます。Developer tool はデータ型の値が動的である動的ポートを作成します。トランスフォーメーションでは、複数の動的ポートを作成できます。

【新規】 ボタンを使用してポートを作成した場合は、Developer tool によってデフォルト名が割り当てられます。動的ポートの名前を変更して、各トランスフォーメーション内でポート名が一意になるようにします。同じ名前のポートをトランスフォーメーションに追加した場合、Developer tool はポート名の競合を解決するために動的ポートまたは生成されたポートに数字を追加します。

次の各トランスフォーメーションで動的ポートを作成できます。

アグリゲータ
式
フィルタ
ジョイナ
ルックアップ
ランク
読み取り
ルータ
シーケンスジェネレータ
ソータ
アップデートストラテジ
書き込み

マッピングに動的ポートを含むことができないトランスフォーメーションが含まれる場合、ソースメタデータが変更されたときにマッピングを手動で更新する必要がある場合もあります。

注: ポート属性への変更はすべて、パイプラインの生成されたポートにプロパゲートされます。変更されたポート属性を手動でプロパゲートする必要はありません。

動的ポートおよび生成されたポートのルールおよびガイドライン

動的ポートおよび生成されたポートに関する作業を行うときは、次のルールおよびガイドラインを検討してください。

- 生成されたポートを仮想テーブルマッピングの出力トランスフォーメーションにリンクすることはできない。
- 生成されたポートを操作マッピングのフォールト、入力、または出力トランスフォーメーションにリンクすることはできない。

動的式

動的出力ポートの式を設定すると、式が動的式になります。動的式では複数の出力ポートを生成できます。

動的式では、ポートセクタまたは動的ポートを参照できます。ポートセクタまたは動的ポートに複数のポートが含まれる場合、動的式は各ポートに対して実行されます。

動的式を設定すると、Developer tool は、生成されたポートが式に対して有効なタイプかどうかを検証しません。例えば、string 型を必要とする式で decimal 型ポートを含むポートセクタを参照する場合、設計時に式は有効と表示されます。

例

式トランスフォーメーションには、生成された以下の入力ポートがあります。

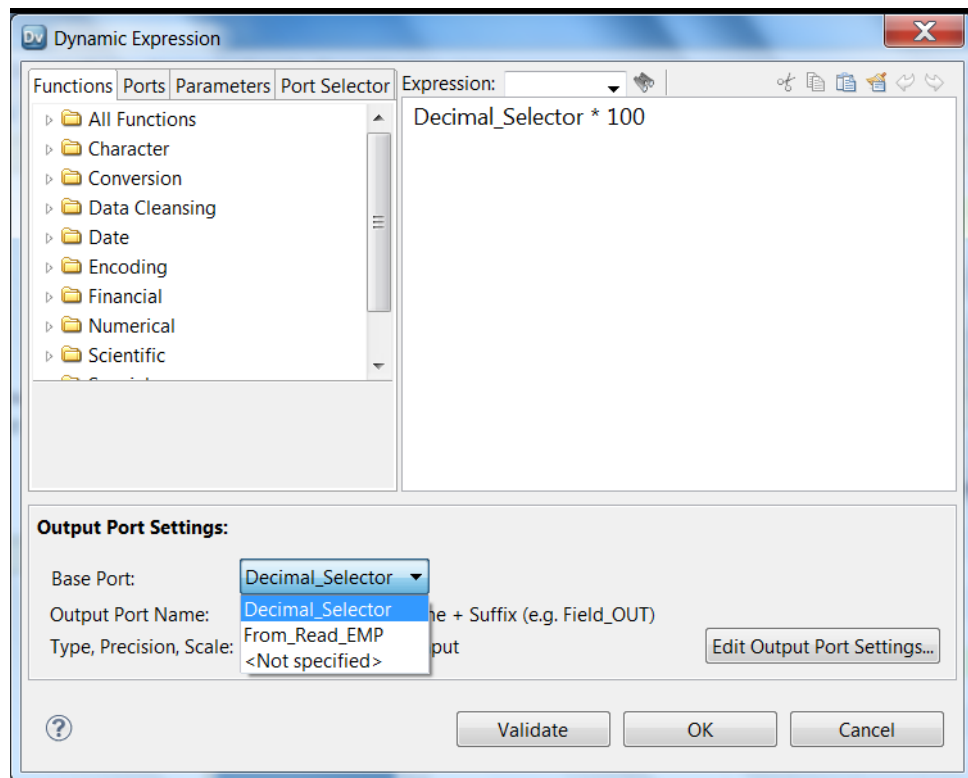
```
EMPNO    Decimal
NAME      String
SALARY    Decimal
DEPTNO    Decimal
```

トランスフォーメーションには MyDynamicPort という名前の動的出力ポートが含まれます。出力ポートは動的式の結果を返します。動的式は、ポートセクタの各ポートの値に 100 をかけます。式はポートセクタの各ポートに対して 1 回実行されます。各インスタンスは異なる結果を返すことがあります。式トランスフォーメーションは、各結果に対して別の出力ポートを生成します。

Decimal_Selector ポートセクタには、decimal データ型のポートを含む選択ルールがあります。

```
EMPNO    Decimal
SALARY    Decimal
DEPTNO    Decimal
```

次の図は、Decimal_Selector ポートセクタを参照する動的式を示しています。



出力ポート設定を編集して、出力ポート名と出力ポートプロパティを変更します。また、ベースポートも選択できます。

入力ルール

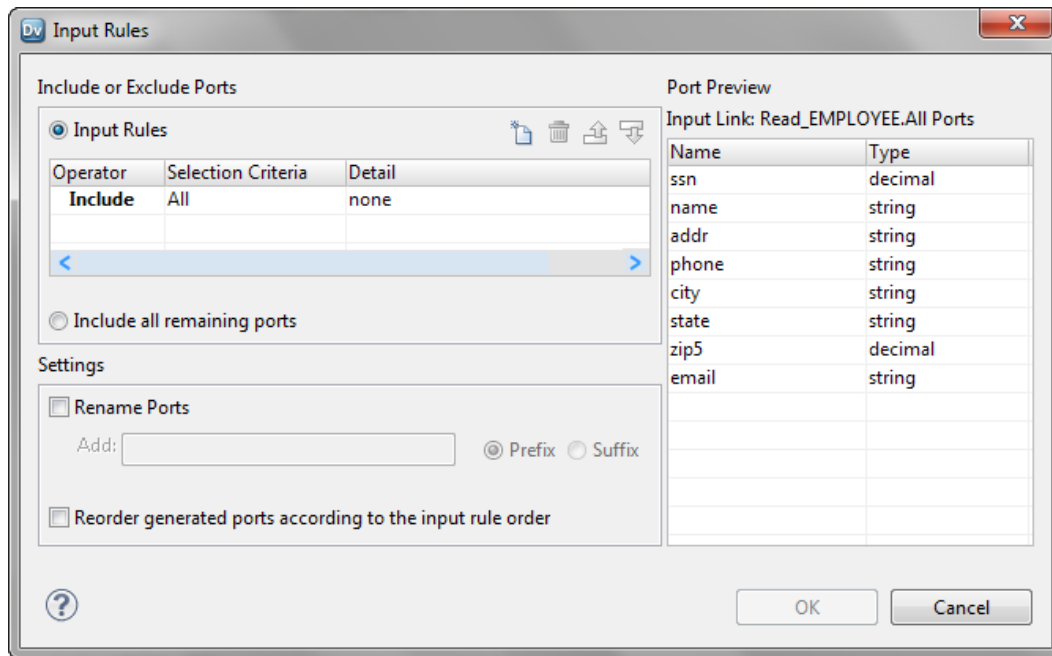
入力ルールとは、動的ポート内の生成するポートを定義する条件のセットです。動的ポートが受信するカラムをフィルタリングして、フィルタリングされたカラムに生成されたポートを作成するよう、入力ルールを設定できます。入力ルールを使用してマッピングの特定カラムをダウンストリームにプロパゲートします。

動的ポートの特定のカラムまたは特定タイプのカラムを処理するには、名前またはデータ型に基づいてカラムをフィルタリングするよう、入力ルールを定義します。例えば、従業員データベーステーブルに decimal、string、および date/time データ型のカラムがあるとします。SAL で始まりデータ型が decimal のカラムのみでデータ処理を行う必要があります。動的ポートを作成して、この条件を満たすカラムのみを含めるよう入力ルールを定義します。

入カールールの設定

【入カールール】 ダイアログボックスを使用して、含めるポートの定義、生成されたポートの名前の変更、生成されたポートの順序変更、およびルールの結果の表示を行います。

次の画像は、アップストリームトランスフォーメーションからのすべてのポートを含めるデフォルトの入カールールを設定した **【入カールール】** ダイアログボックスを示しています。



入カールールを設定した場合、次のプロパティを設定します。

ポートを含めるまたは除外する

ポート名またはデータ型に基づき、動的ポートに含めるまたは除外するポートを指定します。複数のルールを定義できます。データ統合サービスは、入カールールリストに表示される順序でルールを適用します。デフォルトの入カールールではすべてのポートが含まれます。動的ポートの「含む」入カールールを1つ以上作成します。

残りのすべてのポートを含める

トランスフォーメーションの他の動的ポートから除外されたポートを追加します。トランスフォーメーションに複数の動的ポートが含まれる場合、最後の動的ポートのアップストリームトランスフォーメーションから残りのすべてのポートを含めることができます。

生成されたポートの名前の変更

生成されたポート名にプレフィックスまたはサフィックスを追加します。ポートが生成されるトランスフォーメーションを示すか、各トランスフォーメーション内でポート名が一意になるようにするには、プレフィックスまたはサフィックスを使用します。

生成されたポートの順序変更

入力したルールの順序に従って生成されたポートを表示します。デフォルトでは、Developer tool はアップストリームトランスフォーメーションに表示されるのと同じ順序でポートを表示します。

ルールを設定した後で、生成されたポートをプレビューしてルールの組み合わせを確認できます。データ統合サービスは、**【入カールール】** ダイアログボックスに表示される順序でルールを評価します。正しい順序で実行されるように、ルールの順序を変更できます。

ポートを含めるまたは除外する

ポート名またはデータ型に基づいてポートを含めるまたは除外することができます。各入力ルールでは演算子および選択条件を使用してポートをフィルタリングします。複数のルールを定義できます。データ統合サービスは、入力ルールリストに表示される順序でルールを適用します。デフォルトの入力ルールではすべてのポートが含まれます。

次の入力ルールを設定して、含めるまたは除外するポートを決定します。

演算子

ポートを含めるか除外するかを決定します。デフォルトの設定ではポートを含めます。

選択条件

ポート名またはデータ型に基づいてポートをフィルタリングするかどうかを決定します。選択条件を選択した場合、条件に基づいて入力ルールの詳細ダイアログボックスが表示されます。例えば、**【名前】** 選択条件の詳細は **【入力ルールの詳細: 名前リスト順】** ダイアログボックスで指定します。

詳細

入力したポート名またはデータ型の詳細に基づいてフィルタリングするポートを決定します。

次の表に、選択条件と条件の詳細を指定する方法を示します。

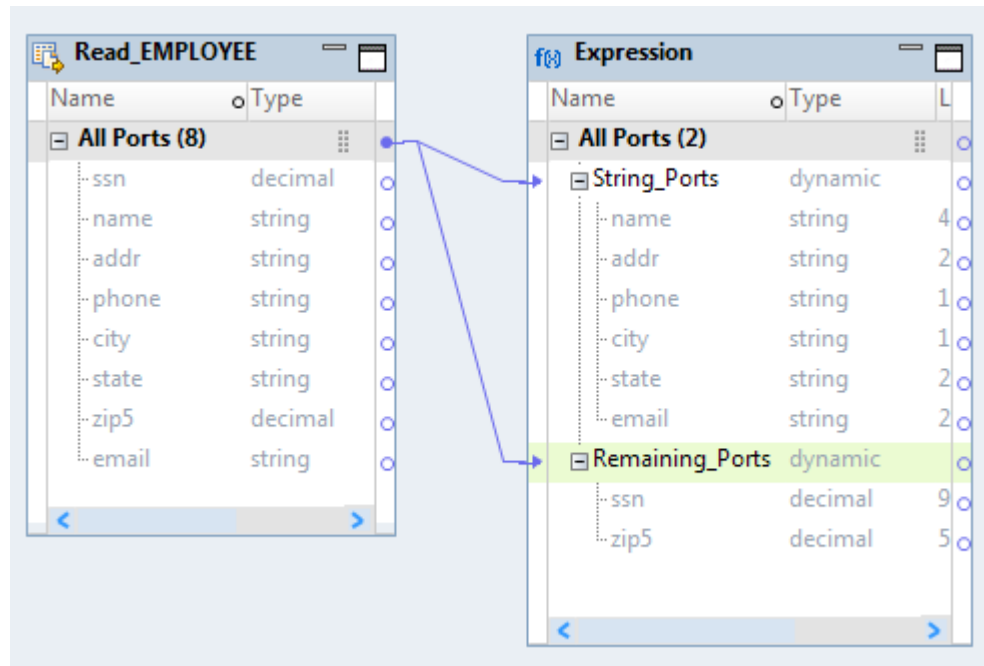
選択条件	説明	条件の詳細
すべて	すべてのポートを含めます。除外演算子とともにこの選択条件を使用しないでください。	詳細を指定する必要はありません。
名前	ポート名に基づいてポートをフィルタリングします。	値のリストからポート名を選択するか、[ポート] または [ポートリスト] のパラメータを使用します。 注: 名前の値の大文字と小文字は区別されません。
タイプ	ポートのデータ型に基づいてポートをフィルタリングします。	リストからデータ型を選択します。
パターン	ポート名のパターンに基づいてポートをフィルタリングします。	ポート名のパターンタイプとして、プレフィックス、サフィックス、または正規表現を選択します。次に、パターンの値を入力するか、文字列タイプのパラメータを使用します。 注: パターンの値の大文字と小文字は区別されません。

残りのすべてのポートを含める

トランスフォーメーションに複数のポートが含まれる場合、他の動的ポートに含まれなかったすべてのポートを含めるように最後の動的ポートを設定できます。

例えば、テーブルの string 型カラムから先頭の空白を削除し、他のすべてのカラムに対するデータとともに string 型データの出力をターゲットに書き込む場合を想定します。式トランスフォーメーションで、2 つの動的ポートを作成します。一方のポートにすべての string 型データを含めて、残りのすべてのデータをもう一方のポートに配置するように入力ルールを設定します。最後の動的ポートの **【残りのすべてのポートを含める】** オプションを選択します。

次の画像は、式トランスフォーメーションの2つの動的ポート String_Ports および Remaining_Ports を示しています。

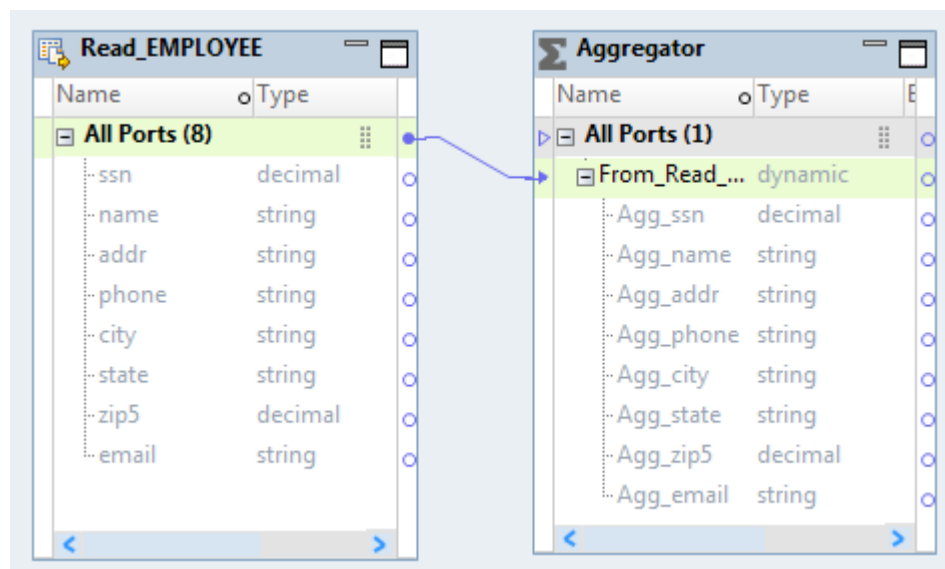


生成されたポートの名前の変更

生成されたポートにプレフィックスまたはサフィックスを付けて名前を変更する設定を使用して、一意のポート名を生成できます。

例えば、Agg_プレフィックスを追加すれば、アグリゲータトランスフォーメーションで生成されたポートであることを示すことができます。

次の画像は、アグリゲータトランスフォーメーションの名前の変更により Agg_プレフィックスがついた生成されたポートを示しています。



同じ名前のポートをトランスフォーメーションに追加する場合、Developer tool は生成されたポートに数字を追加してポート名の競合を解決します。データ統合サービスが実行時にポート名の競合を解決できない場合、生成されたポートの名前を変更する必要がある場合もあります。マッピングで動的ソースを使用していると、データ統合サービスで実行時にポート名の競合が発生する場合があります。データ統合サービスでポート名の競合が発生した場合、生成されたポートの名前を変更しようと試みます。データ統合サービスがポート名の競合を解決できないと、マッピングは失敗します。マッピングが失敗するのは次のような場合です。

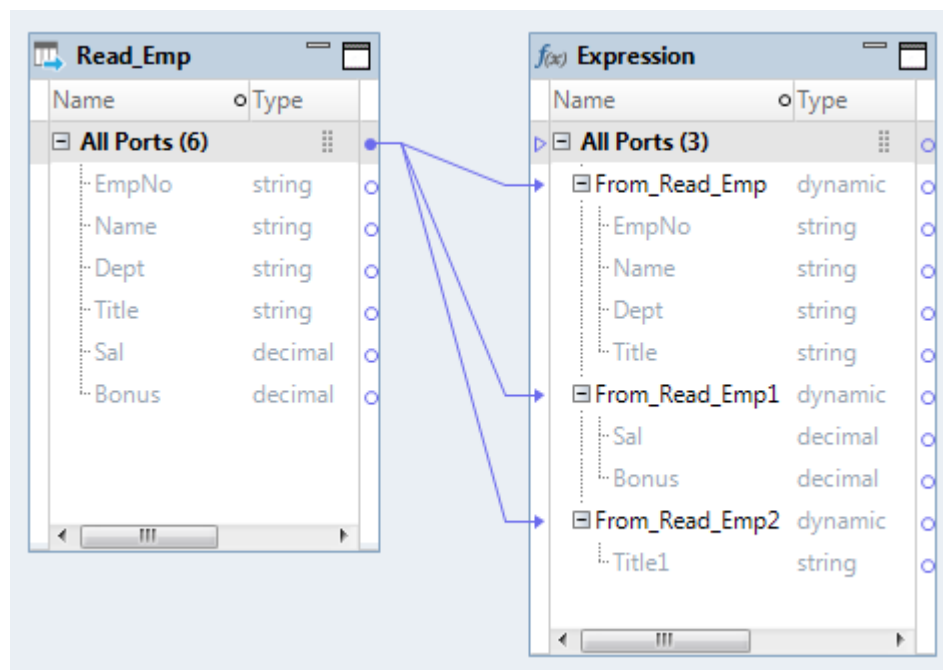
- 名前が変更された生成されたポートから静的ポートへの未解決のリンクが存在する。
- グループ化ポートや結合条件などのトランスフォーメーションプロパティで名前が変更された生成されたポートを使用している。

マッピングエラーを回避するには、各トランスフォーメーション内で一意になるように生成されたポートの名前を変更します。

生成されたポートの名前の変更の例

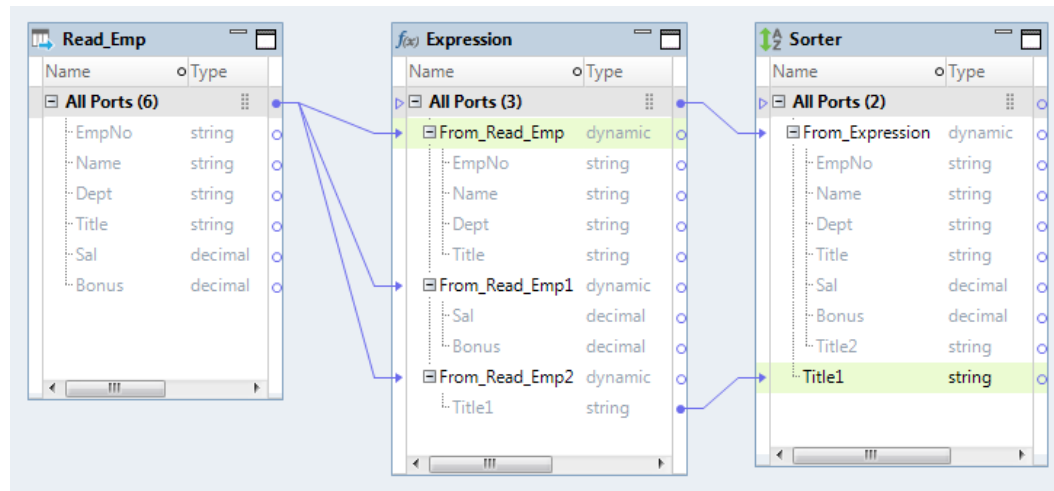
式トランスフォーメーションには3つの動的ポートがあります。動的ポート [From_Read_Emp] および [From_Read_Emp2] には、生成されたポート「Title」が含まれます。名前の競合を回避するため、Developer tool は [From_Read_Emp2] の生成されたポートの名前を「Title1」に変更します。

次の画像は、式トランスフォーメーションの名前が変更された生成されたポート Title1 を示しています。



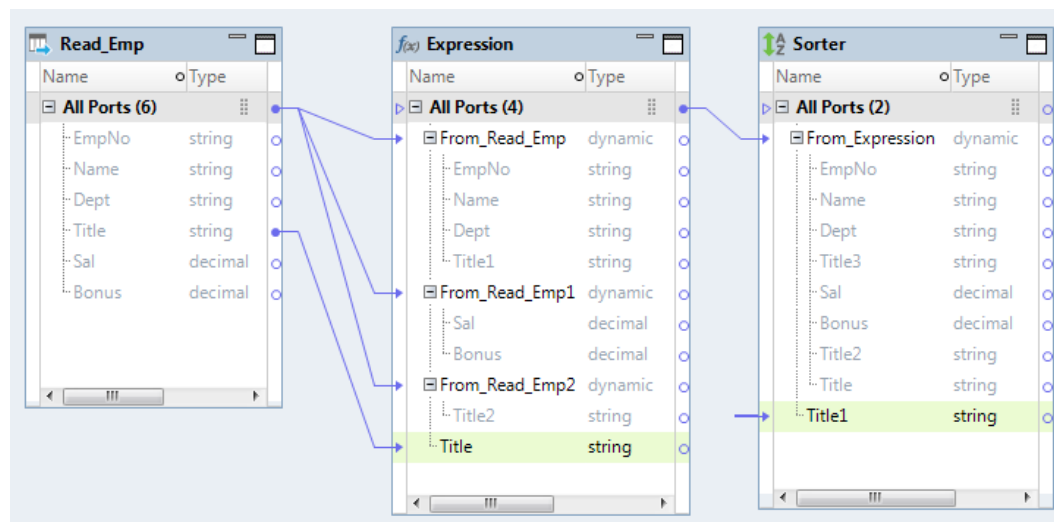
式トランスフォーメーションの生成されたポート「Title1」をソータートランスフォーメーションのポート「Title1」にリンクします。また、ソートキーとしても「Title1」を使用します。

次の画像は、式トランスフォーメーションの生成されたポートからソータートランスフォーメーションのポートへのリンクを示しています。



読み取りトランスフォーメーションのポート「Title」から式トランスフォーメーションのポート「Title」へのリンクを追加する場合、Developer tool は生成されたポートの名前を変更します。動的ポート [From_Read_Emp] の生成されたポートを「Title1」に名前変更します。動的ポート [From_Read_Emp2] の生成されたポートを「Title2」に名前変更します。ソータートランスフォーメーションの「Title1」へのリンクは未解決と表示されます。

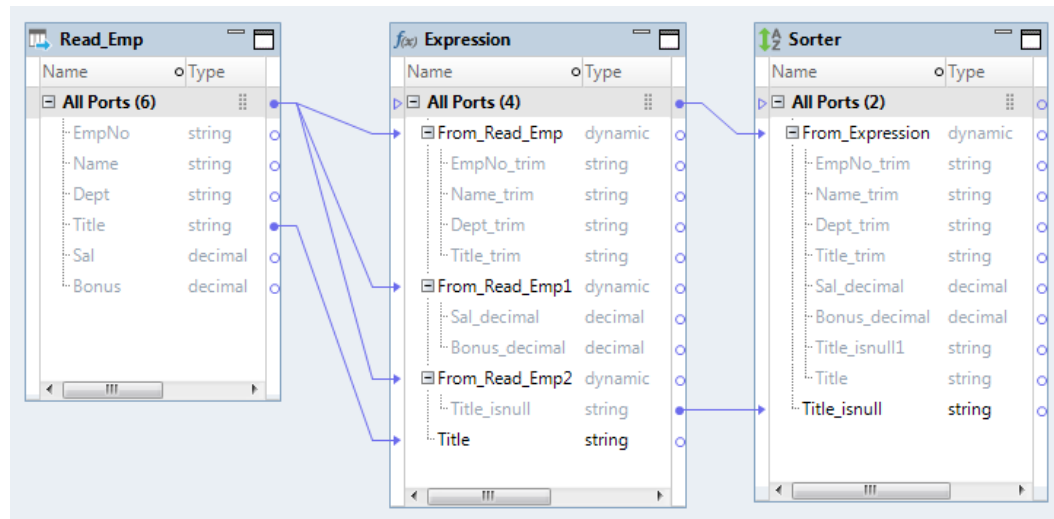
次の画像は、読み取りトランスフォーメーションと式トランスフォーメーション間の新しいリンク、Developer tool が式トランスフォーメーションで名前変更した生成されたポート、およびソータートランスフォーメーションへの未解決のリンクを示しています。



ソートキーに使用されている生成されたポートが使用を意図したポートではない可能性があるため、マッピングは実行時に失敗します。

マッピングエラーを回避するには、各トランスフォーメーション内で一意になるように生成されたポートの名前を変更します。例えば、動的ポート [From_Read_Emp] の文字列ポートの先頭の空白を切り捨てるとします。サフィックス_trim を生成されたポートに追加します。動的ポート [From_Read_Emp2] のポートに NULL 値があるかどうか判定する必要があります。サフィックス_isnull を生成されたポートに追加します。

次の画像は、式トランスフォーメーションの生成されたポートが名前変更された様子を示しています。



生成されたポートの順序変更

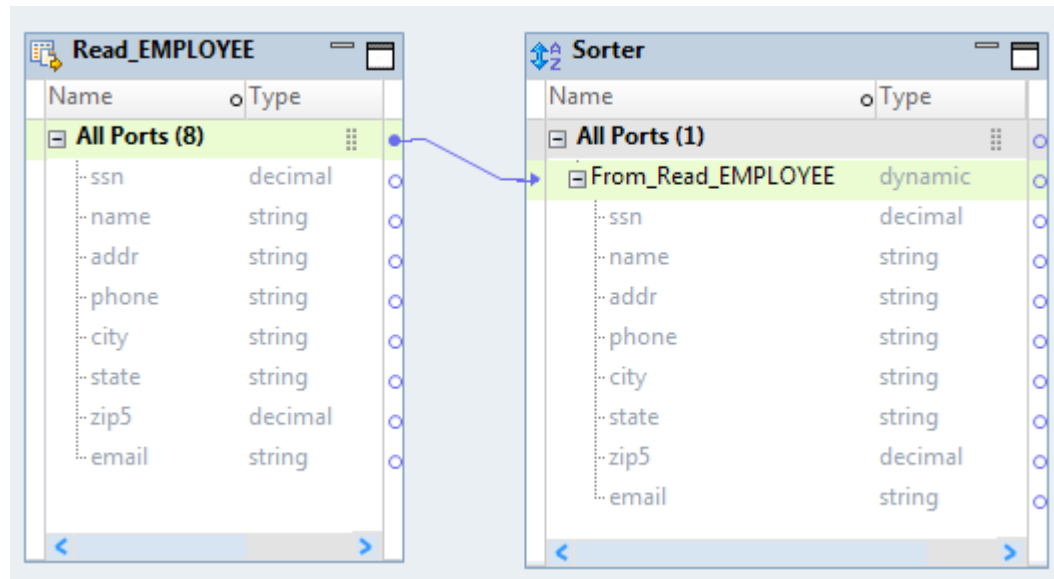
入力ルールに基づいてポートの順序を変更する設定を使用して、生成されたポートの順序を変更できます。

データ統合サービスは、**【入力ルール】** ダイアログボックスに表示された順序でルールを読み取ります。ポートの順序を確認して、入力ルールの順序に基づいて順序を変更してください。データ統合サービスがポートおよびルールを要求に応じた順序で処理するようにできます。ポートを順序変更すると、結果の表示および分析もしやすくなります。

生成されたポートの順序変更の例

従業員フラットファイルソースには頻繁に変更されるカラムが多数あります。ユーザーは、従業員を名前でソートし、先頭のカラムに従業員名、次のカラムに従業員が勤務する市区町村が出現するように従業員データを表示する必要があります。また、decimal 型タイプのカラムは、データ分析の対象外であるため、末尾に移動する必要があります。

次の画像は、動的ポート [From_Read_EMPLOYEE] を示しています。生成されたポートは、元の順序で表示されています。

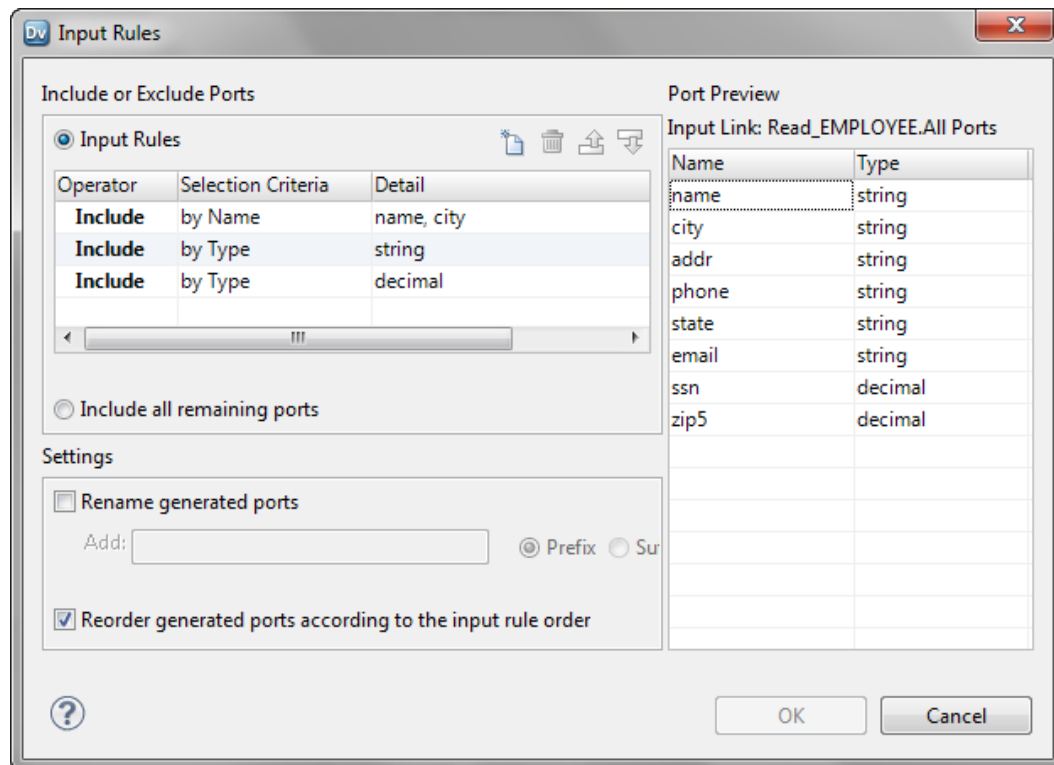


次の入力ルールを設定します。

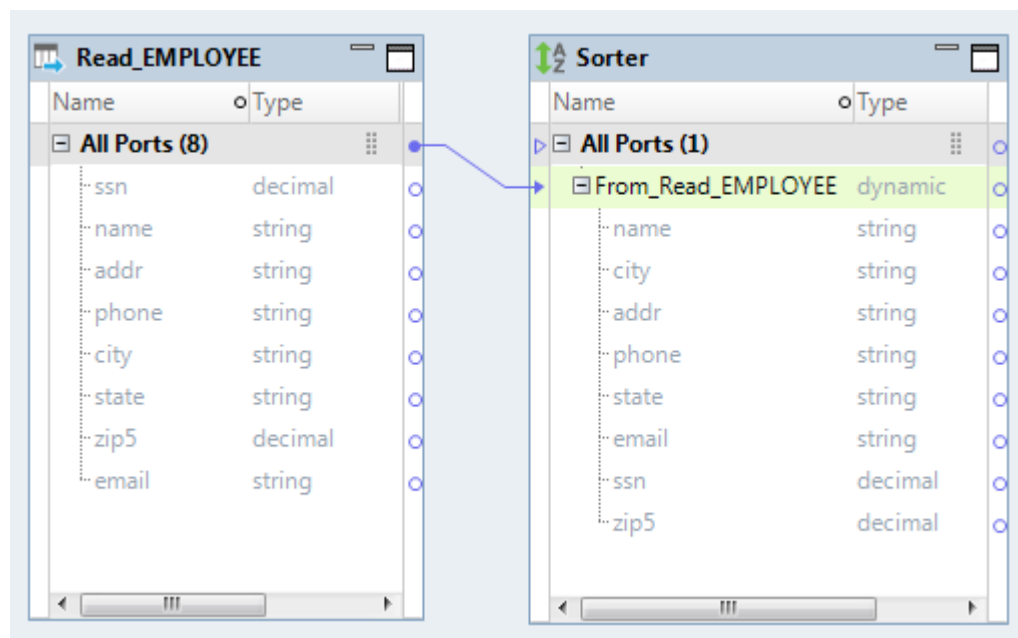
- 「名前」と「市区町村」によってポートを含める。
- string 型によってすべてのポートを含める。
- decimal 型によってすべてのポートを含める。

次に、入力ルールの順序に基づいてポートの順序を変更します。ポートの順序は、プレビューで確認できます。

次の画像は、入力規則の順序を示しています。



次の画像は、入力規則設定に基づいて順序変更された生成されたポートを示しています。



選択ルールとポートセレクト

トランスフォーメーションでポートを生成した場合、生成したポートが変更されてもトランスフォーメーションが正しく実行されるように設定する必要があります。ポートセレクトを使用して、動的式、ルックアップ条件、またはジョイナ条件でどのポートを使用するかを決定できます。

ポートセレクトとは、ポートを定義する選択ルールのセットです。ポートセレクトは式内で参照します。生成したポートが動的マッピング内で変更された場合、ポートセレクトに異なるポートが含まれている可能性があります。ポートセレクトは、式トランスフォーメーション、ルックアップトランスフォーメーション、またはジョイナトランスフォーメーション内で作成できます。これらのトランスフォーメーションには、ポートセレクト内のすべてのポートを参照できる式が含まれます。

以下のマッピングオブジェクトでポートセレクトを設定できます。

式トランスフォーメーション

動的式内でポートセレクトを参照できます。式内でポートセレクトを参照すると、その式がポートセレクト内の各ポートに対して実行されます。動的式は、ポートセレクト内のポートごとに別個の出力ポートに結果を返します。トランスフォーメーションにポートセレクトを参照する複数の式が含まれている場合、そのトランスフォーメーションは、式ごとに追加の出力ポートを返します。

ジョイナトランスフォーメーション

1つの結合条件内で2つのポートセレクトを参照できます。マスタグループ用のポートセレクトと詳細グループ用のポートセレクトを定義します。データ統合サービスは、ポートセレクト内のポート順に従って、マスタグループ内の各ポートを詳細グループ内のポートと比較します。比較するポートのペアごとに1つのタイプの演算子を選択できます。各ポートセレクトには、同じ数のポートが含まれている必要があります。

例えば、ポート A、B、C を含むポートセレクト Master-SelectorX、およびポート D、E、F を含むポートセレクト Detail-SelectorY を設定するとします。このとき、Master-SelectorX = Detail-SelectorY という結合条件を指定すると、Developer tool によって $A = D \text{ AND } B = E \text{ AND } C = F$ という結合条件が作成されます。

ルックアップトランスフォーメーション

ポートのポートセレクトはルックアップ条件で設定できます。データ統合サービスは、各ポートセレクト内のポート順に従って、入力ポートセレクト内の各ポートをルックアップポートセレクト内のポートと比較します。各ポートセレクトには、同じ数のポートが含まれている必要があります。

書き込みトランスフォーメーション

書き込みトランスフォーメーションでポートのポートセレクトを設定できます。

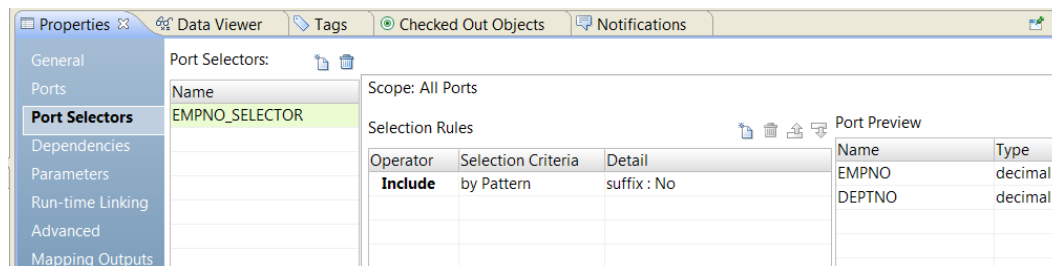
データをリレーショナルターゲットまたは Hive ターゲットに書き込む場合、実行時にターゲットテーブルを作成または置換できます。DDL クエリを定義できます。データ統合サービスは実行時にこの DDL クエリに基づいてターゲットテーブルを作成または置換する必要があります。DDL クエリでポートセレクトを設定することもできます。

ポートセレクトの設定

ポートセレクトを設定するには、選択ルールを定義して、含める生成済みポートを決定します。選択ルールは、動的ポートに設定できる入力ルールと同じです。

ポートセレクトには、静的または生成済みポートを含めることができます。ポートセレクトの設定は、**ポートセレクト**タブで行います。

次の図は、**ポートセレクト** タブを示しています。



ポートセレクトに次のプロパティを設定します。

名前

ポートセレクトを識別します。1つのトランスフォーメーションに複数のポートセレクトを作成し、式で参照できます。

スコープ

ポートセレクトが適用されるポートグループを識別します。ジョイナまたはルックアップトランスフォーメーションのポートセレクトを作成する場合、スコープを選択する必要があります。これらのトランスフォーメーションには複数の入力グループがあります。ジョイナトランスフォーメーションにはマスタまたは詳細スコープがあります。ルックアップトランスフォーメーションにはインポートまたはルックアップスコープがあります。式トランスフォーメーションには入力グループが1つあります。スコープは常に「すべてのポート」です。

選択ルール

ポートセレクトに含めるポートを決定します。選択ルールを作成すると、**【ポートのプレビュー】** パネルに現在の入力ポートのうち適格なポートが表示されます。これらのポートは変わる可能性があります。さまざまなソースからのポートに対応できるように選択ルールを設定します。

選択ルール

ポートセレクトに関連付けられた選択ルールでポートセレクトに含めるポートを決定します。

選択ルールを作成すると、**【ポートのプレビュー】** パネルに現在の入力ポートのうち適格なポートが表示されます。これらのポートは変わる可能性があります。さまざまなソースからのポートに対応できるように選択ルールを設定します。

次の条件に基づいて選択ルールを作成します。

演算子

選択ルールが返すポートを含めるか、除外します。デフォルトは「含める」です。ポートを除外する前にポートを含める必要があります。

選択条件

作成する選択ルールのタイプ。ポートタイプまたはカラム名に基づいてルールを作成できます。カラム名に基づいてポートを含めるには、特定の名前を検索するか、名前に含まれる文字列パターンを検索します。

詳細

選択条件に適用する値。選択条件がカラム名基準になっている場合は、検索する文字列または名前を設定します。選択条件がポートタイプ基準になっている場合は、含めるポートタイプを選択します。

次の表に、選択条件と条件の詳細を指定する方法を示します。

選択条件	説明	詳細
すべて	すべてのポートを含めます。	詳細は不要です。
名前	ポート名に基づいてポートをフィルタリングします。	値のリストからポート名を選択するか、[ポート] または [ポートリスト] のパラメータを使用します。
タイプ	各ポートのデータ型に基づいてポートをフィルタリングします。	リストからデータ型を選択します。
パターン	名前に含まれる文字列または正規表現を使用してポートをフィルタリングします。	ポート名のパターンタイプとして、プレフィックス、サフィックス、または正規表現を選択します。次に、パターンの値を入力するか、文字列タイプのパラメータを使用します。

選択ルールとポートセレクタの例

動的ソースを使用するようにマッピングを設定できますが、各ソースファイルの給与情報が含まれるカラムは名前が異なります。それぞれのソースのカラム名は Salary、Monthly_Salary、または Base_Salary です。

給与を示すあらゆるポート名を持つ式を実行するために、次のタスクを実行します。

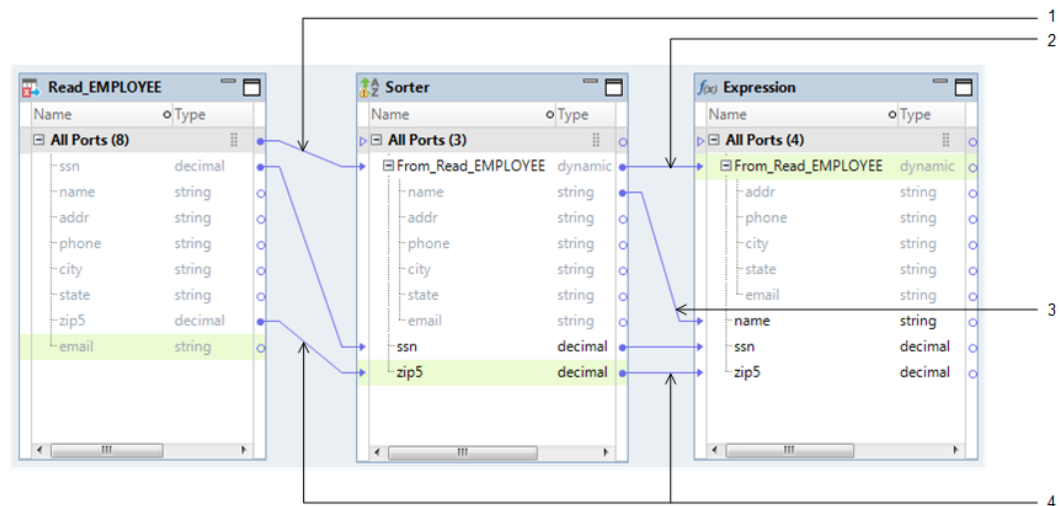
1. 「Salary_PortSelector」というポートセレクタを作成します。
2. サフィックスが「Salary」のあらゆるポート名を受け入れる選択ルールを作成します。
3. 特定のカラム名ではなくポートセレクタ名を含めるように、式を設定します。式の構文は次のとおりです。
Salary_PortSelector * 12

設計時リンク

動的マッピングを設計するときにさまざまなタイプのリンクを作成できます。2つのポート間のリンク、グループと動的ポート間のリンク、2つの動的ポート間のリンク、および生成されたポートから静的ポートへのリンクを作成できます。

設計時リンク（リンク）とは、マッピング内で作成する直接リンクです。トランスフォーメーションは、マッピングを設計するときに直接リンクを作成できないように変更される場合があります。設計時にリンクを作成できない場合、実行時にリンクするポートを決定するランタイムリンクを設定できます。

次の画像は、動的マッピングのリンクを示しています。



1. 特定のグループから動的ポートへのリンク
2. 2つの動的ポート間のリンク
3. 生成されたポートから静的ポートへのリンク
4. 2つのポート間のリンク

マッピングを設計するときに次のタイプのリンクを作成できます。

グループから動的ポートにリンクする。

特定のグループから動的ポートへのリンクを作成した場合、1つ以上のコラムのデータがプロパゲートされます。1つのグループには、1つ以上のポートおよび動的ポートを含めることができます。動的ポートの下に表示される生成されたポートは、動的ポートの入力ルールによって決定されます。デフォルトルールでは、グループ内のすべてのコラムが、ダウンストリームトランスフォーメーションの動的ポート内の生成されたポートとして含められます。

例えば、前の画像は、読み取りトランスフォーメーションの「すべてのポート」グループからソータートランスフォーメーションの動的ポート「From_Read_EMPLOYEE」へのリンクを示しています。ソータートランスフォーメーションの動的ポート「From_Read_EMPLOYEE」の入力ルールには文字列ポートが含まれます。

2つの動的ポートをリンクする。

2つの動的ポート間にリンクを作成した場合、1つ以上のコラムのデータがプロパゲートされます。動的ポートの下に表示される生成されたポートは、動的ポートの入力ルールによって決定されます。デフォルトルールでは、アップストリームの動的ポートからのすべてのコラムが、ダウンストリームトランスフォーメーションの動的ポート内の生成されたポートとして含められます。

例えば、前の画像は、ソータートランスフォーメーションの動的ポート「From_Read_EMPLOYEE」から式トランスフォーメーションの別の動的ポート「From_Read_EMPLOYEE」へのリンクを示しています。式トランスフォーメーションの動的ポートの入力ルールでは、文字列ポートは含められ、[名前]ポートは除外されます。

生成されたポートを静的ポートにリンクする。

生成されたポートから特定のポートへのリンクを作成すると、単一のコラムのデータがプロパゲートされます。

例えば、前の画像は、ソータートランスフォーメーションの動的ポート「From_Read_EMPLOYEE」の下にある生成されたポート「名前」から式トランスフォーメーションのポート「名前」へのリンクを示しています。

2つの静的ポートをリンクする。

他のマッピングの場合と同じ方法でトランスフォーメーション間のポートをリンクします。

関連項目：

- [「ポートのリンク」 \(ページ 22\)](#)

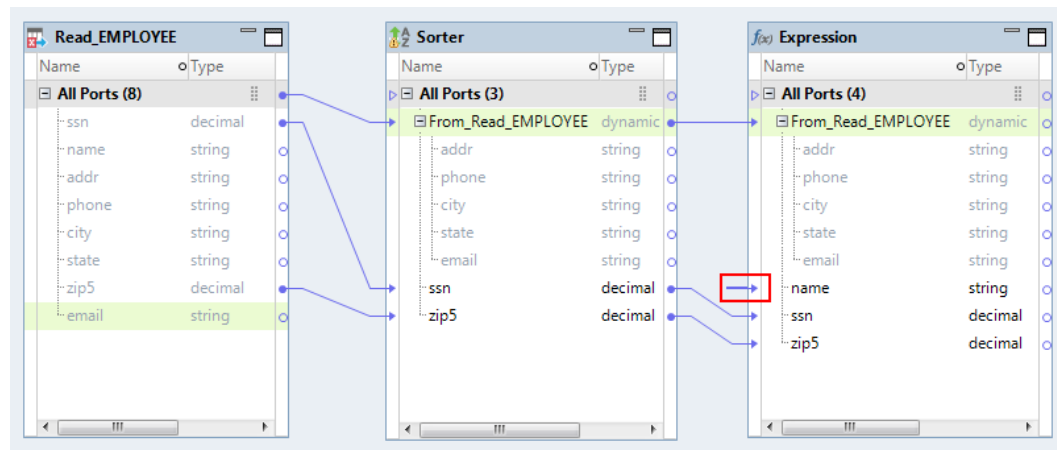
リンクの解決

動的ポート内の生成されたポートは、動的ソースまたは入力ルールに基づいて変更される可能性があります。

生成されたポートからのリンクを作成した場合、生成されたポートが使用不可になると、Developer tool は、そのポートへのリンクを未解決リンクとして表示します。

例えば、ユーザーがソータートランスフォーメーションの動的ポート [From_Read_EMPLOYEE] の入力ルールを更新し、[名前] ポートを除外したとします。その場合、Developer tool は、そのリンクを未解決リンクに変更します。

次の画像は、式トランスフォーメーションのポート [名前] への未解決リンクを示しています。



マッピングを検証すると、Developer tool は未解決リンクに関する警告メッセージを表示します。マッピングの実行時、生成されたポートが使用可能であれば、データ統合サービスは、そのリンクを解決してマッピングを処理します。ただし、データ統合サービスがリンクを解決できない場合は、マッピングが失敗します。マッピングを正常に実行するには、未解決リンクを削除する必要があります。トランスフォーメーションを右クリックし、[未解決リンクのクリア] を選択して、トランスフォーメーションの未解決リンクをすべてクリアします。

ランタイムリンク

ランタイムリンクとは、実行時にポートが変更される可能性があるグループ間のリンクです。データ統合サービスは、ポリシーおよびパラメータに基づいて実行時にリンクするポートを決定します。

実行時にアップストリームトランスフォーメーションのポートが変更される可能性がある場合は、マッピングオブジェクトのグループ間でランタイムリンクを作成します。ポートが実行時に変更される可能性があるとしても、マッピングの設計時にポートをリンクすることはできません。パラメータおよびリンクポリシーを使用して実行時にリンクするポートを決定できるランタイムリンクを作成します。

次の場合に、ランタイムリンクを作成します。

データソースからカラムを取得するため、またはパラメータによって定義されるソースを使用するために読み取りトランスフォーメーションを設定する場合。

例えば、読み取りトランスフォーメーションがパラメータを使用してソースを変更するか、実行時にソースからメタデータの変更を取得するとします。ダウンストリームトランスフォーメーションは、マッピング実行から次のマッピング実行の間に変更される可能性がある生成されたポートから、ポート経由でデータを受信します。ダウンストリームトランスフォーメーションへのランタイムリンクを作成して設定します。実行時に、データ統合サービスはリンクポリシーまたはパラメータ値に基づいてポートを接続します。

データソースまたはデータオブジェクトからカラムを取得するため、またはパラメータによって定義されるターゲットを使用するために書き込みトランスフォーメーションを設定する場合。

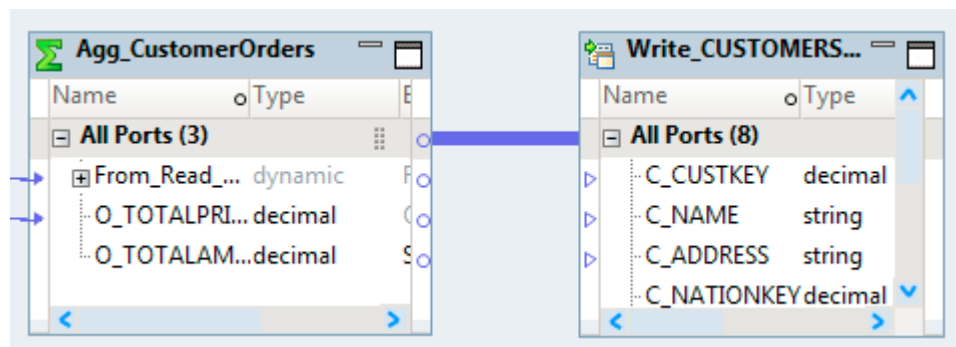
例えば、書き込みトランスフォーメーションが関連付けられているデータオブジェクトに基づいてカラムを定義するとします。書き込みトランスフォーメーションは、パラメータを使用してターゲットを変更するか、実行時にターゲットからメタデータの変更を取得します。書き込みトランスフォーメーションへのランタイムリンクを作成して設定します。

注: マッピングフローに基づいてターゲットカラムを定義する場合は、書き込みトランスフォーメーションへのランタイムリンクを作成しないでください。

実行時に、データ統合サービスはリンクポリシーまたはパラメータ値に基づいてポートを接続し、ダウンストリームポートにデータを渡します。

実行時にアップストリームトランスフォーメーションのポートが変更される可能性がある場合は、トランスフォーメーションのグループ間でランタイムリンクを作成します。データ統合サービスは、定義されたパラメータ、リンクポリシー、またはその両方に基づいて、実行時にリンクするポートを決定します。ランタイムリンクは、マッピングエディタで太線で表示されます。

次の画像は、アグリゲータトランスフォーメーションと書き込みトランスフォーメーション間のランタイムリンクを示しています。

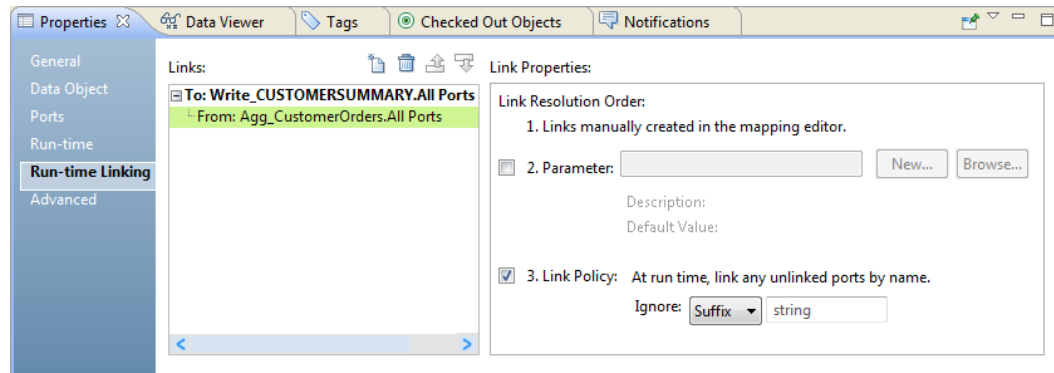


ランタイムリンクの設定

実行時にマッピングオブジェクト間でリンクするポートを決定するようにランタイムリンクプロパティを設定します。パラメータを定義するか、リンクポリシーを選択するか、またはその両方を使用して、リンクするポートを決定できます。

[ランタイムリンク] ダイアログボックスまたは **[ランタイムリンク]** タブを使用してランタイムリンクプロパティを設定します。

次の画像は、書き込みトランスフォーメーションの【ランタイムリンク】タブを示しています。



データ統合サービスは、実行時に、次の順番でポート間のリンクを確立し、解決します。

1. マッピングエディタで手動で作成するリンク。
2. ランタイムリンク用に設定したパラメータに基づくリンク。
3. ランタイムリンク用に設定したリンクポリシーに基づくリンク。

ランタイムリンクの次のプロパティを設定します。

トランスフォーメーションの選択

【リンク】領域で、【新規】ボタンをクリックし、【新しいリンク】ダイアログボックスで、実行時にポートをリンクするトランスフォーメーションを選択します。トランスフォーメーションに入力ランタイムリンクがある場合、【リンク】領域には、リンクの起点となるグループポートが表示されます。

パラメータの設定

マッピング実行から次のマッピング実行の間にポート名が変更される可能性があり、ポート名の値がわかっているときは、パラメータを使用します。入力リンクセットタイプのパラメータを使用して、マッピング実行ごとに名前値でポートを接続します。

例えば、新しい入力リンクセットタイプの `Cust_InputLinkSet` というパラメータを作成して、`C_Name` -> `Cust_name` のようにデフォルト値を指定します。実行時に、データ統合サービスは `C_Name` ポートと `Cust_name` ポート間のリンクを作成します。次のマッピング実行でパラメータの値を `CustFirstName` -> `Cust_name` のように変更できます。

リンクポリシーの設定

リンクポリシーはリンクされていないポートを名前でリンクします。マッピングフローによってターゲットカラムを定義する場合、マッピングによって、ソースまたはアップストリームオブジェクトからのすべてのポートがプロパゲートされます。ランタイムリンクポリシーを使用して、特定のタイプのポートまたは特定の名前を持つポートをプロパゲートします。ポート名が同じ場合、リンクポリシーを選択します。同じ名前のポートを自動的にリンクするときは、リンクポリシーを使用します。

ポート名のサフィックスまたはプレフィックスを無視するようにリンクポリシーを設定することが可能です。例えば、サフィックス「_OUT」を無視するようにリンクポリシーを設定した場合、データ統合サービスは、`[SALARY_OUT]` を `[SALARY]` にリンクします。

グループをリンクするために行ったアクションに基づいて、【ランタイムリンク】ダイアログボックスまたは【プロパティ】の【ランタイムリンク】タブにリンクプロパティが表示されます。

- Ctrl を押しながらアップストリームトランスフォーメーションのグループをダウンストリームトランスフォーメーションのグループにドラッグすると、【ランタイムリンク】ダイアログボックスが開きます。
- 動的ポートを作成できない場合、アップストリームトランスフォーメーションからグループをドラッグすれば作成できます。次に、【リンクの作成】ダイアログボックスで【ランタイムリンクの作成】を選択すると、【ランタイムリンク】ダイアログボックスが開きます。

- ダウンストリームトランスフォーメーションの【プロパティ】ビューで【ランタイムリンク】タブを選択します。

ランタイムリンクの例

部門別に合計給与を計算する再利用可能なアグリゲータトランスフォーメーションがマッピングに含まれているとします。アグリゲータトランスフォーメーションは、生成されたポートを持つ式トランスフォーメーションから従業員データを受信します。

式トランスフォーメーションは、動的式から次のような出力ポートを生成します。

Read_EMP ソース 1:

EMPNO_OUT
NAME_OUT
SALARY_OUT
DEPTNO_OUT

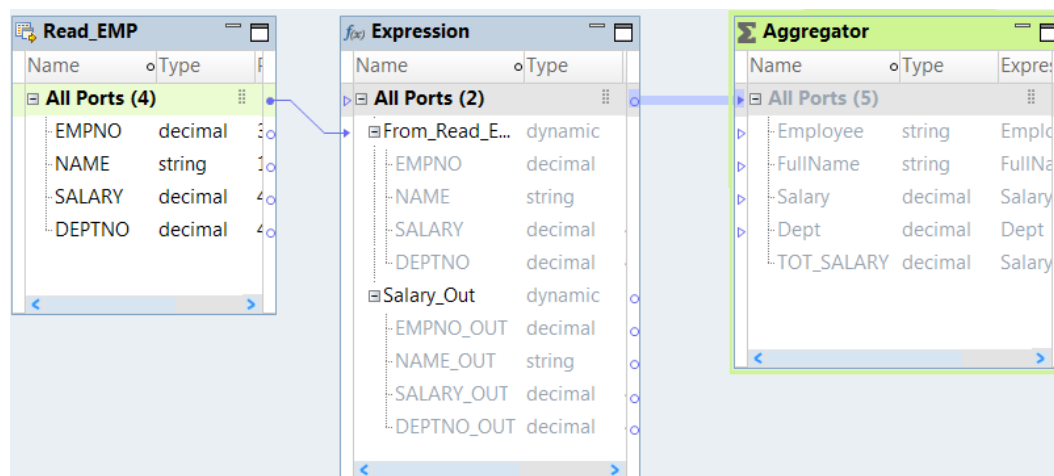
Read_EMP ソース 2:

EMPNUM_OUT
FULLNAME_OUT
SALARY_OUT
DEPT_OUT

アグリゲータトランスフォーメーションインスタンスには、動的ポートはありません。

EMPNO_OUT または EMPNUM_OUT 従業員番号、NAME_OUT または FULLNAME_OUT 文字列、SALARY 番号、および DEPTNO_OUT または DEPT_OUT 部門番号を受信するように、アグリゲータトランスフォーメーションインスタンスのランタイムリンクプロパティを設定します。

次の画像は、式トランスフォーメーションとアグリゲータトランスフォーメーション間のリンクを示しています。



動的マッピングのトラブルシューティング

動的マッピングを設計およびテストする際には、次のトラブルシューティングのヒントを考慮に入れてください。

マッピングの動的ポートに XML データ型のカラムが含まれており、マッピングが失敗した。

以下の条件のいずれかに当てはまる場合、マッピングを使用して XML データをプロパゲートすることはできません。

- データソースからカラムを取得するように読み取りまたは書き込みトランスフォーメーションを設定している。
- データフローからカラムを取得するように書き込みトランスフォーメーションを設定している。
- 実行時にターゲットを作成または置換するようにターゲットのデータオブジェクトを設定している。

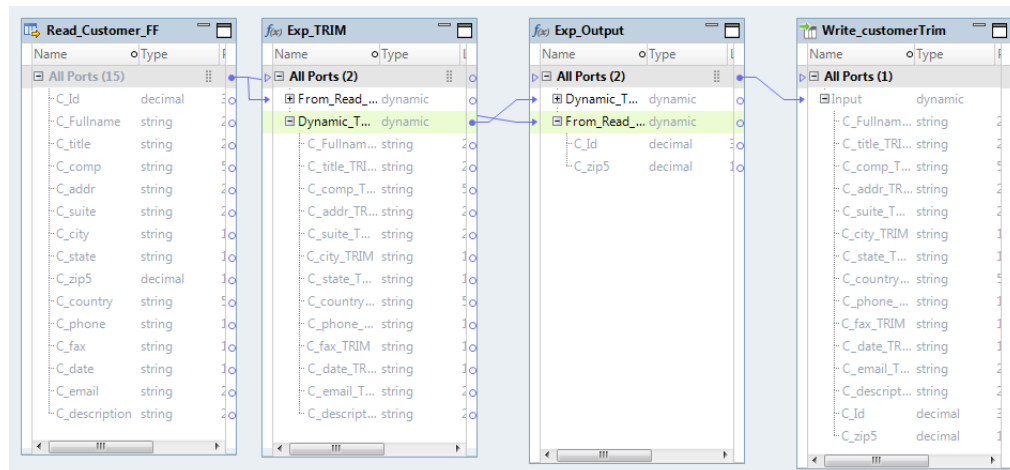
パラメータおよびリンクの実行時エラーで動的マッピングが失敗します。マッピングを実行する前にパラメータとリンクが正常に解決されることを確認する方法を教えてください。

動的マッピングを実行すると、データ統合サービスによってマッピングがコンパイルされ、以下のタスクが実行されます。

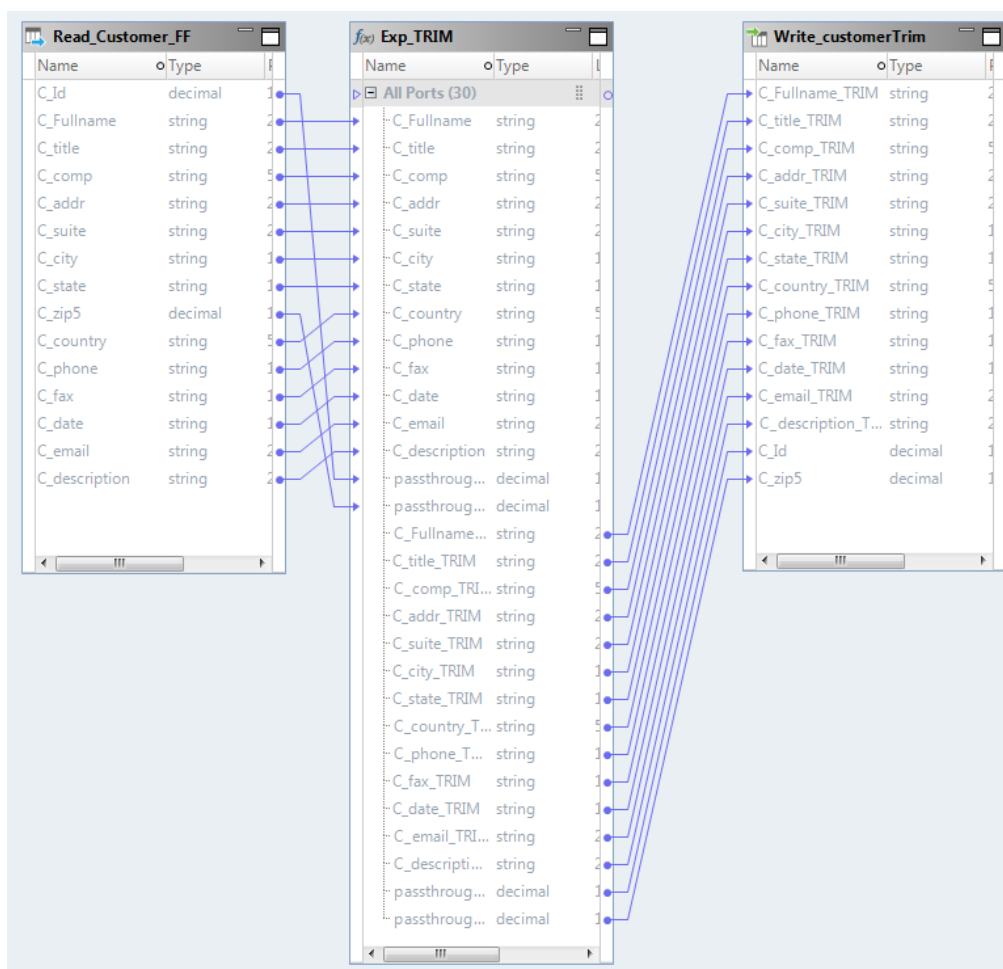
- パラメータ値を解決する。
- 動的ポートを展開して、生成されたポートを静的ポートに変換する。
- 静的ポートをリンクする。
- ランタイムリンクを解決してポートを接続する。

最適化されたマッピングを表示して、コンパイルバージョンのマッピングを確認できます。マッピングエディタの空の領域を右クリックし、**[最適化されたマッピングの表示]** をクリックします。データ統合サービスにより最適化されたマッピングが生成されます。最適化されたマッピングを確認し、問題があれば修正して、マッピングを実行できます。

次の画像は、動的ポートのあるトランスフォーメーションを含む動的マッピングを示しています。



次の画像は、コンパイルバージョンの動的マッピングを示しています。このマッピングでは、生成されたポートを静的ポートに変換し、リンクしています。



第 6 章

動的マッピングを開発して実行する方法

この章では、以下の項目について説明します。

- [動的マッピングの開発と実行, 142 ページ](#)
- [動的ソースの設定, 143 ページ](#)
- [動的ポートの作成, 145 ページ](#)
- [入力ルールを使用した動的ポートの設定, 146 ページ](#)
- [ポートセクタの作成, 150 ページ](#)
- [動的式の作成, 151 ページ](#)
- [動的ターゲットの設定, 153 ページ](#)
- [ランタイムリンクの作成と設定, 158 ページ](#)
- [動的マッピングの検証と実行, 160 ページ](#)

動的マッピングの開発と実行

動的マッピングを作成すると、ソースメタデータの変更の管理や、さまざまなソースおよびターゲットのデータ統合ロジックの再利用ができます。メタデータが変更される可能性のある同じまたは異なるソースとターゲットに対して動的マッピングを実行します。

以下の表に、動的マッピングを作成して実行するタスクの概要を示します。タスクとその実行順序は、マッピングシナリオ、およびマッピングで使用するトランスフォーメーションによって決まります。

タスク	参照
マッピングを作成し、マッピングオブジェクトを追加する。	「マッピングの作成」 (ページ 18) 「マッピングへのオブジェクトの追加」 (ページ 17)
実行時にフラットファイルまたはリレーショナルソースからメタデータの変更を取得するように、読み取りまたはルックアップの各トランスフォーメーションの動的ソースを設定する。	「動的ソースの設定」 (ページ 143)
トランスフォーメーションの動的ポートを作成し、ポートをリンクする。	「動的ポートの作成」 (ページ 145)

タスク	参照
<p>動的ポートの入力ルールを定義し、どの生成されたポートを作成するかを決定する。</p> <ul style="list-style-type: none"> - ポートを含めるまたは除外する入力ルールを定義する。 - 生成されたポートを名前変更する。 - 必要に応じて、生成されたポートを並べ替える。 	「入力ルールを使用した動的ポートの設定」 (ページ 146)
<p>トランスフォーメーションを設定する。</p>	<p>マッピングのトランスフォーメーションを設定するには、『<i>Informatica Developer トランスフォーメーションガイド</i>』で詳細を確認のこと。</p>
<p>必要に応じて、ジョイナ、ルックアップ、または式の各トランスフォーメーションのトランスフォーメーションロジックで使用するポートセレクトタを作成する。</p>	「ポートセレクトタの作成」 (ページ 150)
<p>必要に応じて、式トランスフォーメーションで使用する動的式を作成する。</p>	「動的式の作成」 (ページ 151)
<p>次の方法により、動的ターゲットに書き込むように書き込みトランスフォーメーションを設定する。</p> <ul style="list-style-type: none"> - 関連付けられたデータオブジェクトからカラム定義を定義し、ターゲットファイルからメタデータの変更を取得するか、アップストリームトランスフォーメーションのマッピングフローからカラム定義を定義する。 - 実行時、リレーショナルターゲットを表す書き込みトランスフォーメーションのターゲットテーブルを作成または置換する。 	「動的ターゲットの設定」 (ページ 153)
<p>ランタイムリンクを作成および設定し、実行時にリンクするポートを決定する。</p>	「ランタイムリンクの作成と設定」 (ページ 158)
<p>マッピングでパラメータを使用する場所を決定した後、パラメータを作成して割り当てる。</p> <ul style="list-style-type: none"> - ソースをパラメータとして設定する - ターゲットをパラメータとして設定する - トランスフォーメーションプロパティをパラメータとして設定する 	「パラメータの設定方法」 (ページ 71)
<p>動的マッピングを検証、コンパイル、および実行する。</p>	「動的マッピングの検証と実行」 (ページ 160)

動的ソースの設定

トランスフォーメーションソースの変更時にカラム名を含むメタデータが動的に更新されるように、マッピングの読み取りおよびルックアップの各トランスフォーメーションを設定できます。

動的マッピング用に読み取りおよびルックアップの各トランスフォーメーションを設定するときは、次の方法を1つ以上使用できます。

パラメータをソースとして使用する

読み取りまたはルックアップの各トランスフォーメーションのソースとしてパラメータ値を使用するときは、リポジトリ内の別の場所にある定義済みのソースデータオブジェクトを参照するパラメータ名を選択します。

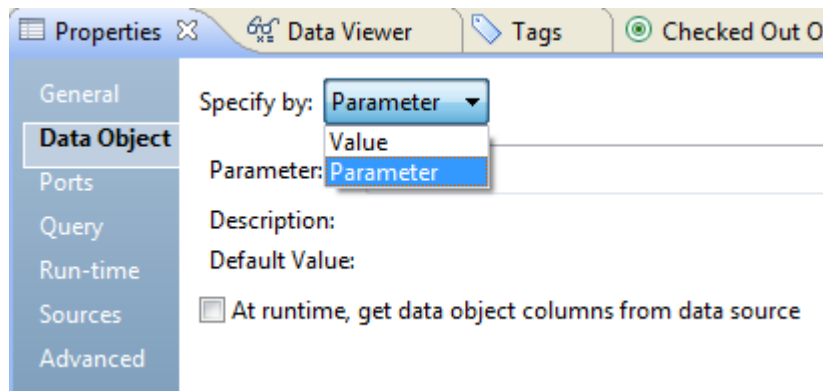
実行時にメタデータを取得するように読み取りおよびルックアップの各トランスフォーメーションを設定する

実行時にソースからデータオブジェクトカラムを取得するように読み取りまたはルックアップの各トランスフォーメーションを設定すると、マッピングの実行時にポート定義が更新されます。

動的マッピングのソースとしてのパラメータの使用

パラメータを動的マッピングソースオブジェクトのソースとして使用できます。

1. マッピングエディタでソースオブジェクトを選択します。
2. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。
3. マッピング実行間で異なる値をソースオブジェクトとして使用するには、**【指定元】** リストで **【パラメータ】** を選択します。



4. パラメータを新規作成する場合は **【新規】** をクリックし、既存のパラメータを選択する場合は **【参照】** をクリックします。

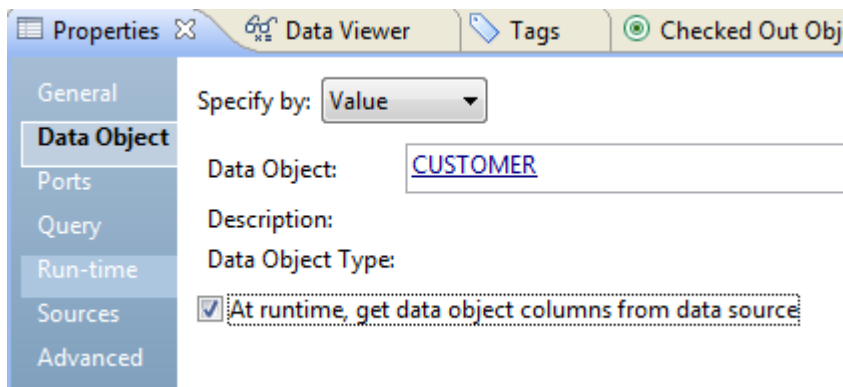
実行時にメタデータの変更を取得するためのソースの設定

実行時にメタデータの変更を取得するように、マッピングのソースオブジェクトのデータソースを設定できます。

マッピングの作成後にデータソースカラムのメタデータが変更された場合は、そのマッピングが期限切れになることがあります。データソースを設定する際、マッピングの実行時にこのデータを取得するオプションを指定できます。

1. マッピングエディタでソースオブジェクトを選択します。
2. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。

3. 実行時にデータソースファイルから動的にカラムを取得するために、[実行時に、データソースからデータオブジェクトのカラムを取得します] を選択します。



動的ポートの作成

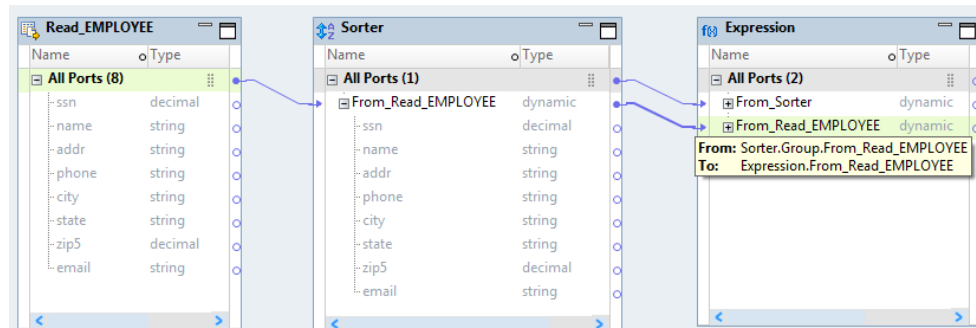
1つの動的ポートを作成して、アップストリームトランスフォーメーションから複数のカラムを受信することができます。これらのカラムは、実行時に変更される場合があります。トランスフォーメーションでは、複数の動的ポートを作成できます。

1. 動的ポートは、次の方法で作成します。

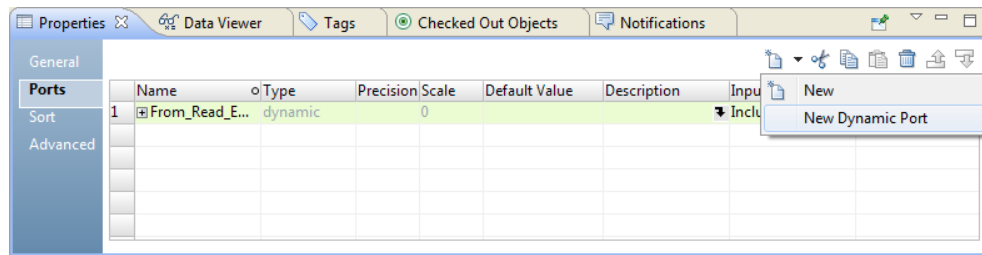
- [すべてのポート] グループまたは動的ポートを別のトランスフォーメーションからドラッグする。

Developer tool は、アップストリームトランスフォーメーションのすべてのカラムについて動的ポートおよび生成されたポートを作成し、これらのポートをリンクします。生成されたポートをフィルタリングするように入力ルールを変更できます。

次の画像は、ソーターおよび式の各トランスフォーメーションの動的ポートを示しています。

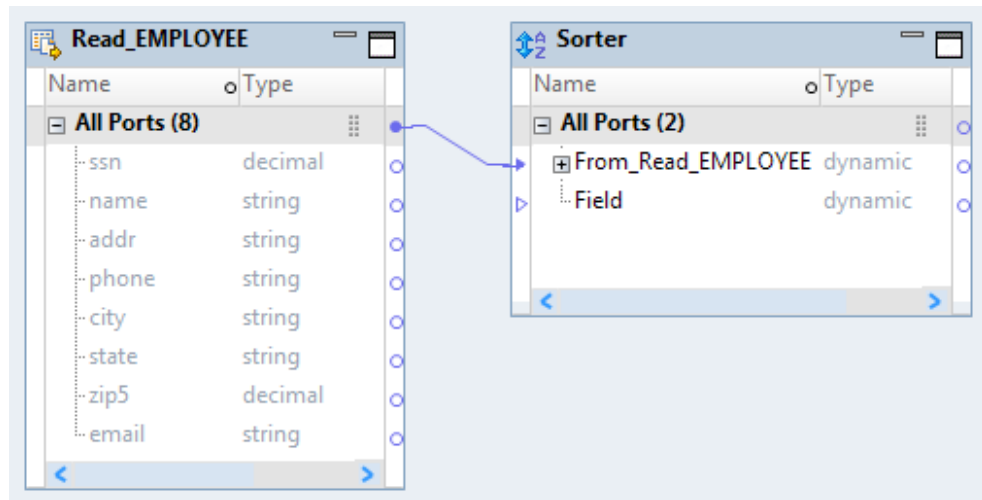


- トランスフォーメーションの【プロパティ】ビューにある【ポート】タブで、【新しい動的ポート】を選択する。



Developer tool は、ユーザーが設定可能な空の動的ポートを作成します。生成されたポートを作成するには、これらのポートを手動でリンクする必要があります。

次の画像は、ソータートランスフォーメーションの新しい動的ポートを示しています。このポートには、生成されたポートはありません。



2. 必要に応じて、この動的ポートの名前を変更し、説明を追加できます。

生成されたポートは、アップストリームトランスフォーメーションからポートのプロパティを継承するため、編集することはできません。

入力ルールを使用した動的ポートの設定

生成するポートを指定し、動的ポートを通じてパイプラインでプロパゲートするには、入力ルールを定義します。

【入力ルール】 ダイアログボックスを使用すると、動的ポートの入力ルールを定義したり、マッピングでのポートの出現場所がわかるように、生成されたポートを名前変更したりすることができるほか、生成されたポートの順序の変更や、ルールの結果の表示も可能になります。ポートを含めたり、除外したりするルールを複数追加することができます。データ統合サービスは、リストの表示順にルールを適用します。

1. 【入力ルール】ダイアログボックスを開きます。

2. トランスフォーメーションの動的ポートごとに 1 つ以上の入力ルールを定義します。入力ルールごとに次の手順を実行します。
 - a. 入力ルールの演算子と選択条件を選択します。
 - b. 選択条件として [名前] を選択した場合は、名前またはパラメータ別に条件の詳細を指定します。
 - c. 選択条件として [タイプ] を選択した場合は、リストからポートのデータ型を選択します。
 - d. 選択条件として [パターン] を選択した場合は、パターンタイプを選択し、値またはパラメータとしてパターン文字列を指定します。必要に応じて、アップストリームトランスフォーメーションからの残りのすべてのポートを含めるように、トランスフォーメーションの最後の動的ポートの入力ルールを定義します。
3. 生成されたポートを名前変更する。
4. 必要に応じて、生成されたポートを並べ替えます。
5. 入力ルールの結果と入力ルールを設定を検証します。

手順 1. [入力ルール] ダイアログボックスを開く

入力ルールを定義または編集するために [入力ルール] ダイアログボックスを開きます。

- ▶ **[入力ルール]** ダイアログボックスは、次の方法で開きます。
 - トランスフォーメーションの動的ポートを右クリックし、**[入力ルールの編集]** を選択する。
 - トランスフォーメーションの **[ポート]** タブで、動的ポートの **[入力ルール]** をクリックする。
- [入力ルール]** ダイアログボックスが表示されたとき、デフォルトでは、[すべてを含める] 入力ルールが選択されています。

手順 2. 入力ルールの定義

アップストリームトランスフォーメーションから動的ポートが受信するポートを含めるまたは除外する入力ルールを定義します。

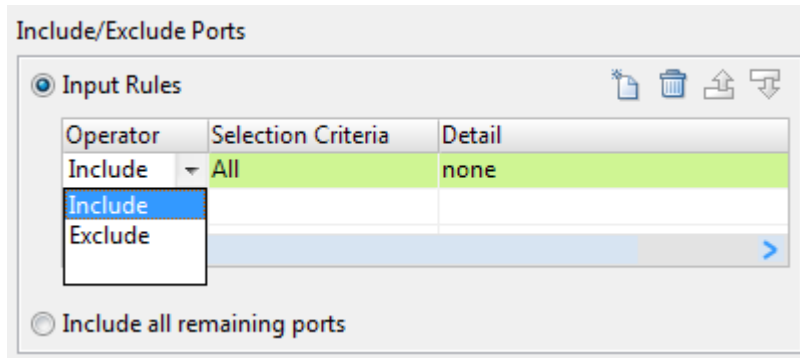
1. **[ポートを含めるまたは除外する]** 領域で **[入力ルール]** を選択します。
2. デフォルトの入力ルールを編集するには、次の手順を実行します。
 - a. 入力ルールの演算子と選択条件を選択します。
 - b. 選択条件の詳細を設定します。
3. 必要に応じて、Developer tool で実行する順序で入力ルールをさらに追加します。
 - a. **[新規]** をクリックし、入力ルールの新しい行を追加します。
 - b. 入力ルールごとに演算子と選択条件を選択し、条件の詳細を指定します。
4. アップストリームトランスフォーメーションからの残りのポートのみを含めるには、次の手順を実行します。
 - a. 別の動的ポートを作成するか、トランスフォーメーションの最後の動的ポートを選択します。
 - b. **[残りのすべてのポートを含める]** を選択します。

このルールでは、アップストリームトランスフォーメーションからのポートのうち、他の動的ポートの一部ではないものが含まれます。

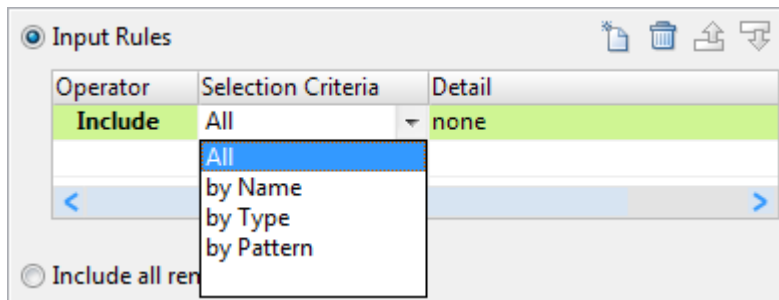
手順 2a. 演算子と選択条件の選択

ポートを含めるまたは除外する演算子、およびポート名またはデータ型に基づいてポートをフィルタリングする選択条件を選択します。

1. **【演算子】** カラムから**【含む】** 演算子または**【除外】** 演算子を選択します。
ルールでポートを含めるのか、除外するのかは、選択した演算子によって決まります。



2. **【選択条件】** カラムでは、次のいずれかのオプションを選択します。
 - **【すべて】**。すべてのポートを含めます。このオプションを選択するときは、**【除外】** 演算子を選択しないでください。
 - **【名前】**。ポート名に基づいてポートを含めるまたは除外します。
 - **【タイプ】**。ポートのデータ型に基づいてポートを含めるまたは除外します。
 - **【パターン】**。ポート名のパターンに基づいてポートを含めるまたは除外します。



3. **【詳細】** カラムで**【詳細】** 矢印をクリックし、選択条件の詳細を指定します。
選択条件に対する**【入力ルールの詳細】** ダイアログボックスが表示されます。

手順 2b. 名前選択条件の詳細の設定

入力ルールの選択条件として**【名前】**を選択した場合は、値のリストからポート名を選択します。あるいは、ポートタイプまたはポートリストタイプのパラメータを使用して、実行時に変更可能なポート名を指定します。

1. **【入力ルールの詳細: 名前別】** ダイアログボックスの**【指定元】** リストから次のいずれかを選択します。
 - **【値】**。ポート名を入力するか、リストからポート名を選択します。
 - **【パラメータ】**。パラメータを新規作成するか、**【ポートリスト】** タイプの既存のパラメータを選択します。
2. 次のいずれかの方法でポート名の値を指定します。
 - **【名前】** ボックスにポート名を入力し、**【追加】** をクリックする。

- **【選択】** をクリックして、**【ポート】** ダイアログボックスでポート名を選択し、**【OK】** をクリックする。
3. ポート名のパラメータを新規作成するには、次の手順を実行します。
 - a. **【新規】** をクリックします。
 - b. **【パラメータ】** ダイアログボックスでパラメータ名を入力します。
 - c. 必要に応じて、パラメータの説明を入力します。
 - d. **【ポート名】** パラメータのデフォルト値を入力します。 **【ポートのリストからポート名を選択する】** をクリックする方法もあります。
 - e. **【OK】** をクリックします。
 4. ポート名の既存のパラメータを選択するには、次の手順を実行します。
 - a. **【参照】** をクリックします。
 - b. **【パラメータの割り当て】** ダイアログボックスでパラメータを選択します。
 - c. 必要に応じて、このダイアログボックスでパラメータを新規作成するか、またはパラメータを編集します。
 - d. **【OK】** をクリックします。

手順 2c. タイプ選択条件の詳細の設定

入力ルールを選択条件として **【タイプ】** を選択した場合は、データ型のリストからタイプを選択します。

1. **【入力ルールの詳細: タイプ別】** ダイアログボックスのリストからデータ型を選択します。
2. **【OK】** をクリックします。

手順 2d. パターン選択条件の詳細の設定

入力ルールを選択条件として **【パターン】** を選択した場合は、ポート名のパターンタイプを選択します。パターンの値を入力するか、または文字列タイプのパラメータを使用して、実行時に変更できる値を指定します。

1. **【入力ルールの詳細: パターン別】** ダイアログボックスの **【パターンタイプ】** リストから次のいずれかを選択します。
 - **プレフィックス**。プレフィックスの文字列で始まるポート名を含めるまたは除外します。
 例えば、プレフィックス値として「E」と入力した場合、入力ルールは、E または e で始まるポート名（EmpNo、empName、EmpTitle など）をフィルタリングします。
 - **サフィックス**。サフィックスの文字列で終わるポート名を含めるまたは除外します。
 例えば、サフィックス値として「E」と入力した場合、入力ルールは、E または e で終わるポート名（empname、EMPTITLE など）をフィルタリングします。
 - **正規表現**。特定のパターンに従うポート名を含めるまたは除外します。
 例えば、値として「E.*No」と入力した場合、入力ルールは、E で始まり No で終わるポート名（ENo、EmpNo、EmployeeNo など）をフィルタリングします。
2. **【指定元】** リストから次のいずれかを選択します。
 - **【値】**。パターンの string 値を入力します。
 - **【パラメータ】**。パラメータを新規作成するか、string 型の既存のパラメータを選択します。
3. **【String】** ボックスにパターン値を指定して、**【OK】** をクリックします。
4. パターンのパラメータを新規作成するには、次の手順を実行します。
 - a. **【新規】** をクリックします。

- b. [パラメータ] ダイアログボックスでパラメータ名を入力します。
 - c. 必要に応じて、パラメータの説明を入力します。
 - d. パターンのデフォルト値を入力し、精度値を入力します。
 - e. **[OK]** をクリックします。
5. ポート名の既存のパラメータを選択するには、次の手順を実行します。
 - a. **[参照]** をクリックします。
 - b. [パラメータの割り当て] ダイアログボックスでパラメータを選択します。
 - c. 必要に応じて、このダイアログボックスでパラメータを新規作成するか、またはパラメータを編集します。
 - d. **[OK]** をクリックします。

手順 3.生成されたポートの名前の変更

生成されたポート名を変更し、トランスフォーメーション内でポート名が一意であることを確認します。

1. **[設定]** 領域で、**[ポートの名前の変更]** を選択します。
2. 生成されたポート名を変更するにあたって、プレフィックスまたはサフィックスを追加するかどうかを選択します。
3. 生成されたポートのプレフィックスまたはサフィックスとしてテキストを追加します。
名前変更されたポートが **[ポートのプレビュー]** 領域に表示されます。

手順 4.生成されたポートの順序変更

結果を効果的に表示して分析できるように、生成されたポートの順序を変更します。

- ▶ **[設定]** 領域で、**[生成されたポートの順序を入力ルールの順序に従って変更]** を選択します。
順序変更されたポートが **[ポートのプレビュー]** 領域に表示されます。 アップストリームトランスフォーメーションに表示される順序ではなく、入力ルールの順序に従って、生成されたポートが表示されます。

手順 5.動的ポート設定の検証

動的ポートに対して定義したルールと設定に基づいて、生成されたポートを表示します。

- ▶ **[入力ルール]** ダイアログボックスの **[ポートのプレビュー]** 領域で、動的ポートの入力ルール設定の結果を確認します。

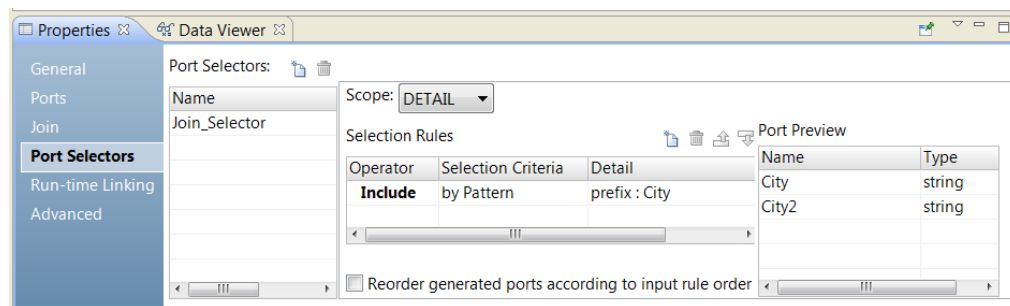
ポートセレクトタの作成

動的式、ルックアップ条件、または結合条件で使用するポートを決定するために、ポートセレクトタを作成します。

1. **[ポートセレクトタ]** タブをクリックします。
2. **[ポートセレクトタ]** 領域で、**[新規]** をクリックします。
Developer tool が、すべてのポートが含まれるデフォルトの選択ルールを使用してポートセレクトタを作成します。

3. **【ポートセレクトア】** 領域で、ポートセレクトア名を一意的な名前に変更します。
4. ジョイントランスフォーメーションまたはルックアップトランスフォーメーションに対して作業している場合は、スコープを選択します。
使用可能なポートは、選択したポートのグループに基づいて変更されます。
5. **【選択ルール】** 領域で、**【演算子】** を選択します。
 - 含む。ポートセレクトアのポートを含めるルールを作成します。ポートを除外する前にポートを含める必要があります。
 - 除外。ポートセレクトアから特定のポートを除外するルールを作成します。
6. **【選択条件】** を選択します。
 - 名前。特定のポートを名前を選択します。スコープ内のポートのリストからポート名を選択できます。
 - タイプ。ポートをタイプで選択します。1 つまたは複数のデータ型を選択できます。
 - パターン。ポート名の文字のパターンでポートを選択します。特定の文字で検索するか、正規表現を作成できます。

次の図は、**【ポートセレクトア】** タブを示しています。



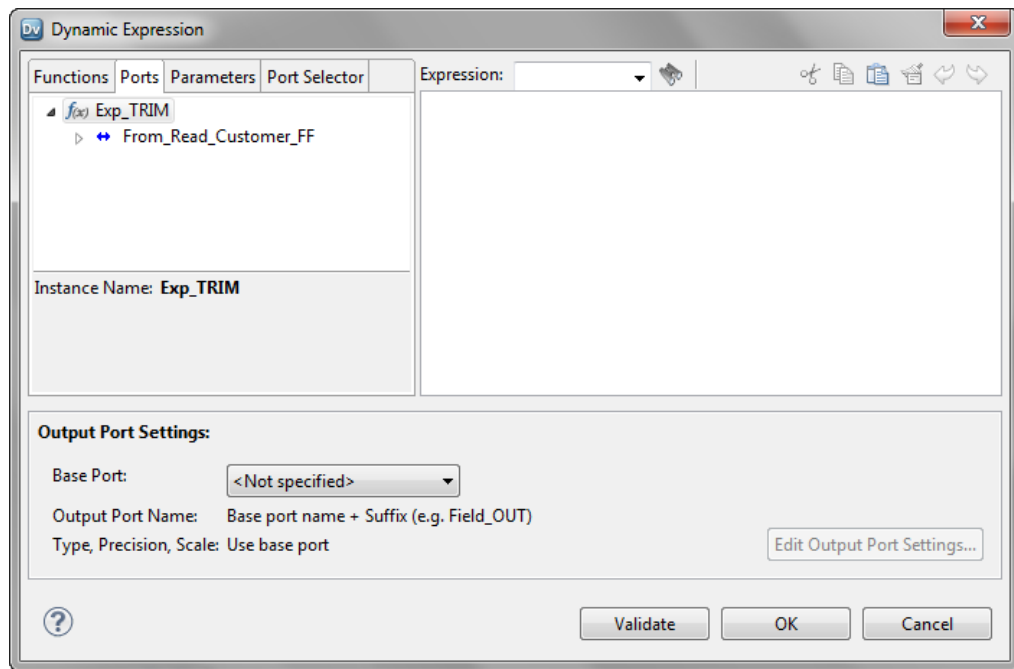
7. **【詳細】** カラムをクリックします。
【入力ルールの詳細】 ダイアログボックスが表示されます。
8. ポートをフィルタ処理する基準となる値を選択します。
 - 名前。値またはパラメータを基準としてポートリストを作成する場合に選択します。**【選択】** をクリックして、リスト内のポートを選択します。
 - タイプ。リストから 1 つ以上のデータ型を選択します。**【ポートのプレビュー】** 領域に、選択したタイプのポートが表示されます。
 - パターン。ポート名のプレフィックスまたはサフィックスから、特定の文字パターンを検索する場合に選択します。または、検索に使用する正規表現を作成することを選択します。パラメータを設定するか、検索に使用するパターンを設定します。
- 【ポートのプレビュー】** 領域に、設定したルールどおりにポートセレクトアのポートが表示されます。
9. ポートセレクトアのポートの順序を変更するには、**【生成されたポートの順序を入力ルールの順序に従って変更】** を選択します。

動的式の作成

式トランスフォーメーションで動的式を作成して、動的ポートまたはポートセレクトアの各ポートで式を 1 回実行します。動的式は、インスタンスごとに個別に生成されたポートに結果を返します。

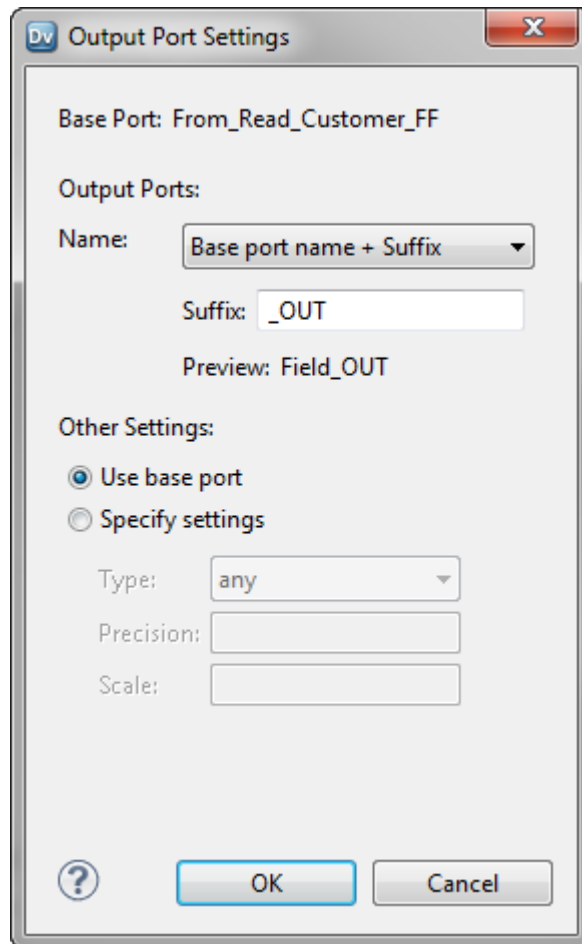
1. 式トランスフォーメーションで **【プロパティ】** ビューに移動して、**【ポート】** タブをクリックします。

2. **【新しい動的ポート】** をクリックします。
Developer tool で、デフォルトのプロパティを使用して動的ポートが作成されます。
3. 動的ポートの名前を変更し、入力オプションを無効にします。
動的ポートは出力ポートである必要があります。
4. 動的出力ポートの **【式】** カラムで、**【開く】** ボタン (🔗) をクリックします。
【動的式】 ダイアログボックスが表示されます。



5. 式エディタに式を入力します。式にはポートセクタまたは動的ポートを含めることができます。
例えば、LTRIM(RTRIM(Dynamic_Customer))です。ここで、Dynamic_Customer は動的ポートです。
6. **【検証】** をクリックして式を検証します。
7. **【OK】** をクリックして、**【式の検証】** ダイアログボックスを終了します。
8. **【出力ポート設定】** 領域で、**【ベースポート】** リストから動的出力ポートを選択するか、式で参照したポートセクタを選択します。
Developer tool で、選択に基づいて出力ポートが生成されます。

9. 次の手順を実行して出力ポートの名前を変更します。
 - a. **【出力ポート設定の編集】** をクリックします。
【出力ポート設定】 ダイアログボックスが表示されます。



- b. **【名前】** リストで、オプションのいずれかを選択してプレフィックスまたはサフィックスの値を入力します。 **【固定文字列 + 自動番号割り当て】** を選択した場合、出力ポート名のテキストを入力します。例えば、出力ポート名に TRIM と入力すると、出力ポート名は TRIM1、TRIM2、TRIM3 と表示されます。
 - c. 必要に応じて、**【その他の設定】** 領域の **【設定の指定】** を選択し、出力ポートのタイプ、精度、およびスケールを変更します。デフォルトでは、出力ポートでベースポートの設定が使用されます。
 - d. **【OK】** をクリックします。
10. **【OK】** をクリックして、**【動的式】** エディタを終了します。

動的ターゲットの設定

ターゲットメタデータが変更された場合、実行時にターゲットからカラムを受信するように書き込みトランスフォーメーションを設定します。必要に応じて、パラメータをターゲットデータオブジェクトとして指定し、

異なる値の割り当てができるようにします。また、関連付けられたオブジェクトまたはマッピングフローのどちらを使用して書き込みトランスフォーメーションがポートを定義するかを指定することもできます。

動的マッピングの書き込みトランスフォーメーションを設定する場合、次の1つ以上の方法を使用できます。

パラメータをターゲットとして使用する

パラメータをターゲットの基になるデータオブジェクトとして指定すると、パラメータを通して書き込みトランスフォーメーションのスキーマを変更できます。

実行時にターゲットからデータオブジェクトのカラムを取得する

実行時にターゲットからデータオブジェクトのカラムを取得するオプションを有効にすると、書き込みトランスフォーメーションのポートがターゲットスキーマの変更内容で動的に更新されます。

DDL クエリを定義して実行時にターゲットを作成または置換する

実行時にターゲットを作成または置換する場合、DDL クエリを定義して、定義したクエリに基づいてターゲットを作成することができます。DDL クエリは、リレーショナルターゲットおよび Hive ターゲットに対して定義できます。

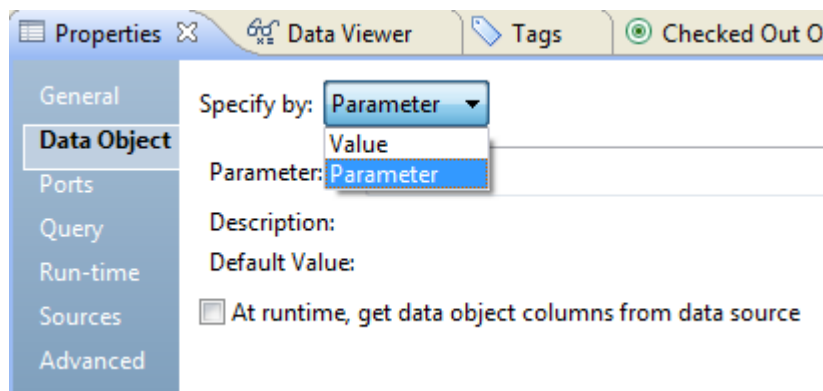
書き込みトランスフォーメーションのポートをマッピングフローから定義する

マッピングフローからポートを定義するように選択した場合、データ統合サービスは、アップストリームのカラムの定義に基づいて書き込みトランスフォーメーションのポートを定義します。ターゲットのカラムが実行時に動的に更新されます。

動的マッピングのターゲットとしてのパラメータの使用

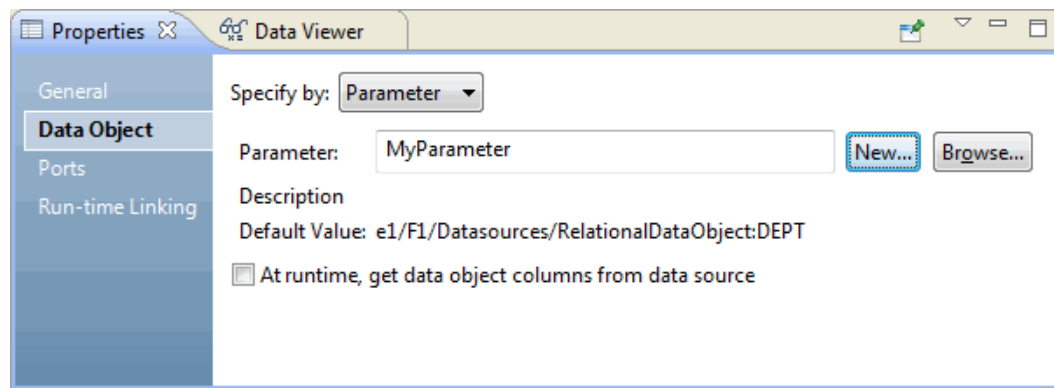
パラメータをトランスフォーメーションのデータオブジェクトとして使用し、パラメータを実行時に変更することができます。

1. マッピングエディタで書き込みトランスフォーメーションを選択します。
2. [プロパティ] ビューで、[データオブジェクト] タブをクリックします。
3. [指定元:] ドロップダウンリストで、[パラメータ] を選択します。



4. 次のいずれかのオプションを選択します。
 - [新規] をクリックしてパラメータを作成します。パラメータに名前を付けて、パラメータのデフォルト値を参照して選択します。
 - [参照] をクリックして、既存のパラメータを選択します。

次の画像は、データソースとしてのパラメータを持つトランスフォーメーションを示しています。

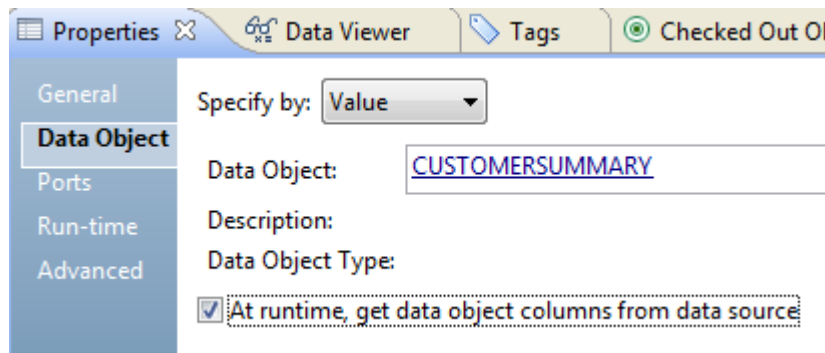


実行時のデータソースからのターゲットオブジェクトカラムの取得

実行時にデータソースからデータオブジェクトのカラムを取得するオプションを有効にすることができます。

実行時にデータソースからデータオブジェクトのカラムを取得するオプションを選択した場合、マッピングの実行時にマッピングはデータオブジェクトのカラムをトランスフォーメーションに取り込みます。データソースのカラムおよびメタデータが変更されると、マッピングは変更された情報を取得します。

1. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。
2. **【実行時に、データソースからデータオブジェクトのカラムを取得します】** を選択します。



実行時にターゲットを作成または置換するための DDL クエリの定義

実行時にターゲットを作成または置換する場合、DDL クエリを定義することができます。データ統合サービスは、この DDL クエリに基づいて実行時にターゲットテーブルを作成または置換します。DDL クエリは、リレーショナルターゲットおよび Hive ターゲットに対して定義できます。DDL クエリにはプレースホルダおよびパラメータを入力できます。

1. **【プロパティ】** ビューで **【詳細】** タブをクリックします。
2. **【実行時にテーブルを作成または置換】** オプションを選択します。
【DDL クエリ】 フィールドが入力可能になります。
3. **【編集】** をクリックします。
【DDL クエリ】 ダイアログボックスが表示されます。
4. エディタで DDL クエリを入力します。

DDL クエリにはプレースホルダを入力できます。データ統合サービスにより、実行時にプレースホルダが実際の値に置き換えられます。例えば、テーブルに 50 個のカラムが含まれている場合、DDL クエリにすべてのカラム名を入力する代わりに、プレースホルダを入力できます。

DDL クエリには、次のプレースホルダを入力できます。

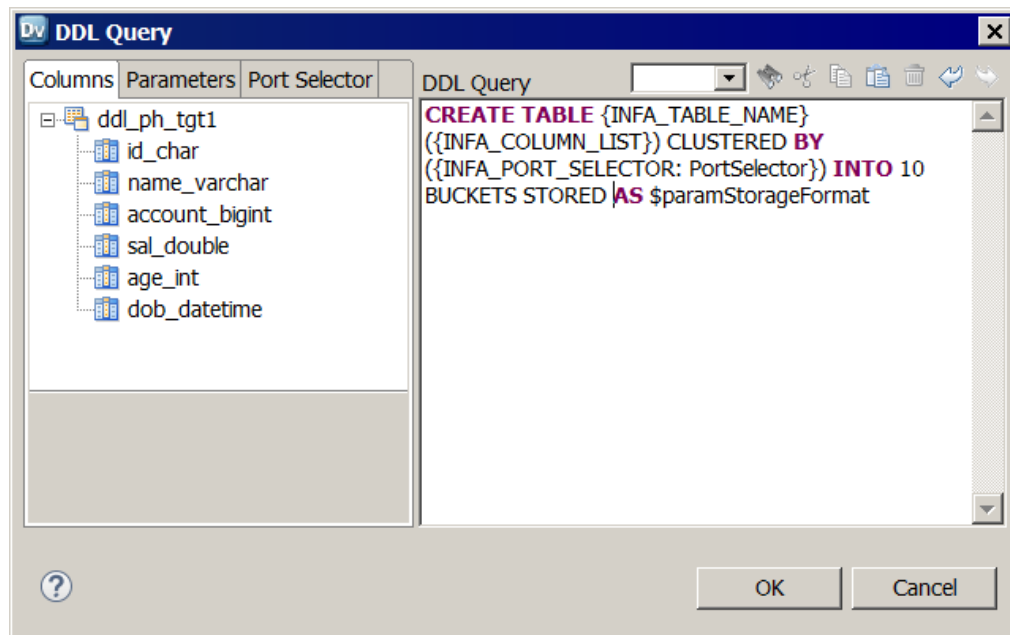
- INFA_TABLE_NAME. 実行時にターゲットテーブル名を取得します。
- INFA_COLUMN_LIST. 実行時にターゲットテーブルのカラムのリストを取得します。
- INFA_PORT_SELECTOR. ポートセレクトを追加します。

注: プレースホルダ名では、大文字と小文字が区別されます。プレースホルダは 2 つの中括弧で囲む必要があります。例えば、{INFA_TABLE_NAME} のようになります。

次の手順を実行して DDL クエリを定義することもできます。

- カラム名を追加するには、**[カラム]** タブでカラムをダブルクリックします。
- パラメータを定義するには、**[パラメータ]** タブをクリックし、パラメータ名をダブルクリックします。**[パラメータの管理]** をクリックしてパラメータを追加、編集、または削除することもできます。
- ポートセレクトを設定するには、**[ポートセレクト]** タブをクリックし、ポートセレクトをダブルクリックします。**[新規]** をクリックして新しいポートセレクトを設定することもできます。

次の図に、Hive ターゲットテーブルを作成するための DDL クエリを示します。



この図の DDL クエリには、INFA_TABLE_NAME、INFA_COLUMN_LIST、および INFA_PORT_SELECTOR というプレースホルダがあります。保存形式を定義するためのパラメータも含まれています。

DDL クエリを入力しない場合、データ統合サービスはマッピングフローまたはデータオブジェクトに基づいてターゲットを作成します。

5. **[OK]** をクリックして DDL クエリを保存します。

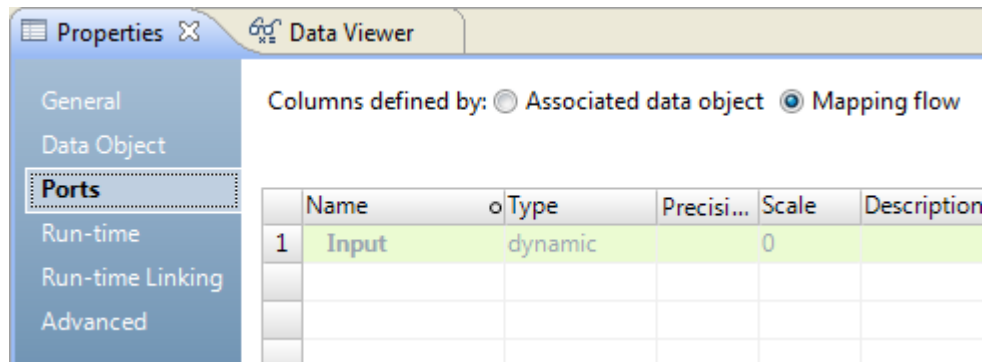
書き込みトランスフォーメーションポートの定義

マッピングフローでターゲットオブジェクトのカラムを定義することで、アップストリームのマッピングオブジェクトによって書き込みトランスフォーメーションの入力ポートを更新できます。

1. **[プロパティ]** ビューの **[ポート]** タブをクリックします。

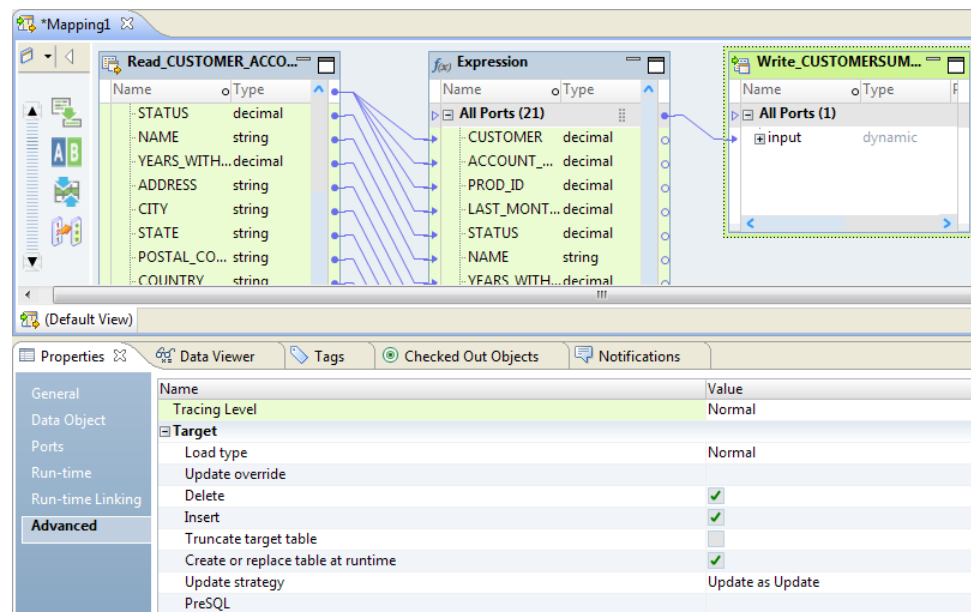
2. [次によって定義されたカラム:] で [マッピングフロー] を選択します。

次の画像は、[ポート] タブを示しており、関連付けられたデータオブジェクトで定義されたポートが表示されています。



3. 動的ポートおよびターゲットを有効にします。
 - a. 書き込みトランスフォーメーションの [入力] ペインにアップストリームのポートをドラッグします。
ターゲットはカラムの定義をアップストリームのマッピングオブジェクトから取得します。
 - b. [プロパティ] ビューで [詳細] タブをクリックします。
 - c. [実行時にテーブルを作成または置換] を選択します。

次の画像は、ターゲットオブジェクトの [詳細] タブにある [実行時にテーブルを作成または置換] オプションを示しています。



実行時にデータ統合サービスはターゲットテーブルを作成するか、ターゲットテーブルを削除して置換します。

注: マッピングに複数のターゲットが含まれており、それらのカラムが同じ物理データオブジェクトによって定義されている場合、[実行時にテーブルを作成または置換] オプションを1つのターゲットに対してのみ有効にします。複数のターゲットに対してこのオプションを有効にした場合、マッピングで作成されたテーブルのメタデータと一致するターゲットが1つのみとなるため、マッピングが失敗します。

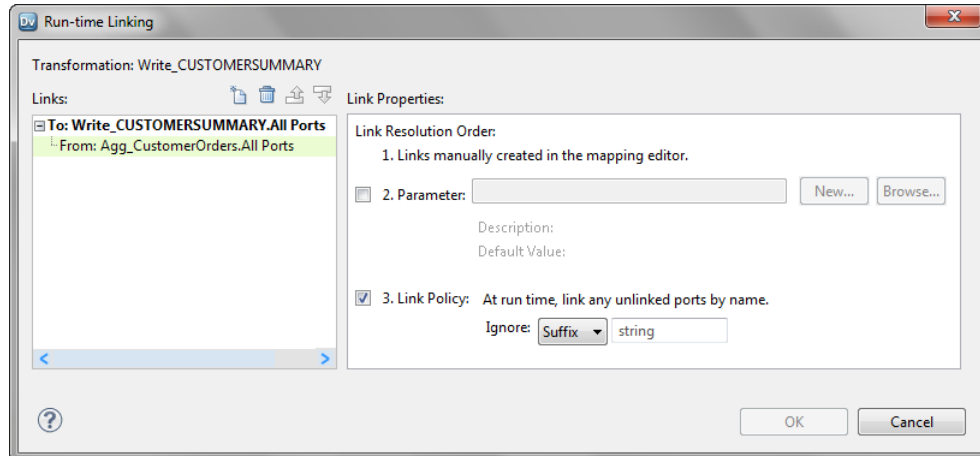
ランタイムリンクの作成と設定

トランスフォーメーショングループ間のランタイムリンクを作成して、パラメータまたはリンクポリシーあるいはその両方に基づいてポートを実行時にリンクします。

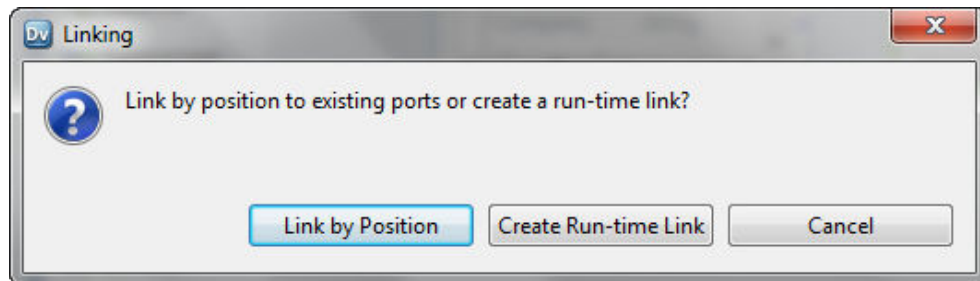
1. 以下の方法でランタイムリンクを作成します。

- Ctrl キーを押しながら、グループを動的マッピングのダウストリームトランスフォーメーションにドラッグします。

【ランタイムリンク】 ダイアログボックスが表示されます。




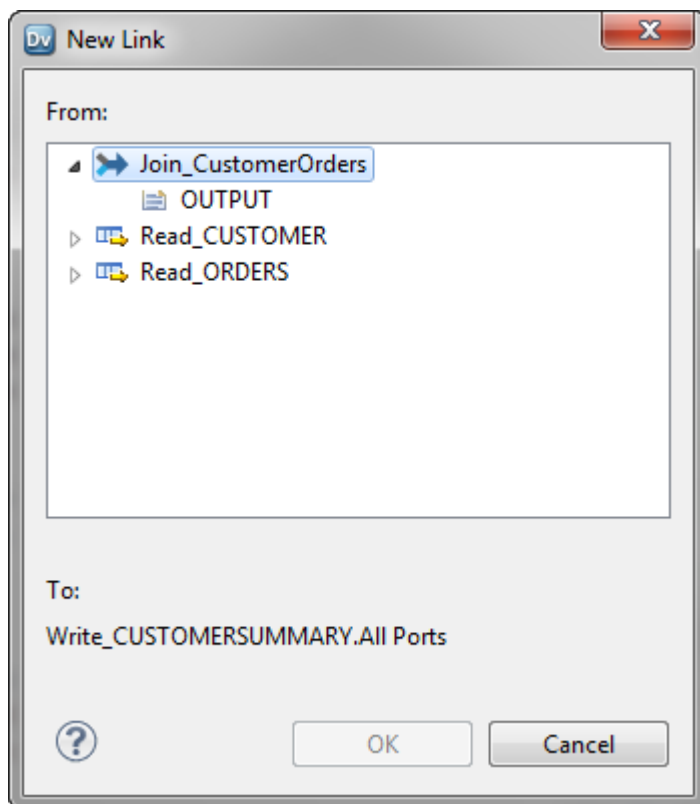
- 書き込みトランスフォーメーションまたは再利用可能なトランスフォーメーションへのランタイムリンクを作成するには、グループをアップストリームトランスフォーメーションから、再利用可能なトランスフォーメーションまたは書き込みトランスフォーメーション内のグループにドラッグします。次に、**【リンクの作成】** ダイアログボックスで **【ランタイムリンクの作成】** を選択すると、**【ランタイムリンク】** ダイアログボックスが開きます。
- 作成するランタイムリンクのリンク先とするダウストリームトランスフォーメーションで、**【プロパティ】** ビューに移動して、**【ランタイムリンク】** タブをクリックします。



2. **【リンクのプロパティ】** 領域で、次のオプションの 1 つまたは両方を選択して、実行時にどのポートをリンクするかを特定します。

- **【パラメータ】**。マッピングの実行間でポート名が変わる可能性があり、そのポート名を認識している場合、パラメータを使用します。新しいパラメータを作成するか、入力リンクセットタイプの既存のパラメータを選択します。
- **リンクポリシー**。名前によってポートを自動的にリンクするには、リンクポリシーを使用します。このオプションはデフォルトで選択されています。ポート名にプレフィックスまたはサフィックスが含まれる場合、無視する文字列を入力します。

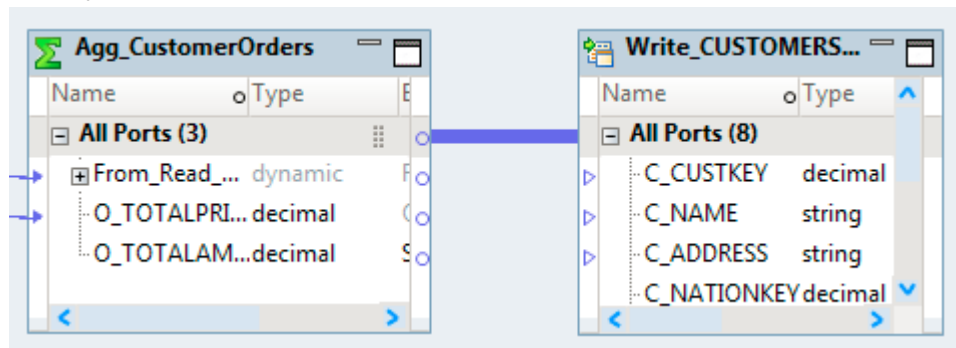
3. [入力リンクセット] タイプの新しいパラメータを作成するには、次の手順を実行します。
 - a. **【新規】** をクリックします。
 - b. **【パラメータ】** ダイアログボックスでパラメータ名を入力します。
例: Cust_InputLinkSet
 - c. 必要に応じて、パラメータの説明を入力します。
 - d. パラメータのデフォルト値を、カンマ区切りのポートのペアとして入力します。
例えば、次のようにデフォルト値を入力します。
C_NAME->Cust_name, C_ACCTBAL->Cust_acctbal
 - e. **【OK】** をクリックします。
4. [入力リンクセット] タイプの既存のパラメータを選択するには、次の手順を実行します。
 - a. **【参照】** をクリックします。
 - b. **【パラメータの割り当て】** ダイアログボックスでパラメータを選択します。
 - c. 必要に応じて、このダイアログボックスでパラメータを新規作成するか、またはパラメータを編集します。
 - d. **【OK】** をクリックします。
5. 必要に応じて、**【ランタイムリンク】** ダイアログボックスから別のランタイムリンクを追加するには、次の手順を実行します。
 - a. **【新規】** ボタン () (**【リンク】** 領域内) をクリックします。
【新しいリンク】 ダイアログボックスが表示されます。



- b. 動的マッピング内の別のトランスフォーメーションからグループを選択します。

6. **[OK]** をクリックして、ランタイムリンクを作成します。

Developer tool によって、グループ間にランタイムリンクが作成されます。



7. 既存のランタイムリンクを編集するには、そのリンクを右クリックして、**[ランタイムリンク]** を選択します。

[ランタイムリンク] ダイアログボックスが表示され、そこでリンク先のポートを決定するオプションを変更できます。

ランタイムリンクの作成

ランタイムリンクを作成するには、**[ランタイムリンク]** ダイアログボックスを開くか、**[プロパティ]** ビューの **[ランタイムリンク]** タブを使用します。???

- ▶ 以下の方法でランタイムリンクを作成します。

- 1. **[すべてのポート]** グループを動的マッピングのダウンストリームトランスフォーメーションにドラッグします。

[リンクの作成] ダイアログボックスが表示されます。

- 2. **[ランタイムリンクの作成]** をクリックします。

- Ctrl キーを押しながら、**[すべてのポート]** グループを動的マッピングのダウンストリームトランスフォーメーションにドラッグします。

[ランタイムリンク] ダイアログボックスが表示されます。

- 1. ランタイムリンクを作成するダウンストリームトランスフォーメーションを選択します。
- 2. **[プロパティ]** ビューで、

動的マッピングの検証と実行

マッピングを検証して、データ統合サービスがマッピング全体を読み取って処理できることを確認してから、マッピングを実行して変換されたデータをターゲットに書き込みます。

- 1. マッピングを開いて、**[編集]** > **[検証]** をクリックします。

[検証ログ] ビューにエラーが表示される場合は、エラーを修正し、再度マッピングを検証します。

- 2. マッピングが有効であれば、**[ファイル]** > **[保存]** をクリックしてマッピングを保存します。

- 3. **[実行]** > **[マッピングの実行]** をクリックします。

[マッピングの実行] ウィンドウに、マッピング実行の進行状況が表示されます。マッピングが実行され、出力がターゲットファイルに書き込まれます。

4. **【ウィンドウ】 > 【ビューの表示】 > 【進行状況】** をクリックして、マッピング実行の進行状況を表示します。

【進行状況】 ビューが開きます。

5. マッピング実行と次のマッピング実行の間でパラメータ値を変更します。
6. ソーススキーマが変更された後、またはパラメータ値を変更した後に、マッピングを検証して再実行します。

第 7 章

動的マッピングの使用例

この章では、以下の項目について説明します。

- [使用例: リレーショナルソースのメタデータ変更のための動的マッピング, 162 ページ](#)
- [使用例: さまざまなソースおよびターゲットの動的マッピングの再利用, 173 ページ](#)

使用例: リレーショナルソースのメタデータ変更のための動的マッピング

顧客注文の合計額を集計する必要がある組織の開発者という立場を想定してみましょう。その組織では、顧客データと顧客注文データを 2 つのテーブルとして異なる部門から毎週受信しています。各部門では頻繁に、カラムの順序を変更したり、テーブルに新規のカラムを追加したりします。開発者は、ソーススキーマの変更に対応して顧客注文の合計額を集計できる動的マッピングを開発する必要があります。

ソーステーブル

CUSTOMER と ORDERS は、マッピング内の読み取りトランスフォーメーションのソーステーブルです。

次の表に、C_CUSTKEY カラムがプライマリキーとして設定された CUSTOMER テーブルの各カラムとメタデータを示します。

名前	ネイティブタイプ	精度	スケール
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0

次の表に、ORDERS テーブルの各カラムとメタデータを示します。

名前	ネイティブタイプ	精度	スケール
O_ORDERKEY	number(p,s)	38	0
O_CUSTKEY	number(p,s)	38	0
O_ORDERSTATUS	varchar2	1	0
O_TOTALPRICE	number(p,s)	10	2
O_ORDERDATE	date	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIOIRITY	number(p,s)	30	0

ターゲットテーブル

CUSTOMERSUMMARY は、マッピング内の書き込みトランスフォーメーションのターゲットテーブルです。

次の表に、CUSTOMERSUMMARY テーブルの各カラムとメタデータを示します。

名前	ネイティブタイプ	精度	スケール
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_TOTALAMOUNT	number(p,s)	10	2

動的マッピング

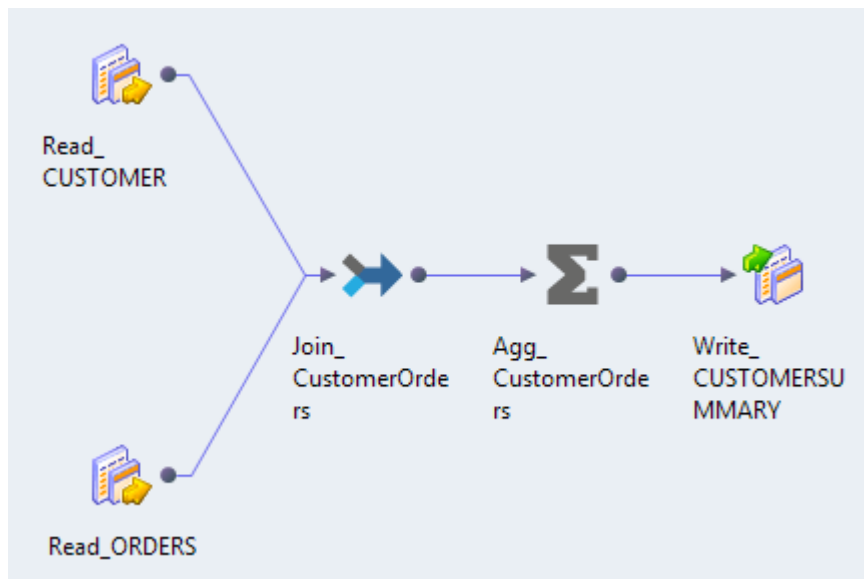
マッピング m_CustomerLoad を作成して、次の動的マッピング機能を設定します。

- 動的ソースから読み取ることができる読み取りトランスフォーメーション
- 新しいカラムおよび変更されたカラムを渡すことができる、ダウンストリームトランスフォーメーションの動的ポート
- 動的ターゲットに書き込むことができる書き込みトランスフォーメーション
- ポートを実行時に書き込みトランスフォーメーションに接続できるランタイムリンク

マッピングを実行すると、データ統合サービスによって以下のタスクが実行されます。

1. データオブジェクトの構造とソースファイルにおけるメタデータの変更を取得します。
2. 新しいカラムおよび変更されたカラムを動的ポート経由で各トランスフォーメーションに渡します。
3. 新しいポートおよび変更されたポートを書き込みトランスフォーメーションに接続します。
4. 変換されたデータをターゲットに書き込みます。

次の図は、マッピング内のオブジェクトを示しています。



マッピングには次のオブジェクトが含まれます。

Read_CUSTOMER

リレーショナルソース CUSTOMER を表す読み取りトランスフォーメーション。このリレーショナルテーブルには、顧客ごとに 1 行が存在しています。

Read_ORDERS

リレーショナルソース ORDERS を表す読み取りトランスフォーメーション。このリレーショナルテーブルには、顧客注文ごとにそれぞれ 1 行が存在しています。

Join_CustomerOrders

CUSTOMER ソースと ORDERS ソースを結合するジョイナトランスフォーメーション。

Agg_CustomerOrders

顧客注文合計を集計するアグリゲータトランスフォーメーション。

Write_CUSTOMERSUMMARY

リレーショナルターゲット CUSTOMERSUMMARY を表す書き込みトランスフォーメーション。このリレーショナルテーブルには、顧客別の注文合計の集計値を書き込むマッピング用のカラムが含まれています。

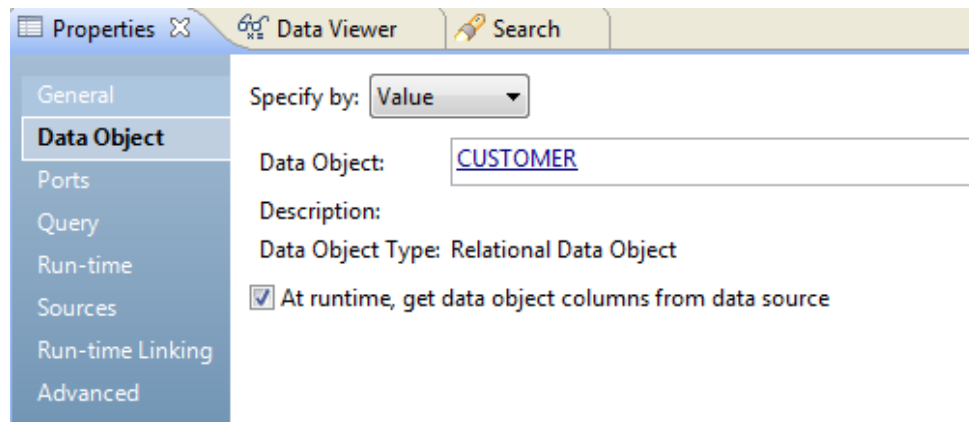
手順 1. 読み取りトランスフォーメーションの設定

実行時に動的ソースから直接、カラムおよびメタデータの変更を取得するように、読み取りトランスフォーメーションを設定します。

1. CUSTOMER および ORDERS リレーショナルデータオブジェクトを表す 2 つの読み取りトランスフォーメーションを追加します。

2. 実行時にソースから直接、カラムおよびメタデータの変更を取得するように、Read_CUSTOMER トランスフォーメーションを設定します。
 - a. Read_CUSTOMER トランスフォーメーションを選択します。
 - b. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。
 - c. **【実行時に、データソースからデータオブジェクトのカラムを取得します】** を選択します。

次の画像は、Read_CUSTOMER トランスフォーメーションの **【データオブジェクト】** タブの設定を示しています。



3. 実行時にソースから直接、カラムおよびメタデータの変更を取得するように、Read_ORDERS トランスフォーメーションを設定します。
 - a. Read_ORDERS トランスフォーメーションを選択します。
 - b. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。
 - c. **【実行時に、データソースからデータオブジェクトのカラムを取得します】** を選択します。

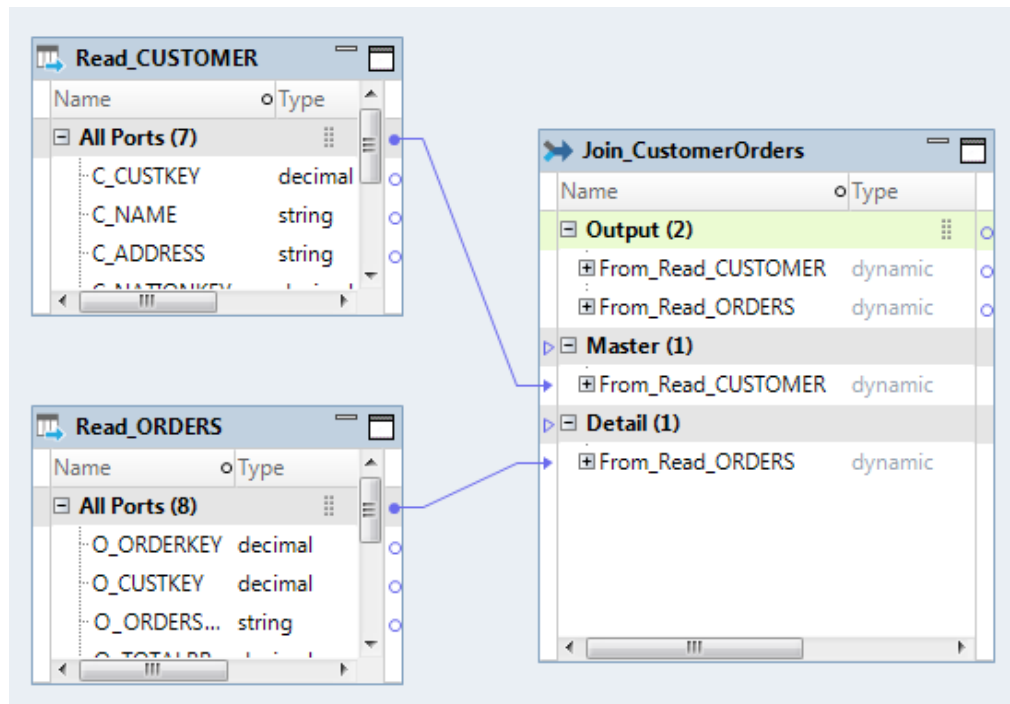
手順 2. ジョイナトランスフォーメーションの設定

マッピングにジョイナトランスフォーメーションを追加して、読み取りトランスフォーメーションから新しいカラムおよび変更されたカラムを受信するように、動的ポートを設定します。2つのソーステーブル CUSTOMER と ORDERS を結合するための結合条件を定義します。

1. ジョイナトランスフォーメーション Join_CustomerOrders をマッピングに追加します。
2. 以下の手順で、ジョイナトランスフォーメーションに動的ポートを作成します。
 - a. Read_Customer トランスフォーメーションの **【すべてのポート】** グループを、ジョイナトランスフォーメーションの **【マスタ】** グループにドラッグします。
Developer tool は、マスタグループと出力グループに動的ポート From_Read_CUSTOMER を作成します。
 - b. Read_Orders トランスフォーメーションの **【すべてのポート】** グループを、ジョイナトランスフォーメーションの **【詳細】** グループにドラッグします。
Developer tool は、詳細グループと出力グループに動的ポート From_Read_ORDERS を作成します。

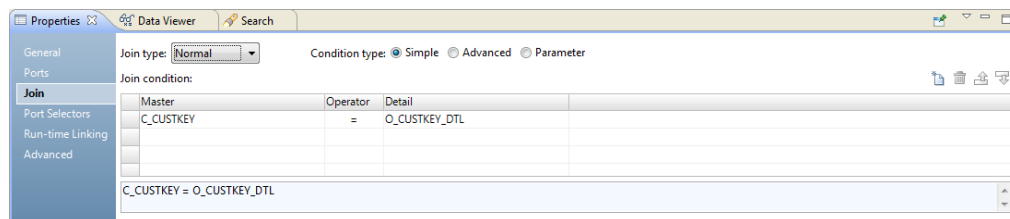
動的ポートには、対応する読み取りトランスフォーメーションのすべてのポートが生成されたポートとして含まれます。

次の図では、読み取りトランスフォーメーションの「すべてのポート」グループが、ジョイナトランスフォーメーション内の2つの動的ポートにリンクされています。



3. **【プロパティ】** ビューで、**【結合】** タブをクリックします。
4. **【新規】** ボタンをクリックし、結合条件として $C_CUSTKEY = O_CUSTKEY_DTL$ を定義します。

次の図は、結合条件が定義された**【結合】** タブを示しています。

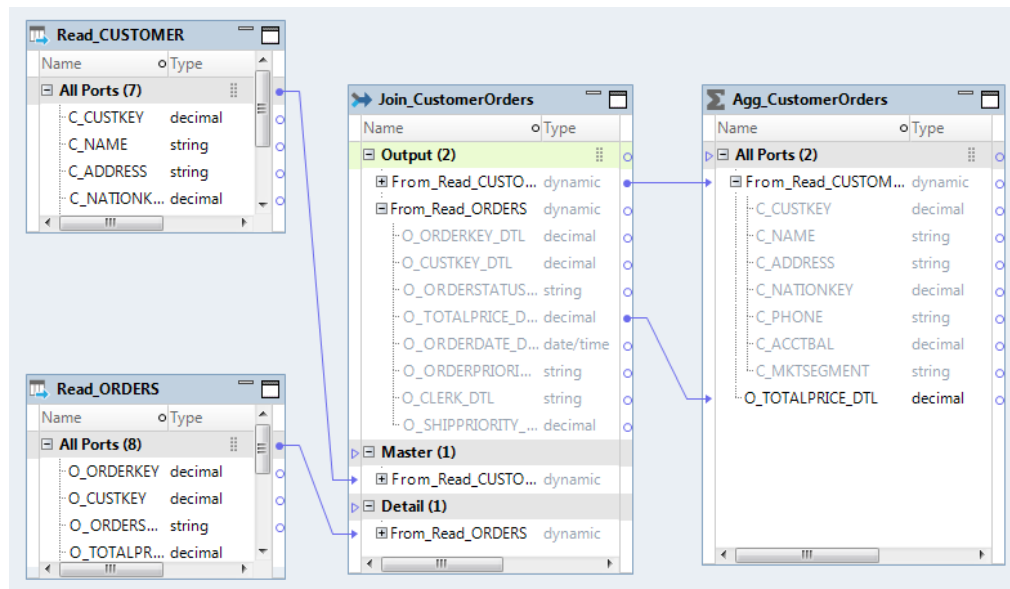


手順 3. アグリゲータトランスフォーメーションの設定

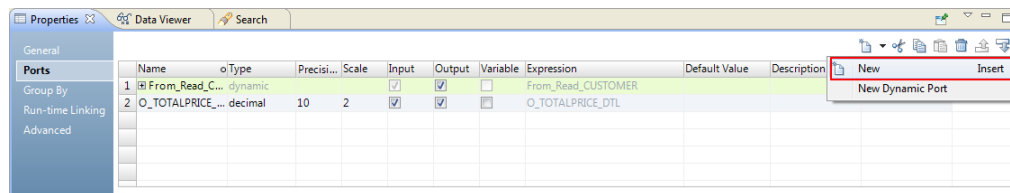
マッピングにアグリゲータトランスフォーメーションを追加して、ジョイナトランスフォーメーションから新しいカラムおよび変更されたカラムを受信するように、動的ポートを設定します。集計式を作成して、顧客注文の合計価格を計算し、さらに顧客別に集計をグループ化します。

1. アグリゲータトランスフォーメーション **Agg_CustomerOrders** をマッピングに追加します。
2. 以下の手順で、アグリゲータトランスフォーメーションに動的ポートを作成します。
 - a. ジョイナトランスフォーメーションの出力グループから、動的ポート **From_Read_CUSTOMER** をアグリゲータトランスフォーメーションにドラッグします。
アグリゲータトランスフォーメーションに、動的ポート **From_Read_CUSTOMER** が表示されます。
 - b. ジョイナトランスフォーメーションの出力グループ内の動的ポート **From_Read_ORDERS** から、生成されたポート **O_TOTALPRICE_DTL** をアグリゲータトランスフォーメーションにドラッグします。

次の図では、ジョイント変換のポートがアグリゲータ変換にリンクされています。



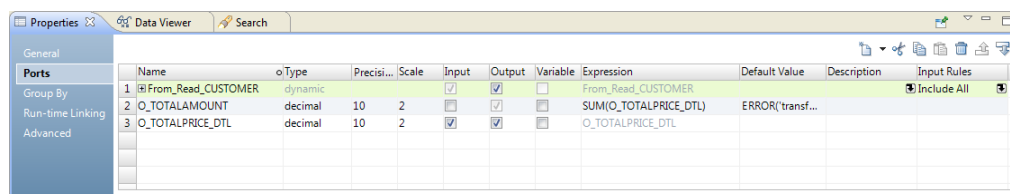
3. **プロパティ** ビューの **ポート** タブをクリックします。
4. **新規** ボタンをクリックして、注文価格を集計するポートを作成します。



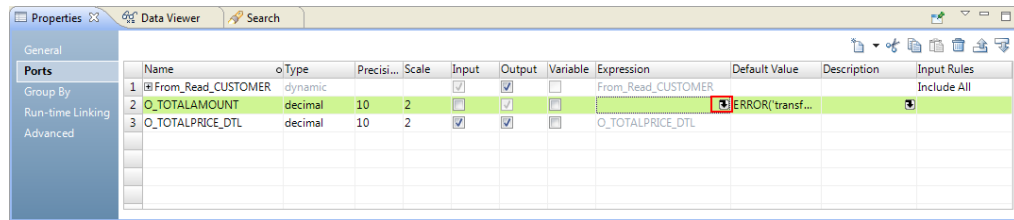
Developer tool は、Field という名前の新しいポートを作成します。

5. 新しいポートを選択し、各カラムの値を次のように変更します。
 - 名前: **O_TOTALAMOUNT**
 - タイプ: **decimal**
 - 精度: **10**
 - スケール: **2**
 - 入力: 選択を解除してこのポートを出力専用ポートにします。

次の図は、アグリゲータ変換のポートを示しています。

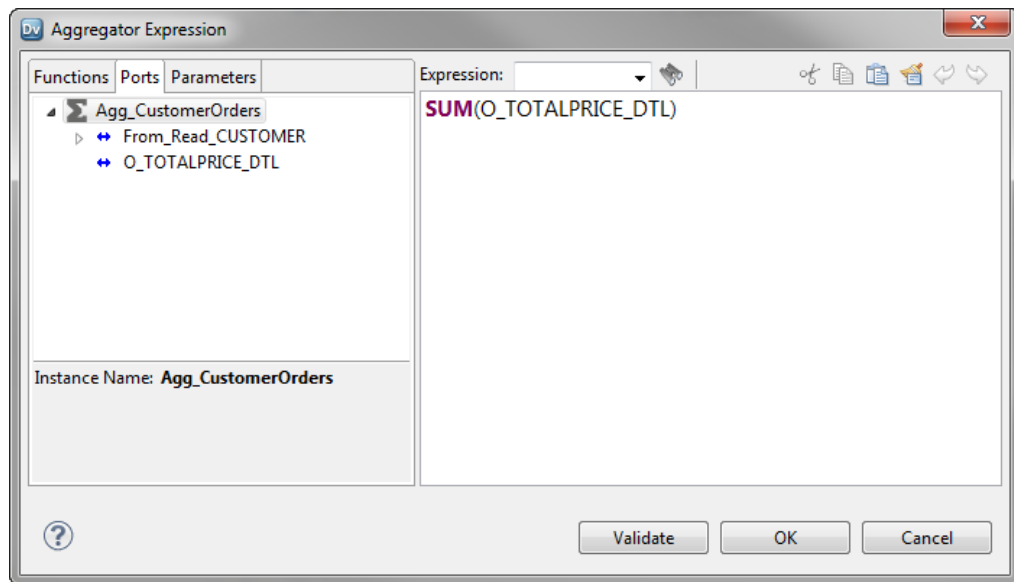


6. O_TOTALAMOUNT ポートの式カラムで、**【開く】** ボタンをクリックします。



【アグリゲータ式】 ウィンドウが表示されます。

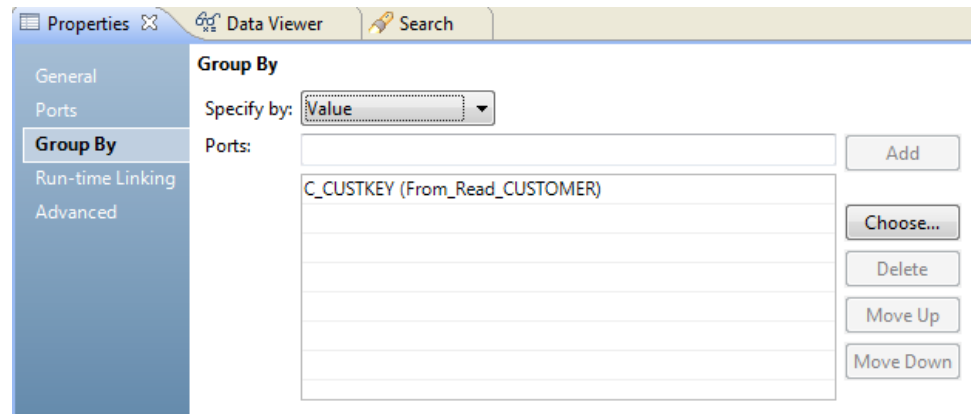
7. エディタの既存の式を、次の式で置換します。SUM(O_TOTALPRICE_DTL)



8. **【検証】** をクリックして式を検証します。
9. **【OK】** をクリックします。
10. **【OK】** をクリックして、**【アグリゲータ式】** エディタを終了します。
11. **【プロパティ】** ビューで **【グループ化】** タブをクリックします。
12. 市場区分別に合計価格を集計するグループ化ポートを、次の手順で指定します。
- 【指定元:】** ドロップダウンリストで **【値】** が選択されていることを確認します。
 - 【選択】** をクリックします。
- 【ポート】** ダイアログボックスが表示されます。

- c. C_CUSTKEY の横のチェックボックスを選択し、[OK] をクリックします。

次の図は、選択されたグループ化ポートを示しています。



アグリゲータトランスフォーメーションデータをプレビューして、期待どおりの結果になっていることを確認できます。マッピングエディタでアグリゲータトランスフォーメーションを右クリックし、**【データビューアの実行】**を選択します。トランスフォーメーションによって計算されたデータが、**【データビューア】**ビューに表示されます。

	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_MKTSEGMENT	O_TOTALAMOUNT	O_TOTALPRICE_DTL
1	65536	Customer#000065536	QK9rk0yHs3...	14	24-965-688-5...	833.21	BUILDING	3320391.15	105991.01
2	131072	Customer#000131072	EHF8GcoL4...	9	19-862-247-6...	3090.02	BUILDING	1178715.91	52437.51
3	256	Customer#000000256	eJ6AggYh80...	10	20-229-271-4...	1299.92	HOUSEHOLD	2925500.20	61122.48
4	65792	Customer#000065792	DLwqCXA0h...	7	17-754-692-6...	8847.80	BUILDING	1145637.31	152952.65
5	512	Customer#000000512	e5 kymvjf6V...	2	12-144-416-6...	3937.58	BUILDING	847430.41	130631.83
6	131584	Customer#000131584	G 24DXCJ,x...	6	16-354-100-1...	1982.52	FURNITURE	3795211.12	189277.59

Row 1 to 1,000

手順 4.書き込みトランスフォーメーションの設定

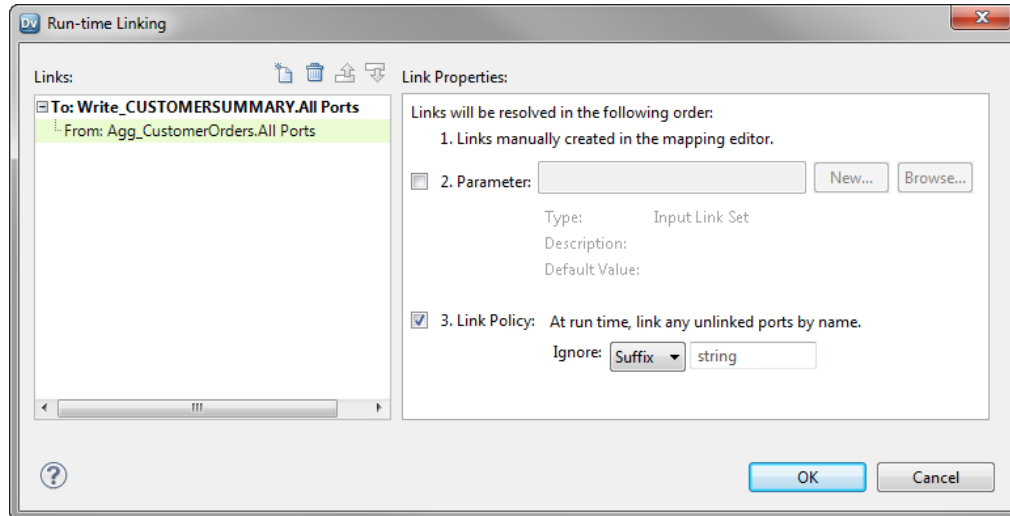
書き込みトランスフォーメーションを追加して、実行時にターゲットから直接、カラム変更を取得するように書き込みトランスフォーメーションを設定します。

1. CUSTOMERSUMMARY リレーショナルデータオブジェクトを書き込みトランスフォーメーションとして追加します。
書き込みトランスフォーメーションが、Write_CUSTOMERSUMMARY としてエディタに表示されます。
2. 書き込みトランスフォーメーションが、メタデータの変更を自動的に再インポートするように設定されていることを確認します。
 - a. **【プロパティ】** ビューで **【全般】** タブをクリックします。
 - b. **【入力ポートの同期】** が選択されていることを確認します。
3. 実行時にターゲットテーブルから直接カラムを取得するように、書き込みトランスフォーメーションを設定します。
 - a. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。
 - b. **【実行時に、データソースからデータオブジェクトのカラムを取得します】** を選択します。

手順 5.ランタイムリンクの作成と設定

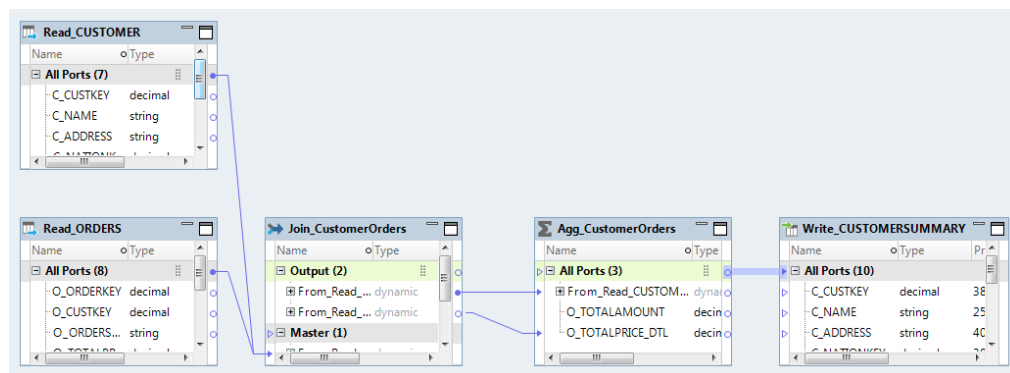
書き込みトランスフォーメーションに対するランタイムリンクを作成し、実行時にポート名でリンクを確立および解決するように、リンクポリシーを設定します。

1. Ctrl キーを押して、アグリゲータトランスフォーメーションから [すべてのポート] グループを、書き込みトランスフォーメーションの [すべてのポート] グループにドラッグします。
[ランタイムリンク] ダイアログボックスが表示されます。
2. [リンクのプロパティ] 領域の [リンクポリシー] が選択されていることを確認します。これにより、実行時に自動的に名前がポートがリンクされます。



3. [OK] をクリックします。

Developer tool は、アグリゲータトランスフォーメーションと書き込みトランスフォーメーション間のランタイムリンクを作成します。



手順 6.マッピングの検証と実行

マッピングを検証して実行します。ターゲットデータオブジェクト内のデータをプレビューして結果を確認します。

1. マッピングエディタで、[編集] > [検証] をクリックします。
2. マッピングが有効であれば、[ファイル] > [保存] をクリックしてマッピングを保存します。
3. [実行] > [マッピング] をクリックします。

【マッピングの実行】 ウィンドウに、マッピング実行の進行状況が表示されます。 マッピングが実行され、出力がターゲットファイルに書き込まれます。

4. 【オブジェクトエクスプローラ】 ビューで、プロジェクト内の CUSTOMERSUMMARY データオブジェクトを探してダブルクリックします。

データオブジェクトがエディタに表示されます。

5. 【ウィンドウ】 > 【ビューの表示】 > 【データビューア】 をクリックします。

【データビューア】 ビューが表示されます。

6. 【データビューア】 ビューで、【実行】 をクリックします。

【データビューア】 ビューが起動して、データが表示されます。

この例では、C_TOTALAMOUNT カラムに、集計された顧客注文の合計価格が表示されています。

Output								
Name: CUSTOMERSUMMARY								
	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_MKTSEGME...	C_TOTALAMOUNT
1	287	Customer#000...	KTsaTAJRC0e...	4	14-330-840-6321	1734.18	MACHINERY	701351.00
2	1055	Customer#000...	Z3AggyEMPM...	7	17-802-131-7180	639.93	HOUSEHOLD	1549236.00
3	32	Customer#000...	JD2xZzi UmId,D...	15	25-430-914-2194	3471.53	BUILDING	1336868.00
4	544	Customer#000...	Jv7vcM,oE,HEy...	5	15-572-651-1323	4974.68	AUTOMOBILE	2900638.00
5	289	Customer#000...	NUilehg0nVok...	10	20-456-773-7693	-215.75	AUTOMOBILE	2893675.00
6	545	Customer#000...	AsYw6k,nDUQ...	10	20-849-123-8918	7505.33	AUTOMOBILE	975375.00
7	1057	Customer#000...	xyV8 FbW4xS,J...	24	34-750-735-1314	-377.11	AUTOMOBILE	2838452.00
8	34	Customer#000...	Q6G9wZ6dncz...	15	25-344-968-5422	8589.70	HOUSEHOLD	4295230.00
9	290	Customer#000...	8OIPT9G 8UqV...	4	14-458-625-5633	1811.35	MACHINERY	618490.00
10	1058	Customer#000...	R0NIeC5VDQ4r...	19	29-818-620-9637	6807.55	MACHINERY	1252089.00

手順 7. ソーススキーマ変更後のマッピングの実行

顧客データテーブルと顧客注文データテーブルを提供している部門が、各テーブルに新しいカラム Comments を追加します。 動的マッピングのカラム変更を確認して、マッピングを検証および再実行します。 ターゲットデータオブジェクト内のデータをプレビューして、更新された結果を確認します。

次の表に、新しい C_COMMENT カラムで更新された CUSTOMER テーブルの各カラムとメタデータを示します。

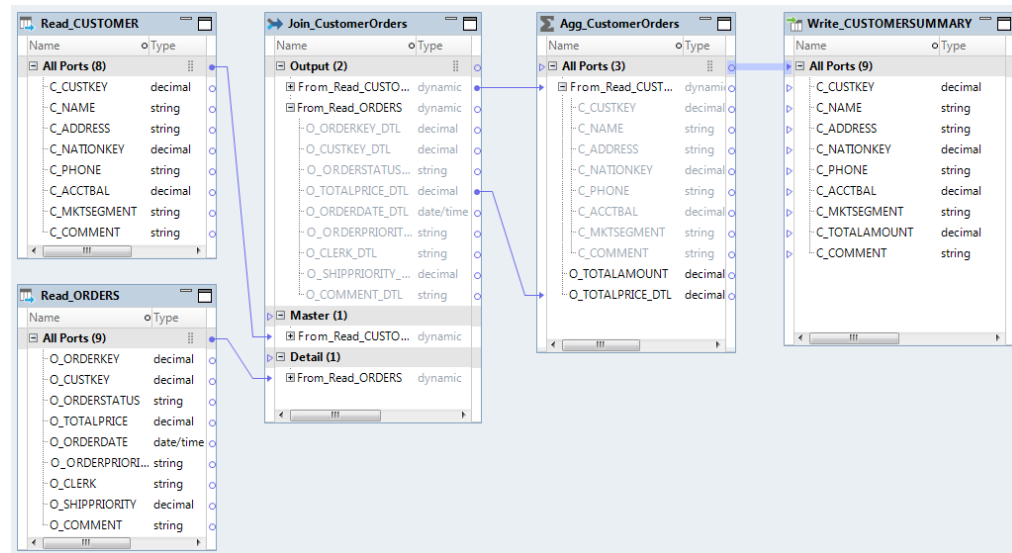
名前	ネイティブタイプ	精度	スケール
C_CUSTKEY	number(p,s)	38	0
C_NAME	varchar2	25	0
C_ADDRESS	varchar2	40	0
C_NATIONKEY	number(p,s)	38	0
C_PHONE	varchar2	15	0
C_ACCTBAL	number(p,s)	10	2
C_MKTSEGMENT	varchar2	10	0
C_COMMENT	varchar2	117	0

次の表に、新しい O_COMMENT カラムで更新された ORDERS テーブルの各カラムとメタデータを示します。

名前	ネイティブタイプ	精度	スケール
O_ORDERKEY	number(p,s)	38	0
O_CUSTKEY	number(p,s)	38	0
O_ORDERSTATUS	varchar2	1	0
O_TOTALPRICE	number(p,s)	10	2
O_ORDERDATE	date	19	0
O_ORDERPRIORITY	varchar2	15	0
O_CLERK	varchar2	15	0
O_SHIPPRIORITY	number(p,s)	30	0
O_COMMENT	varchar2	79	0

1. マッピングエディタで、マッピングに対する変更を確認します。

読み取りおよび書き込みトランスフォーメーションによって、新しいカラムが自動的に反映されます。ジョイントランスフォーメーションおよびアグリゲータトランスフォーメーションの動的ポートには、新しいカラム C_COMMENT と O_COMMENT が、それぞれ対応する読み取りトランスフォーメーションから自動的に取り込まれます。



2. 変更されたマッピングを検証するには、**【編集】** > **【検証】** をクリックします。
3. マッピングが有効であれば、**【ファイル】** > **【保存】** をクリックしてマッピングを保存します。
4. **【実行】** > **【マッピング】** をクリックします。
【マッピングの実行】 ウィンドウに、マッピング実行の進行状況が表示されます。マッピングが実行され、出力がターゲットファイルに書き込まれます。
5. **【オブジェクトエクスプローラ】** ビューで、プロジェクト内の CUSTOMERSUMMARY データオブジェクトを探してダブルクリックします。

データオブジェクトがエディタに表示されます。

6. **【ウィンドウ】** > **【ビューの表示】** > **【データビューア】** をクリックします。
【データビューア】 ビューが表示されます。
7. **【データビューア】** ビューで、**【実行】** をクリックします。
【データビューア】 ビューが起動して、データが表示されます。
8. ソーススキーマの変更後、マッピングに期待どおりの結果が表示されていることを確認します。
C_TOTALAMOUNT カラムに、集計された顧客注文の合計価格が表示されています。

使用例: さまざまなソースおよびターゲットの動的マッピングの再利用

string 値の先頭と末尾の空白を削除するためにさまざまなデータファイルをクリーンする必要がある組織の開発者という立場を想定してみましょう。データファイルにはさまざまなカラム名と、複数の string 型カラムが含まれています。開発者は、さまざまなソースから文字列の先頭と末尾の空白を削除し、出力をさまざまなターゲットに書き込む動的マッピングを開発する必要があります。

ソースファイル

ソースファイルは、先頭と末尾に空白を含む string 型データを含むフラットファイルです。読み取りトランスフォーマーションのソースファイルには Customer_FF および orders_FF が含まれています。

この例では、以下の手順に従って、最初のマッピング実行で Customer_FF ファイルから読み取り、2 番目のマッピング実行で orders_FF ファイルから読み取っています。

Customer_FF のカラムとデータ

Customer_FF には、以下のカラムがあります。

```
C_Id  
C_Fullname  
C_title  
C_comp  
C_addr  
C_suite  
C_city  
C_state  
C_zip5  
C_country  
C_phone  
C_fax  
C_date  
C_email  
C_description
```

ここで、C_ID および C_zip5 カラムのデータ型は数値、その他のカラムのデータ型は string です。

Customer_FF には、以下のデータが含まれています。

```
C_Id,C_Fullname,C_title,C_comp,C_addr,C_suite,C_city,C_state,C_zip5,C_country,C_phone,C_fax,C_date,C_email  
,C_description  
1, Smith John,Account Executive,DKR MANAGEMENT COMPANY INC,100 High Street,5406,Anytown,TN,22342,USA,  
4047668150,2124031386,31/08/1985,bwilliams@yahoo.com, ACTIVE  
2,Balasubramanian Krishna,Account Executive,EASTON & COMPANY,71 Congress Parkway,789,Bangalore,Karnataka,  
38103,India,4046345228,4151689756,29/10/1985,bmatthewc@univ.edu, ACTIVE  
3, Johnson Lars,Regional Sales Exec,GREATER BAY BANCORP,123 Snow St.,43543,St. Paul,MN,55103,USA,  
4046581534,6122945948,7/9/1992, ehpuniv.edu,INACTIVE
```

4,Zogby Kevin,Regional Sales Exec, HEWLETT-PACKARD,317 29th. St.,5856,San Francisco,CA,94116,USA,
 4042662730,4155466814,7/8/1985,grobertwuniv.edu, ACTIVE
 5,Franklin Roosevelt,Sales Representative,JAYD TRADING,1511 Wacker Dr,6334,Chicago,IL,60606,USA,
 7703965851,2065075486,20/10/1982,trichard@univ.edu,INACTIVE
 6, Cruz Emilio,Sales Representative,JEFFERSON-PILOT LIFE INSURANCE,700 Ponce de Leon Blvd,757,Miami,FL,
 33134,USA,4043500799,2127655499,31/07/1983,ahelle@mailcity.com, ACTIVE
 7, King BB,Sales Representative,KUWAIT PETROLEUM CORPORATION,18 Beale St,967,Memphis,TN,38103,USA,
 4046243979,2151717120,27/09/1989, glizziem@univ.edu ,INACTIVE
 8,Presley Elvis,Sales Representative,PRINCIPIA PARTNERS,45 N Green St.,43546,Tupelo,MS,38804,USA,
 4043733125,3311313591,26/07/1992,, ACTIVE
 9,Olson Floyd,Acct MGR., SOLITON ASSOCIATES INC.,21 Lake Harriet Pkwy,869790,Mineapolis,MN,55410,USA,
 7706425402,3232429056,27/08/1993,,INACTIVE
 10,Chu Steven,Account Executive,WQXR,2100 Sepulveda Blvd,3434,Los Angeles,CA,90049,USA,
 4042319005,2126509756,29/09/1988,akennetha@univ.edu, ACTIVE

例えば、1 行目と 3 行目では名前の先頭にスペースがあります。

1, Smith John,
 3, Johnson Lars,

orders_FF のカラムとデータ

orders_FF には、以下のカラムがあります。

OrderID
 Customer_ID
 Company
 CompanyAddress
 CompanyCity
 CompanyState
 CompanyZip
 OrderContact
 DeliveryAddress
 DeliveryCity
 DeliveryState
 PaymentType
 PaymentTerms
 Title
 DeliveryOption
 DeliveryVendor
 ConfirmationCode
 OrderAmount
 OrderType
 ProductDescription

ここで、Customer_ID カラムのデータ型は数値、その他のカラムのデータ型は string です。

orders_FF には、以下のデータが含まれています。

0-5079,10110085,JOSEPH TAL LYON & ROSS,96 FISHER ROAD, MAHWAH,NJ,7430,PARKE PERSLEY OR RAYFORD LECROY,
 96 FISHER ROAD,MAHWAH,NJ,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
 44162,\$21.00 ,Generic,O/L/B P/W L/S TAWNY SHIMMER .08 OZ.
 0-6658,10110086,NRCA,10255 W.HIGGINS RD., ROSEMONT,IL,60018-5607,ROLANDA SORTO,10255 W.HIGGINS
 RD.,ROSEMONT,IL,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
 44163,\$56.40 ,Generic,O-L-B PW LIPSTYLO LASTING PERFECTION .08 OZ.
 0-8195,10110087,POND EQUITIES,4522 FT. HAMILTON PKWY., BROOKLYN,NY,11219, KONSTANTIN PEDDICORD,4522 FT.
 HAMILTON PKWY.,BROOKLYN,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
 44164,\$78.00 ,Generic,O/L/B P/W L/S TAWNY SHIMMER LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .08 OZ.
 0-9130,10110088, SCHRODER & COMPANY ,787 SEVENTH AVENUE, NEW YORK,NY,10019,GIORGIA
 TWITCHELL,787 SEVENTH AVENUE,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel
 Service Air,44165,\$14.00 ,Generic,A/COL L PERFECTION L/S REF P SUPREME LASTING PERFECTION LIPSTYLO TAWNY
 SHIMMER .08 OZ.
 0-9352,10110089,YUASA TRADING COMPANY (AMERICA),150 EAST 52ND STREET,NEW YORK,NY,10005,STEFFI MCGLOWN,150
 EAST 52ND STREET,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,
 44166,\$54.00 ,Generic,O/L/B L PERFECTION REF LIPSTYLO COFFEE PEACH SUPREME .08 OZ.
 0-9517,10110090,DAI ICHI KANGYO BANK,1 WORLD TRADE CENTRE SUITE 49 - 11,NEW YORK,NEW YORK,10048,AIKEN
 DOBRICK,1 WORLD TRADE CENTRE SUITE 49 - 11,NEW YORK,NEW YORK,American Express,CHARGE,Account
 Executive,UPR,United Parcel Service Red,44167,\$58.00 ,Generic,LASTING PERFECTION LIP COLOR HOLLYWOOD
 GLAMOUR 1.7 G MAUVE ICE #752
 0-9639,10110091,FIRST GLOBAL SECURITIES,614 EAST COLORADO BLVD.,PASADENA,CA,91101, KIRSTENI SIPPEL,614
 EAST COLORADO BLVD.,PASADENA,CA,American Express,CHARGE,Account Executive,FSO,Federal Express Overnight,

44168,\$24.00 ,Generic,A/COL L PERFECTION L/S REF P SUPREME .08 OZ.
 0-9761,10110092,MILTON PARTNERS,56 MASON STREET, GREENWICH ,CT,6830,ORLANTA DYSON,56 MASON
 STREET,GREENWICH,CT,American Express,CHARGE,Account Executive,UPI,United Parcel Service International,
 44169,\$75.20 ,Generic,LASTING PERFECTION LIPSTYLO PEACH SU .08 OZ.
 0-9883,10110093, TAX ANALYSTS ,6830 N. FAIRFAX DRIVE,ARLINGTON,VA,22213,NEWLIN MCCART,6830 N. FAIRFAX
 DRIVE,ARLINGTON,VA,American Express,CHARGE,Account Executive,FSO,Federal Express Overnight,
 44170,\$275.40 ,Generic,O/L/B L PERFECTION L/STYLO REF P SUPRE
 0-5438,10110094,VECTORMEX,535 MADISON AVENUE,NEW YORK,NY,10022,LONNA HUGGINS,535 MADISON AVENUE,NEW
 YORK,NY,American Express,CHARGE,Account Executive,FSO,Federal Express Overnight,
 44171,\$60.00 ,Generic,LASTING PERFECTION DOUBLE PERFORMANCE LIPSTICK PEACH SUPREME .08 OZ.

例えば、4 行目では会社名の先頭と末尾にスペースがあります。

0-9130,10110088, SCHRODER & COMPANY ,

ターゲットファイル

ターゲットファイルは、マッピングが文字列値の先頭および末尾の空白を削除した後、データを書き出すフラットファイルです。ターゲットデータオブジェクトのターゲットファイルとして customerTrim.csv ファイルを作成します。

異なるデータソースを使用する場合は、パラメータを使用して実行時に出力ファイル名を変更します。データ統合サービスは、ターゲットファイル名のパラメータ値に基づいて出力ファイルを作成し、システム上の Informatica サービスがインストールされているターゲットディレクトリに保存します。

動的マッピング

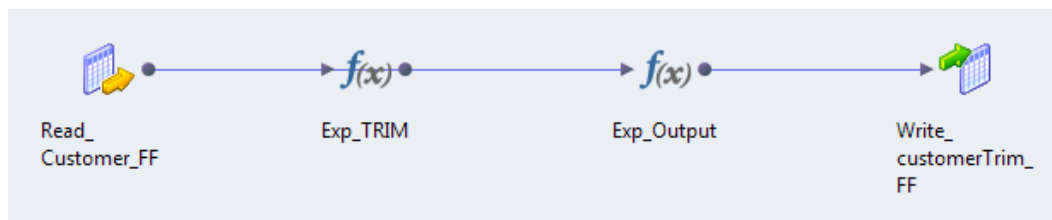
マッピング m_Replication_Template を作成して、次の動的マッピング機能を設定します。

- データオブジェクトのパラメータを使用して、さまざまなソースからデータを読み取る読み取りトランスフォーマーション
- 新しいカラムまたは変更されたカラムを渡すことができる、ダウストリームトランスフォーマーションの動的ポート
- 文字列の先頭および末尾の空白を削除する動的式を含む式トランスフォーマーション
- マッピングフローに基づいてターゲットカラムを作成し、ターゲットファイル名としてターゲットデータオブジェクト内のパラメータを使用する書き込みトランスフォーマーション

マッピングを実行すると、データ統合サービスによって以下のタスクが実行されます。

1. ソースデータオブジェクトのパラメータ値に基づいて適切なソースファイルからデータを読み取ります。
2. 新しいカラムおよび変更されたカラムを動的ポート経由でダウストリームトランスフォーマーションに渡します。
3. 動的式を展開し、動的ポート内の生成された各ポートについて式の関数を処理します。
4. マッピングフローに基づいて書き込みトランスフォーマーションでカラムを作成し、変換されたデータをパラメータ値に基づいて適切なターゲットファイルに書き込みます。

次の図は、マッピング内のオブジェクトを示しています。



マッピングには次のオブジェクトが含まれます。

Read_Customer_FF

フラットファイルソースを表す読み取りトランスフォーメーション。フラットファイルには、先頭および末尾に空白を含む文字列データが含まれています。

Exp_TRIM

string 型ポートの先頭および末尾の空白を削除する動的式を含む式トランスフォーメーション。

Exp_Output

変換された string 型ポートとソースオブジェクトの残りのポートを含む式トランスフォーメーション。

Write_customerTrim_FF

フラットファイルターゲットを表す書き込みトランスフォーメーション。マッピングは出力をフラットファイルターゲットに書き込みます。

手順 1. Read_Customer_FF 読み取りトランスフォーメーションの設定

リソースタイプのパラメータを使用してマッピング実行ごとにソースデータオブジェクトを変更するように、Read_Customer_FF 読み取りトランスフォーメーションを設定します。

1. Customer_FF フラットファイルデータオブジェクトを表す読み取りトランスフォーメーションを追加します。

読み取りトランスフォーメーションが Read_Customer_FF としてエディタに表示されます。

2. **【プロパティ】** ビューで、**【データオブジェクト】** タブをクリックします。

3. **【指定元:】** ドロップダウンリストで、**【パラメータ】** を選択します。

4. **【新規】** をクリックして新しいパラメータを作成します。

【パラメータ】 ダイアログボックスが表示されます。

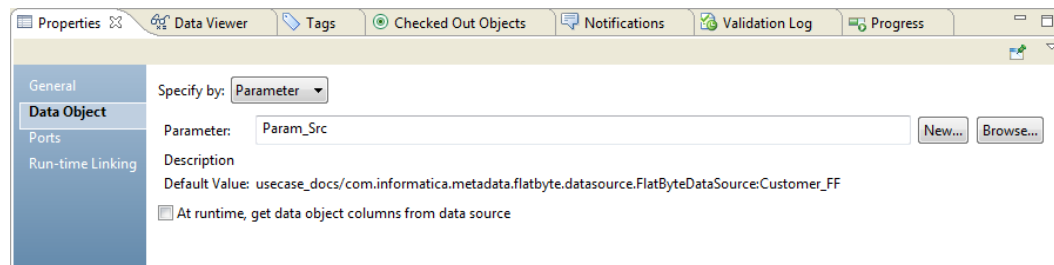
5. パラメータ名として、Param_Src と入力します。

6. **【デフォルト値】** の **【参照】** をクリックします。

7. **【場所の選択】** ダイアログボックスで、デフォルト値として設定するデータオブジェクトを選択します。

サンプルのデフォルト値は MRS//Cust_Dept/Customer_FF です。ここで、MRS はモデルリポジトリサービス、Cust_Dept は Customer_FF データオブジェクトが格納されているプロジェクトです。パラメータの値はマッピングの実行時に変更できます。

次の図は、設定を定義した後の **【データオブジェクト】** タブを示しています。



手順 2. Exp_TRIM 式トランスフォーメーションの設定

式トランスフォーメーション Exp_TRIM をマッピングに追加し、文字列の先頭と末尾のスペースを削除するようにトランスフォーメーションを設定します。

1. 読み取りトランスフォーメーションからカラムを受信するように動的ポートを作成して、文字列ポートのみを含めるように入力ルールを定義します。

2. 動的出力ポートを作成し、文字列の先頭と末尾のスペースを削除するように動的式を定義します。

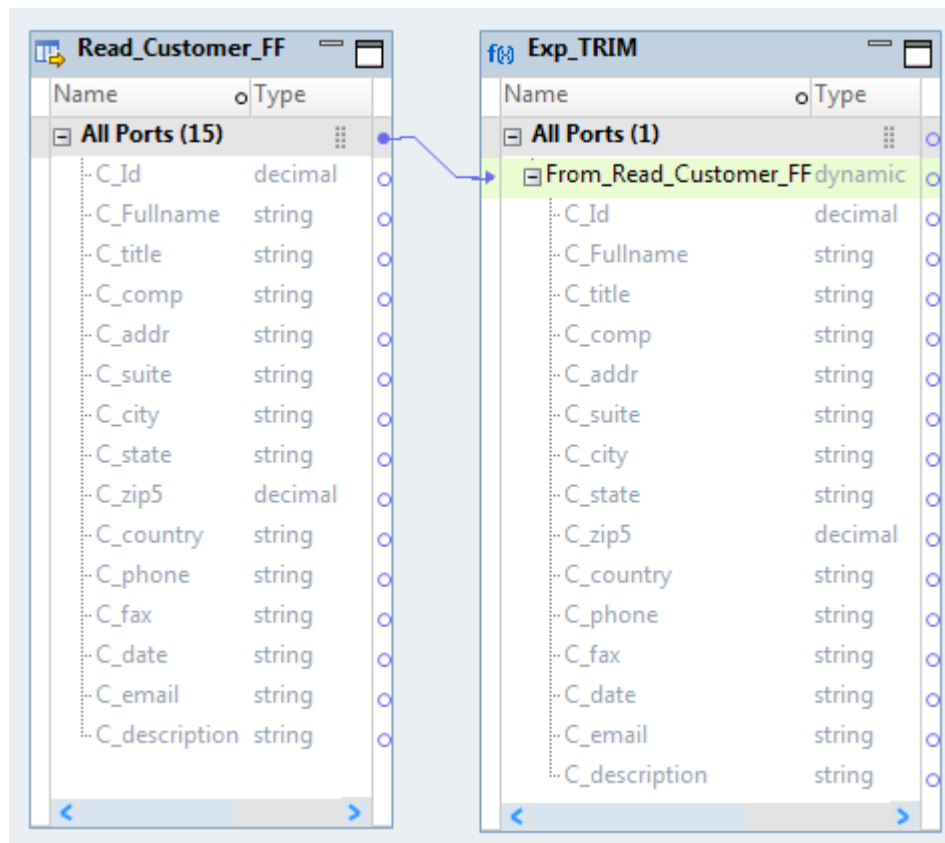
動的ポートの作成と入力ルールの定義

読み取りトランスフォーメーションからカラムを受信するように動的ポートを作成します。動的ポートに文字列ポートのみを含めるように入力ルールを定義します。

1. Read_Customer_FF トランスフォーメーションの [すべてのポート] グループを、Exp_TRIM トランスフォーメーションの [すべてのポート] グループにドラッグします。

Developer tool は、Exp_TRIM トランスフォーメーションに動的ポート From_Read_Customer_FF を作成します。

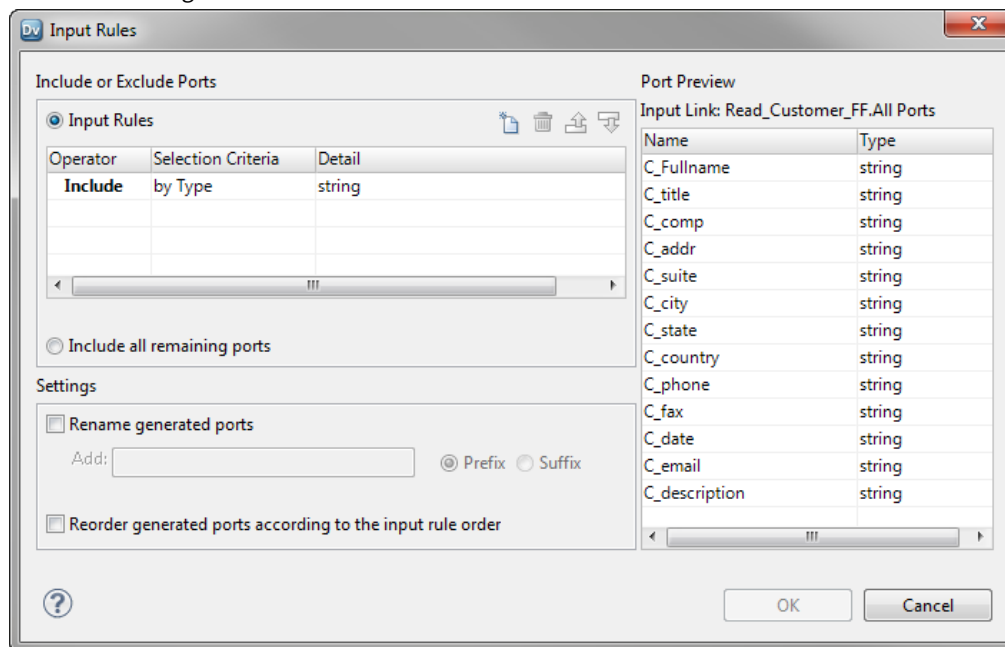
次の図は、読み取りトランスフォーメーションのすべてのポートを生成されたポートとして含む、Exp_TRIM トランスフォーメーションの動的ポートを示しています。



2. 動的ポートを右クリックして、[入力ルール編集] を選択します。
[入力ルール] ダイアログボックスが表示されます。
3. [選択条件] カラムから [タイプ] を選択します。
4. [詳細] ボタンをクリックして、含めるデータ型を選択します。
5. [入力ルール詳細: タイプ別] ダイアログボックスで、リストから [string] データ型を選択します。

6. [ポートのプレビュー] 領域で、string 型ポートのみが表示されていることを確認します。

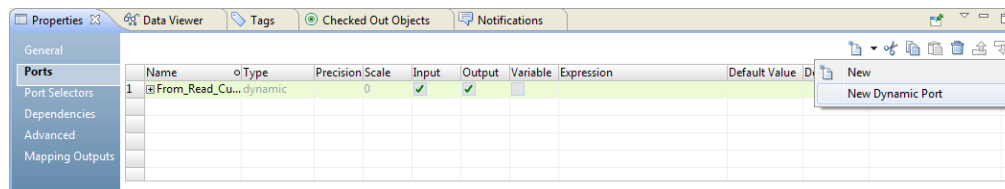
次の図は、[入力ルール] ダイアログボックスの [ポートのプレビュー] 領域に表示された、更新された入力ルールと string 型ポートを示しています。



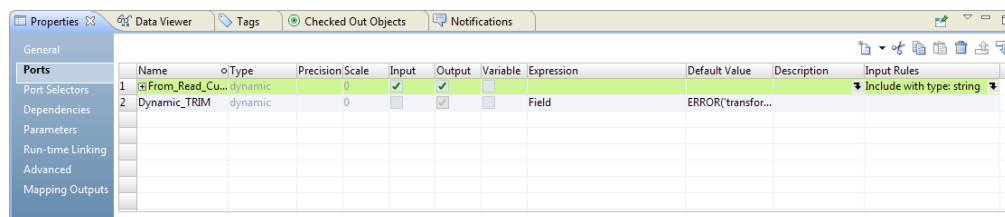
動的ポートの作成と動的式の定義

Exp_TRIM トランスフォーメーションで出力のみのポートとして動的ポートを作成します。文字列の先頭と末尾のスペースを削除するように動的式を定義します。

1. Exp_TRIM トランスフォーメーションの [プロパティ] ビューで、[ポート] タブをクリックします。
2. [新しい動的ポート] をクリックします。

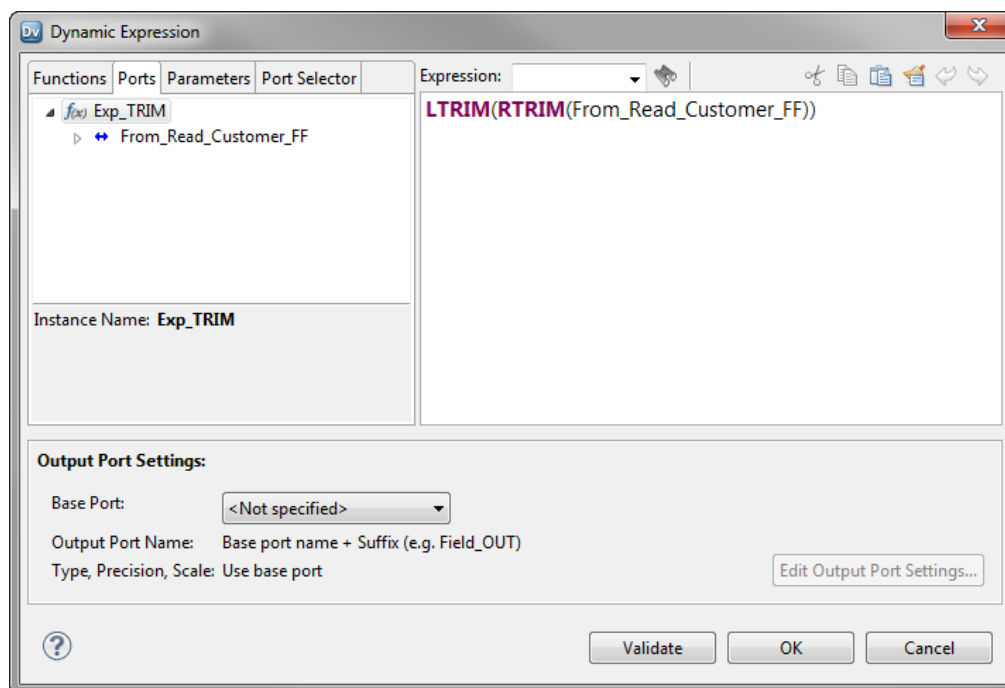


3. [入力] カラムをクリアして、このポートを出力専用ポートにします。
4. Dynamic_TRIM として作成した動的ポートの名前を変更します。



5. Dynamic_TRIM 動的ポートの [式] カラムで、[開く] ボタン (🔗) をクリックします。
[動的式] ウィンドウが開きます。

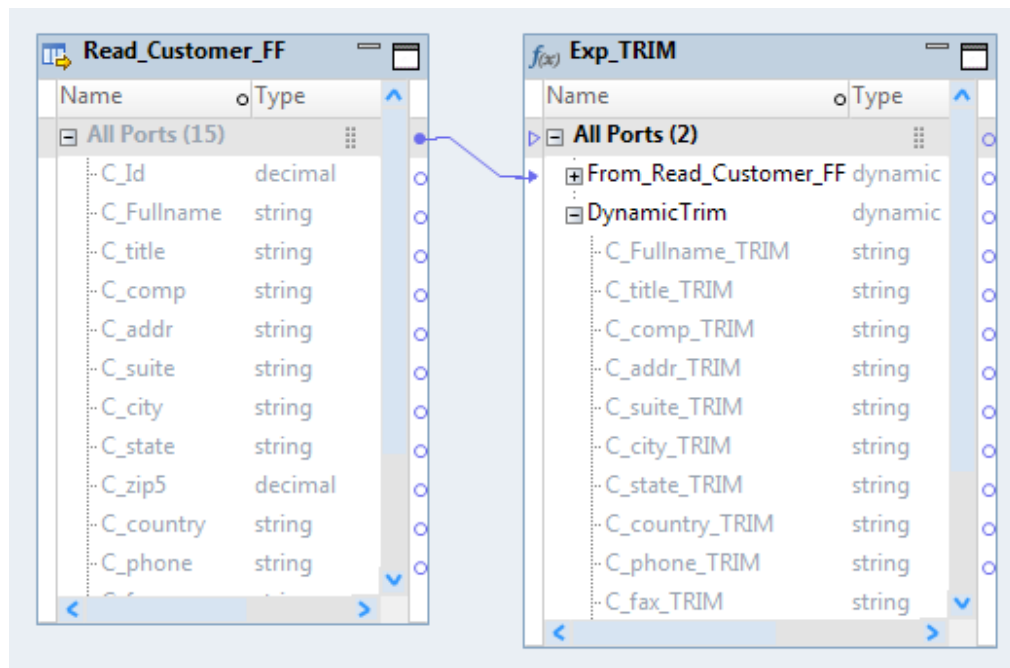
6. エディタの既存の式を、次の式で置換します。LTRIM(RTRIM(From_Read_Customer_FF))



7. **【検証】** をクリックして式を検証します。
8. **【OK】** をクリックして、**【式の検証】** ダイアログボックスを終了します。
9. 式の出力ポートの名前を次のように変更します。
- 【出力ポート設定】** 領域で、ベースポートとして From_Read_Customer_FF を選択します。
 - 【出力ポート設定の編集】** をクリックします。
【出力ポート設定】 ダイアログボックスが表示されます。
 - 【名前】** リストで、**【ベースポート名+サフィックス】** を選択します。
 - 【サフィックス】** ボックスに「_TRIM」と入力します。
 - 【OK】** をクリックします。

10. **[OK]** をクリックして、**[動的式]** エディタを終了します。

次の図は、Dynamic_TRIM 動的ポートと、生成されたポートで名前変更されたものを示しています。



手順 3. Exp_Output 式トランスフォーメーションの設定

式トランスフォーメーション Exp_Output をマッピングに追加します。Exp_TRIM トランスフォーメーションから出力ポートを取得するように動的ポートを作成します。読み取りトランスフォーメーションからポートを取得するように別の動的ポートを作成して、未使用のポートのみを含めるように入力ルールを定義します。

1. Exp_TRIM トランスフォーメーションから、DynamicTrim 動的ポートを Exp_Output トランスフォーメーションの [すべてのポート] グループにドラッグします。

Developer tool で、Exp_Output トランスフォーメーションに動的ポート DynamicTrim が作成されます。

2. Read_Customer_FF トランスフォーメーションの [すべてのポート] グループを、Exp_Output トランスフォーメーションの [すべてのポート] グループにドラッグします。

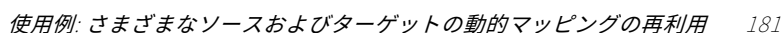
Developer tool で、Exp_Output トランスフォーメーションに動的ポート From_Read_Customer_FF が作成されます。

The screenshot displays three windows from a software interface, likely a data processing or simulation tool, showing the flow of data between different components.

- Read_Customer_FF**: This window lists various customer attributes. The 'C_Id' attribute is highlighted in green, and a blue arrow points from it to the 'Dynamic_TRIM' attribute in the 'Exp_TRIM' window.
- Exp_TRIM**: This window shows a list of attributes, including 'From_Read_Customer_FF' and 'Dynamic_TRIM'. The 'Dynamic_TRIM' attribute is highlighted in green, and a blue arrow points from it to the 'Dynamic_TRIM' attribute in the 'Exp_Output' window.
- Exp_Output**: This window shows a list of attributes, including 'Dynamic_TRIM' and 'From_Read_Customer_FF'. The 'Dynamic_TRIM' attribute is highlighted in green, and a blue arrow points from it to the 'From_Read_Customer_FF' attribute in the 'Exp_Output' window.

The windows are organized into a grid, and the data flow is indicated by blue arrows connecting the highlighted attributes across the windows.

- 次の図は、**【入力ルール】** ダイアログボックスの **【ポートのプレビュー】** 領域に表示された、更新された入力ルールとポートを示しています。





手順 4. Write_customerTrim_FF 書き込みトランスフォーメーションの設定

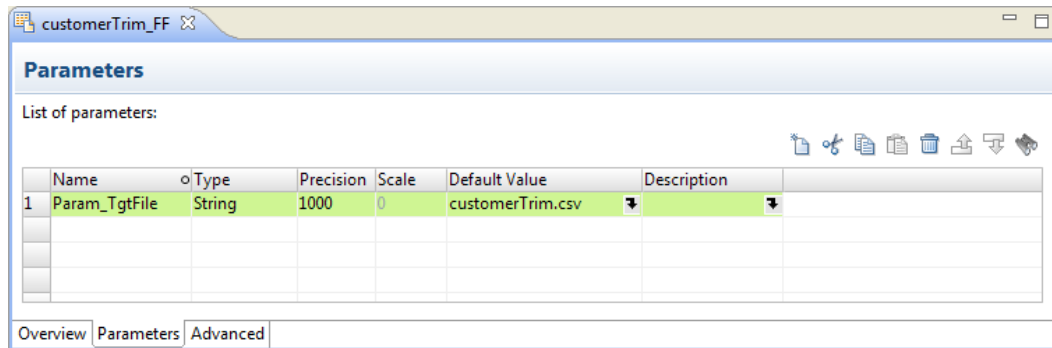
customerTrim_FF データオブジェクトを作成し、出力ファイル名に string 型のパラメータを使用するように設定します。実行時に Exp_Output トランスフォーメーションの列に基づいてターゲットファイルを作成するように Write_customerTrim_FF トランスフォーメーションを設定します。

パラメータを使用するためのデータオブジェクトの設定

customerTrim_FF データオブジェクトを作成して、マッピング内に書き込みトランスフォーメーションとして追加します。出力ファイル名として string 型のパラメータを使用するようにデータオブジェクトを設定します。

- customerTrim.csv ファイルに基づいて、customerTrim_FF データオブジェクトを作成します。
- パラメータを出力ファイルとして使用するには、以下の手順を実行します。
 - データオブジェクトの **【パラメータ】** タブで、**【新規】** ボタン () をクリックして新しいパラメータを作成します。
 - 【名前】** 列で、パラメータ名を Param_TgtFile に変更します。
 - 【デフォルト値】** 列で、**【開く】** ボタン () をクリックします。
【パラメータ値の編集】 ウィンドウが表示されます。
 - デフォルトのファイル名の値として customerTrim.csv を入力し、**【OK】** をクリックします。
- customerTrim_FF データオブジェクトを保存します。

次の図は、新しいパラメータが表示された **【パラメータ】** タブを示しています。

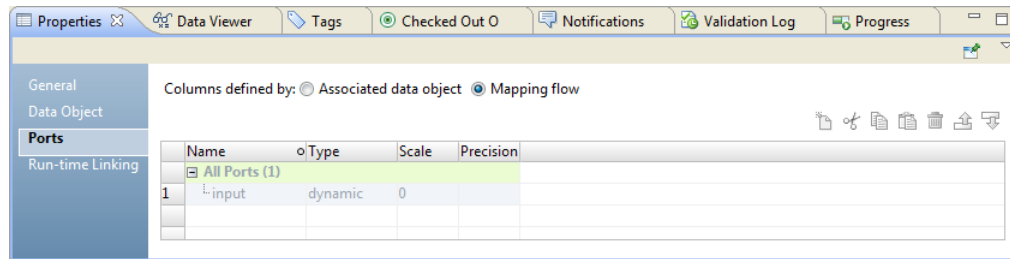


マッピングフローからのターゲット列の作成

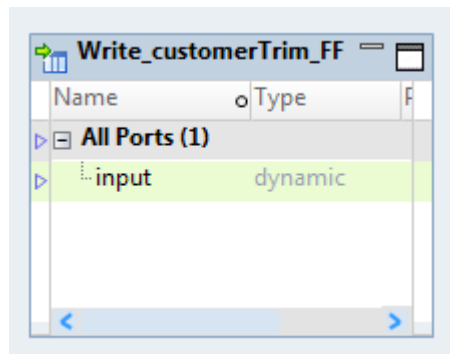
書き込みトランスフォーメーションをマッピングに追加して、Exp_Output トランスフォーメーションの列に基づいて実行時にターゲットファイルを作成するように Write_customerTrim_FF トランスフォーメーションを設定します。

- customerTrim_FF データオブジェクトを、書き込みトランスフォーメーションとしてマッピングに追加します。
- 書き込みトランスフォーメーションの **【プロパティ】** ビューで、**【ポート】** タブをクリックします。
- 【マッピングフロー】** オプションを選択してターゲットの列を定義します。
Developer tool で、Write_customerTrim_FF トランスフォーメーションに動的ポート **input** が作成されます。

次の図は、オプションを選択した後の【ポート】タブを示しています。



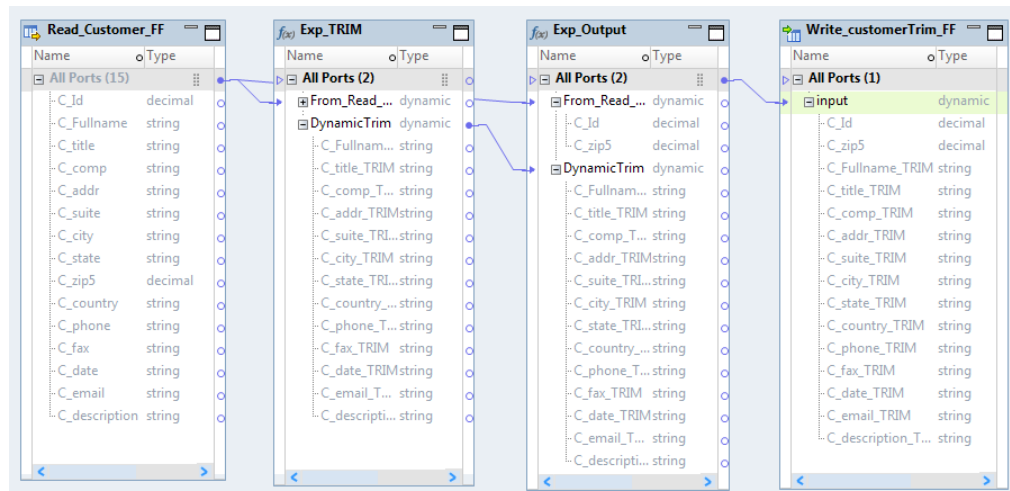
次の画像は、Write_customerTrim_FF トランスフォーメーションの新しい動的ポート **input** を示しています。



4. Exp_Output トランスフォーメーションの【すべてのポート】グループを、Write_customerTrim_FF トランスフォーメーションの input ポートにドラッグします。

Developer tool でリンクが作成されて、Exp_Output トランスフォーメーションの【すべてのポート】グループの各カラムから書き込みトランスフォーメーションの input 動的ポートへのフローが作成されます。

次の図は、書き込みトランスフォーメーションが設定された m_Replication_Template マッピングを示しています。



手順 5. マッピングの検証と保存

ソースデータオブジェクトとターゲットファイルのデフォルトのパラメータ値を使用して、m_ReplicationTemplate マッピングを検証および実行し、結果を表示します。

1. マッピングエディタで、**【編集】** > **【検証】** をクリックします。
2. マッピングが有効であれば、**【ファイル】** > **【保存】** をクリックしてマッピングを保存します。

手順 6. 異なるソースおよびターゲットに対する動的マッピングの実行

動的マッピングを開発すると、マッピングを実行して異なるソースにアクセスし、パラメータ値に基づいて異なるターゲットに書き込めるようになります。

Customer_FF ソースのマッピングの実行

ソースデータオブジェクトとターゲットファイルのデフォルトのパラメータ値を使用して m_ReplicationTemplate マッピングを実行し、結果を表示します。マッピングは Customer_FF ソースファイルから読み取り、customerTrim.csv ターゲットファイルに書き込みます。

1. **【実行】** > **【マッピング】** をクリックします。
【マッピングの実行】 ウィンドウに、マッピング実行の進行状況が表示されます。マッピングが実行され、出力がターゲットファイルに書き込まれます。
2. ターゲットファイルに書き込まれた結果を確認するには、システム上の Informatica サービスがインストールされているターゲットディレクトリに移動します。

```
<Informatica Installation Directory>\tomcat\bin\target
```

3. customerTrim.csv ファイルを開き、文字列値の先頭と末尾に空白が含まれていないことを確認します。

ファイルの各行には、各カラムのデータがターゲットオブジェクトに現れる順番で表示されます（例：C_id、C_zip5、C_Fullname、C_title、C_comp）。例えば、ファイルの先頭の 5 行には、各文字列値の先頭と末尾の空白が削除された次のようなデータが格納されます。

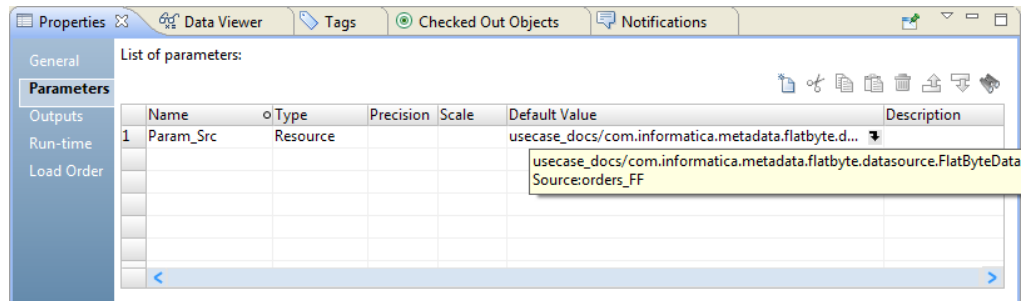
```
1,22342,Smith John,Account Executive,DKR MANAGEMENT COMPANY INC,100 High Street,5406,Anytown,TN,USA,
4047668150,2124031386,31/08/1985,bwilliams@yahoo.com,ACTIVE
2,38103,Balasubramanian Krishna,Account Executive,EASTON & COMPANY,71 Congress Parkway,
789,Bangalore,Karnataka,India,4046345228,4151689756,29/10/1985,bmatthewc@univ.edu,ACTIVE
3,55103,Johnson Lars,Regional Sales Exec,GREATER BAY BANCORP,123 Snow St.,43543,St. Paul,MN,USA,
4046581534,6122945948,7/9/1992,ehpuniv.edu,INACTIVE
4,94116,Zogby Kevin,Regional Sales Exec,HEWLETT-PACKARD,317 29th. St.,5856,San Francisco,CA,USA,
4042662730,4155466814,7/8/1985,grobertwuniv.edu,ACTIVE
5,60606,Franklin Roosevelt,Sales Representative,JAYD TRADING,1511 Wacker Dr,6334,Chicago,IL,USA,
7703965851,2065075486,20/10/1982,trichard@univ.edu,INACTIVE
```

パラメータ値の変更

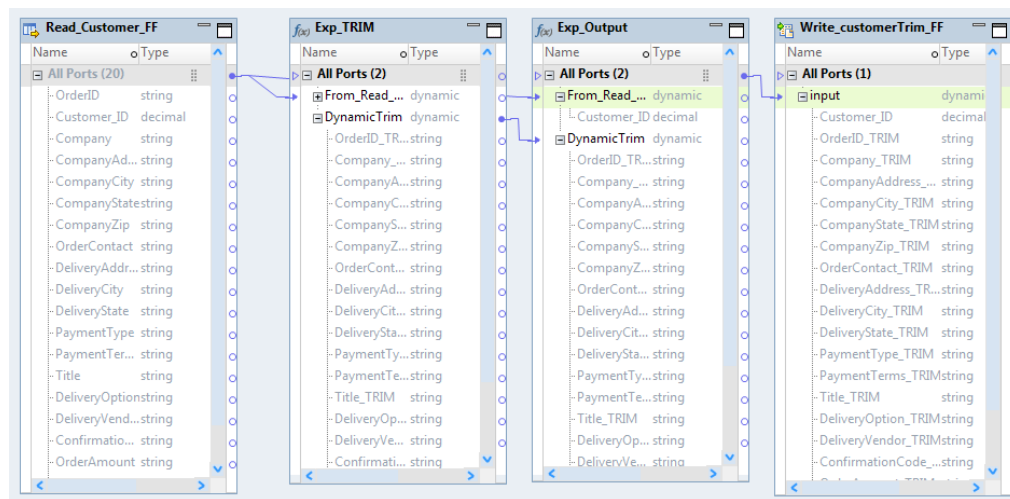
ソースデータオブジェクトのパラメータ値とターゲットデータオブジェクトの出力ファイル名を変更します。

1. ソースデータオブジェクトのパラメータ値を変更するには、以下の手順を実行します。
 - a. マッピングの**【プロパティ】** ビューで、**【パラメータ】** タブをクリックします。
 - b. ソースオブジェクトの Param_Src パラメータを探します。
 - c. **【デフォルト値】** カラムで、**【開く】** ボタン (🔗) をクリックします。
【場所の選択】 ダイアログボックスが表示されます。
 - d. orders_FF データオブジェクトを選択します。

次の図は、デフォルト値が更新されたマッピングの【パラメータ】タブを示しています。

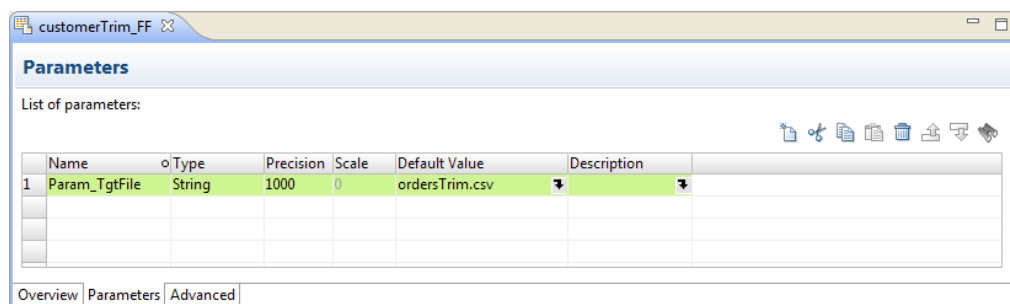


次の図は、読み取りトランスフォーメーションの orders_FF データオブジェクトからのポートを反映するマッピングを示しています。動的ポートは、新しく生成されたポートを反映しています。



2. ターゲットファイル名のパラメータ値を変更するには、以下の手順を実行します。
 - a. customerTrim_FF ターゲットデータオブジェクトを開きます。
 - b. データオブジェクトの【パラメータ】タブで、ターゲットファイル名を表す Param_TgtFile パラメータを探します。
 - c. 【デフォルト値】カラムで、【開く】ボタン (🔍) をクリックします。
【パラメータ値の編集】ウィンドウが表示されます。
 - d. デフォルトのファイル名の値を ordersTrim.csv に変更し、【OK】をクリックします。

次の図は、デフォルト値が更新された customerTrim_FF データオブジェクトの【パラメータ】タブを示しています。



orders_FF ソースのマッピングの実行

マッピングを検証し、異なるソースおよびターゲットの m_ReplicationTemplate マッピングを実行します。マッピングは orders_FF ソースファイルから読み取り、ordersTrim.csv ターゲットファイルに書き込みます。

1. マッピングエディタで、**[編集]** > **[検証]** をクリックします。
2. マッピングが有効であれば、**[ファイル]** > **[保存]** をクリックしてマッピングを保存します。
3. **[実行]** > **[マッピング]** をクリックします。
[マッピングの実行] ウィンドウに、マッピング実行の進行状況が表示されます。マッピングが実行され、出力がターゲットファイルに書き込まれます。
4. ターゲットファイルに書き込まれた結果を確認するには、システム上の Informatica サービスがインストールされているターゲットディレクトリに移動します。
<Informatica Installation Directory>\tomcat\bin\target
5. ordersTrim.csv ファイルを開き、文字列値の先頭と末尾に空白が含まれていないことを確認します。

ファイルの各行には、各カラムのデータがターゲットオブジェクトに現れる順番で表示されます（例: Customer_Id、Order_ID、Company、CompanyAddress、CompanyCity）。例えば、ファイルの先頭の 5 行には、各文字列値の先頭と末尾の空白が削除された次のようなデータが格納されます。

```
10110085,0-5079,JOSEPH THAL LYON & ROSS,96 FISHER ROAD,MAHWAH,NJ,7430,PARKE PERSLEY OR RAYFORD LECROY,96  
FISHER ROAD,MAHWAH,NJ,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44162,$21.00,Generic,0/L/B P/W L/S TAWNY SHIMMER .08 OZ.  
10110086,0-6658,NRCA,10255 W.HIGGINS RD.,ROSEMONT,IL,60018-5607,ROLANDA SORTO,10255 W.HIGGINS  
RD.,ROSEMONT,IL,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44163,$56.40,Generic,0-L.B PW LIPSTYLO LASTING PERFECTION .08 OZ.  
10110087,0-8195,POND EQUITIES,4522 FT. HAMILTON PKWY.,BROOKLYN,NY,11219,KONSTANTIN PEDDICORD,4522 FT.  
HAMILTON PKWY.,BROOKLYN,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44164,$78.00,Generic,0/L/B P/W L/S TAWNY SHIMMER LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .08 OZ.  
10110088,0-9130,SCHRÖDER & COMPANY,787 SEVENTH AVENUE,NEW YORK,NY,10019,GIORGIA TWITCHELL,787 SEVENTH  
AVENUE,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44165,$14.00,Generic,A/COL L PERFECTION L/S REF P SUPREME LASTING PERFECTION LIPSTYLO TAWNY SHIMMER .08  
OZ.  
10110089,0-9352,YUASA TRADING COMPANY (AMERICA),150 EAST 52ND STREET,NEW YORK,NY,10005,STEFFI MCGLOWN,150  
EAST 52ND STREET,NEW YORK,NY,American Express,CHARGE,Account Executive,UPA,United Parcel Service Air,  
44166,$54.00,Generic,0/L/B L PERFECTION REF LIPSTYLO COFFEE PEACH SUPREME .08 OZ.
```

第 8 章

マッピング管理

この章では、以下の項目について説明します。

- [マッピング管理の概要, 187 ページ](#)
- [マッピングジョブのプロパティの表示, 188 ページ](#)
- [マッピングジョブのサマリ統計の表示, 188 ページ](#)
- [マッピングジョブの詳細統計の表示, 188 ページ](#)
- [マッピングジョブのログの表示, 189 ページ](#)
- [デプロイ済みのマッピングジョブの再発行, 189 ページ](#)
- [マッピングジョブのキャンセル, 190 ページ](#)
- [拒否ファイル, 190 ページ](#)

マッピング管理の概要

アドホックのマッピングジョブを実行する場合、またはデータ統合サービスにマッピングをデプロイする場合は、Monitoring ツールまたは Administrator ツールでジョブを監視できます。データ統合サービスがターゲットに行を書き込むことができない場合、拒否ファイルで行に関する情報を確認することもできます。ジョブを監視したり拒否ファイルを表示したりするには、適切な特権が必要です。

マッピングジョブは、以下の場所で監視することができます。

- Monitoring ツール。Developer tool で、**[進行状況]** ビューの **[メニュー]** ボタンをクリックし、**[ジョブの監視]** を選択します。マッピングジョブを実行するデータ統合サービスを選択して、**[OK]** をクリックします。Monitoring ツールが開きます。
- Administrator ツール。Administrator ツールでマッピングを監視するには、**[モニタ]** タブをクリックします。

マッピングジョブを監視する場合、ジョブのサマリ統計または実行統計を表示できます。**[サマリ統計]** ビューには、ドメイン内のマッピングジョブのステータスの概要がグラフィック表示されます。マッピングプロパティと統計の表示、ジョブログの表示、ジョブのキャンセル、デプロイ済みのマッピングの再発行を行うには、**[実行統計]** ビューを使用します。

マッピングジョブのプロパティの表示

アドホックまたはデプロイ済みのマッピングジョブを監視する場合、ジョブのプロパティを表示できます。プロパティには、ジョブ ID、ジョブの開始ユーザー、ジョブの継続時間が含まれます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. **【アドホックジョブ】** フォルダを選択するか、アプリケーションを展開して **【デプロイ済みのマッピングジョブ】** を選択します。

ジョブのリストが **【コンテンツ】** パネルに表示されます。**【コンテンツ】** パネルには、ジョブの名前、状態、ID、継続時間など、各種プロパティが表示されます。

4. **【コンテンツ】** パネルでジョブを選択します。
【詳細】 パネルにジョブのプロパティが表示されます。

マッピングジョブのサマリ統計の表示

アドホックまたはデプロイ済みのマッピングジョブのスループットおよびリソース使用状況の統計を表示できます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. **【アドホックジョブ】** フォルダを選択するか、アプリケーションを展開して **【デプロイ済みのマッピングジョブ】** を選択します。

ジョブのリストが **【コンテンツ】** パネルに表示されます。

4. **【コンテンツ】** パネルでジョブを選択します。
【詳細】 パネルにジョブのプロパティが表示されます。
5. **【詳細】** パネルで **【サマリ統計】** ビューをクリックします。

【サマリ統計】 ビューには、ソースおよびターゲットのスループットおよびリソース使用状況の統計が表示されます。

必要に応じて、昇順または降順で統計をソートすることができます。列ヘッダーをクリックして昇順に列をソートします。再度列ヘッダーをクリックして降順に列をソートします。

マッピングジョブの詳細統計の表示

個別のローカルプロセスで実行されている、アドホックまたはデプロイ済みのマッピングジョブのスループットおよびリソース使用状況のグラフを表示できます。1 分を超えて実行されているジョブの詳細な統計が表示されます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. **【アドホックジョブ】** フォルダを選択するか、アプリケーションを展開して **【デプロイ済みのマッピングジョブ】** を選択します。

ジョブのリストが [コンテンツ] パネルに表示されます。

4. [コンテンツ] パネルでジョブを選択します。

[詳細] パネルにジョブのプロパティが表示されます。

5. [詳細] パネルで [詳細統計] ビューをクリックします。

[詳細統計] ビューに、スループットのグラフとリソース使用状況のグラフが表示されます。

必要に応じて、[詳細統計] ビューで次のタスクを実行することもできます。

タスク	説明
グラフの拡大	グラフ上にカーソルを移動し、拡大鏡アイコンをクリックします。
拡大されたグラフの一部を拡大	カーソルをドラッグして拡大する領域を選択します。
スループットグラフの行とバイトの切り替え	[バイト] オプションまたは [行] オプションをクリックします。
スループットグラフにプロットする統計の選択	[スループット] フィールドで、表示するソースとターゲットを選択します。

マッピングジョブのログの表示

ジョブ詳細を表示するジョブのログをダウンロードすることができます。

1. [実行統計] ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. [アドホックジョブ] フォルダを選択するか、アプリケーションを展開して [デプロイ済みのマッピングジョブ] を選択します。

ジョブのリストが [コンテンツ] パネルに表示されます。

4. [コンテンツ] パネルでジョブを選択します。

5. [アクション] > [選択したオブジェクトのログの表示] をクリックします。

ログファイルを開く、または保存するオプションのついたダイアログボックスが表示されます。

デプロイ済みのマッピングジョブの再発行

マッピングジョブが失敗した場合は、デプロイ済みのマッピングジョブの再発行ができます。デプロイ済みのマッピングジョブを再発行する場合は、データ統合サービスでジョブが再度実行されます。

1. [実行統計] ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開して、[デプロイ済みのマッピングジョブ] を選択します。
[コンテンツ] パネルにデプロイ済みのマッピングジョブのリストが表示されます。
4. デプロイ済みのマッピングジョブを選択します。

5. **【アクション】** > **【選択したジョブの再発行】** をクリックします。

マッピングジョブのキャンセル

アドホックまたはデプロイ済みのマッピングジョブを実行中にキャンセルできます。例えば、応答を停止したジョブ、または完了までの所要時間が極端に長いジョブをキャンセルすることができます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. **【アドホックジョブ】** フォルダを選択するか、アプリケーションを展開して **【デプロイ済みのマッピングジョブ】** を選択します。
ジョブのリストが **【コンテンツ】** パネルに表示されます。
4. **【コンテンツ】** パネルでジョブを選択します。
5. **【アクション】** > **【選択したジョブのキャンセル】** をクリックします。

拒否ファイル

マッピングの実行中、データ統合サービスはマッピングの中の各ターゲットインスタンスに対して拒否ファイルを作成します。データ統合サービスがターゲットに行を書き込むことができない場合、データ統合サービスは拒否された行を拒否ファイルに書き込みます。拒否ファイルおよびマッピングログには、その拒否の原因の判断に役立つ情報が含まれています。

拒否ファイルに拒否された行が含まれていない場合、データ統合サービスはマッピングの実行が終了したときに拒否ファイルを削除します。

マッピングを実行するたびに、データ統合サービスは拒否されたデータを拒否ファイルに追加します。問題の原因によっては、マッピングおよびターゲットデータベースを修正することで、それ以降のマッピングでの拒否の発生を防ぐことができます。

拒否ファイルの場所

データ統合サービスは、マッピング内のターゲットインスタンスごとに拒否ファイルを作成します。拒否ファイルはターゲットの拒否ファイルディレクトリに作成されます。

マッピング内のフラットファイルまたはリレーショナルなターゲットに関するランタイムプロパティで、ターゲットの拒否ファイルのディレクトリを設定します。デフォルトで、データ統合サービスは、RejectDir システムパラメータで定義されたディレクトリに、拒否ファイルを作成します。データ統合サービスは、拒否ファイルのファイル名をターゲットのインスタンス名に合わせて設定します。拒否ファイルのデフォルトのファイル名は<file_name>.bad です。

データ統合サービスが1つのターゲットに複数のパーティションを作成する場合、データ統合サービスはパーティションごとに別々の拒否ファイルを作成し、そのファイル名は<file_name><partition_number>.bad という形式になります。例えば、3つのパーティションが、MyOutput1.bad、MyOutput2.bad、MyOutput3.bad という名前の拒否ファイルに書き込みを行うことがあります。

拒否ファイルの内容

拒否ファイルが見つかったら、拒否ファイルの使用するコードページを表示できるテキストエディタでファイルを読むことができます。

拒否ファイルには、ライタまたはターゲットデータベースに拒否されたデータの行が入っています。データ統合サービスの書き込みは、拒否ファイル内の行全体に対して行われます。ただし、問題は通常その行の 1 カラムに集中します。拒否された行のどのカラムに原因があったかの判定に役立つように、拒否ファイルには各カラムについての詳しい情報を提供するインジケータが含まれています。

拒否ファイルに含まれるインジケータは以下のとおりです。

行インジケータ

拒否ファイルの各行の最初のカラムは、行インジケータです。行インジケータは、挿入、更新、または拒否のいずれのマークが行に付いていたかを示します。

カラムインジケータ

カラムインジケータは、カラムごとにデータの後に表示されます。カラムインジケータは、カラムに含まれているデータが有効か、オーバーフローしているか、NULL か、または切り詰められているかを示します。

行インジケータ

リジェクトファイルの最初のカラムは行インジケータです。行インジケータは、データ行の更新戦略を定義するフラグです。

以下の表に、拒否ファイルの行インジケータを示します。

行インジケータ	意味	拒否元
0	挿入	writer またはターゲット
1	更新	writer またはターゲット
2	削除	writer またはターゲット
3	拒否。アップデートストラテジの式によって拒否のマークが付いている。	Writer
4	ロールバックされた挿入	Writer
5	ロールバックされた更新	Writer
6	ロールバックされた削除	Writer
7	コミットされた挿入	Writer
8	コミットされた更新	Writer
9	コミットされた削除	Writer

次に示す拒否ファイルのサンプルでは、各行で行インジケータが 0 になっています。これはその行の挿入アップデートストラテジを表しています。

```
0,D,1921,D,Nelson,D,William,D,415-541-5145,D
0,D,1922,D,Page,D,Ian,D,415-541-5145,D
0,D,1923,D,Osborne,D,Lyle,D,415-541-5145,D
```

0,D,1928,D,De Souza,D,Leo,D,415-541-5145,D
0,D,2001123456789,0,S. MacDonald,D,Ira,D,415-541-514566,T

カラムインジケータ

カラムインジケータは、各データカラムのあとに表示されます。カラムインジケータは、データが有効であったか、オーバーフローしていたか、NULL であったか、または切り詰められていたかを示します。

以下の表に、拒否ファイルのカラムインジケータを示します。

カラムインジケータ	データの種類	Writer による扱い
D	有効なデータ。	正しいデータ。writer はデータをターゲットデータベースに渡します。ターゲットは、重複キーなどのデータベースエラーが起こらない限りデータを受け入れます。
N	NULL。カラムに NULL が格納されている。	正しいデータ。writer はデータをターゲットに渡し、ターゲットデータベースが NULL 値を受け入れない場合はターゲットがデータをリジェクトします。
T	切り詰め。文字列データが、カラムに指定されている精度の範囲を超えているため、値は切り詰められました。	マッピングターゲットがオーバーフローまたは切り詰めをリジェクトするように設定している場合は不良データ。

拒否ファイル内では、NULL カラムはカンマでマークして示されます。次の例に、良好なデータに囲まれた NULL カラムを示します。

0,D,5,D,,N,5,D

各行インジケータの後にカラムインジケータ「D」も表示されます。次の例では、行インジケータ「0」の後にカラムインジケータ「D」が表示されています。

0,D,2001123456789,0,S. MacDonald,D,Ira,D,415-541-514566,T

行は、writer またはターゲットデータベースによって拒否されることがあります。ログを調べて、拒否された原因を判定します。

第 9 章

PowerCenter へのエクスポート

この章では、以下の項目について説明します。

- [PowerCenter へのエクスポートの概要, 193 ページ](#)
- [PowerCenter のリリースの互換性, 194 ページ](#)
- [マプレットのエクスポート, 194 ページ](#)
- [パラメータを含むマッピングのエクスポート, 195 ページ](#)
- [PowerCenter へのエクスポートのオプション, 195 ページ](#)
- [PowerCenter へのオブジェクトのエクスポート, 197 ページ](#)
- [エクスポートの制限, 198 ページ](#)
- [PowerCenter へのエクスポートに関するルールおよびガイドライン, 199 ページ](#)
- [PowerCenter へのエクスポートのトラブルシューティング, 200 ページ](#)

PowerCenter へのエクスポートの概要

オブジェクトを PowerCenter で使用するために Developer ツールからエクスポートできます。

以下のオブジェクトをエクスポートできます。

- マッピング。マッピングを PowerCenter マッピングまたはマプレットにエクスポートします。
- マプレット。マプレットを PowerCenter マプレットにエクスポートします。
- 論理データオブジェクトモデル。論理データオブジェクトモデルを PowerCenter マプレットにエクスポートします。

オブジェクトは PowerCenter リポジトリまたは XML ファイルにエクスポートします。オブジェクトを XML ファイルにエクスポートした場合、PowerCenter ユーザーはそのファイルを PowerCenter リポジトリにインポートできます。

オブジェクトをエクスポートするときに、PowerCenter のリリース、マッピングとマプレットを変換する方法、参照テーブルをエクスポートするかどうかなどのエクスポートオプションを指定します。

パラメータを含むマッピングおよびマプレットをエクスポートできます。パラメータは、マッピングを PowerCenter リポジトリにインポートするときにデフォルト値に解決されます。

PowerCenter Express から PowerCenter にオブジェクトをエクスポートするには、PowerCenter Express クライアントをホストしているマシンに Metadata Converter をインストールする必要があります。

PowerCenter のリリースの互換性

オブジェクトが特定の PowerCenter のリリースとの互換性を持つことを検証するには、PowerCenter のリリースの互換性レベルを設定します。互換性レベルは、Developer ツールで表示できるすべてのマッピング、マプレット、および論理データオブジェクトモデルに適用されます。

特定のリリースの PowerCenter と照らし合わせて検証を行うように Developer ツールを設定するか、リリースの互換性の検証をスキップするように Developer ツールを設定することができます。デフォルトでは、Developer ツールは PowerCenter のリリースと照らし合わせたオブジェクトの検証を行いません。

オブジェクトを PowerCenter にエクスポートする前に、互換性レベルを PowerCenter のリリースに設定します。互換性レベルを設定すると、マッピング、マプレット、または論理データオブジェクトモデルを検証するときに 2 つの検証チェックが実行されます。まず、オブジェクトが Developer ツールで有効であることが検証されます。オブジェクトが有効な場合は、オブジェクトが選択したリリースの PowerCenter へのエクスポートに有効であることが検証されます。【検証ログ】ビューで互換性エラーを表示できます。

互換性レベルの設定

互換性レベルを設定して、PowerCenter のリリースと照らし合わせてマッピング、マプレット、および論理データオブジェクトモデルを検証します。[なし] を選択した場合は、オブジェクトの検証時にリリースの互換性の検証がスキップされます。

1. 【編集】 > 【互換性レベル】 をクリックします。
2. 互換性レベルを選択します。

メニューの選択した互換性レベルの横にドットが表示されます。互換性レベルは、Developer ツールで表示できるすべてのマッピング、マプレット、および論理データオブジェクトモデルに適用されます。

マプレットのエクスポート

マプレットをエクスポートするとき、またはマッピングをマプレットとしてエクスポートするときに、エクスポートプロセスによってマプレット内のオブジェクトが作成されます。また、いくつかのマプレットオブジェクトの名前が変更されます。

エクスポートプロセスでは、以下のマプレットオブジェクトがエクスポート XML ファイルに作成されます。

式トランスフォーメーション

エクスポートプロセスによって、マプレットの各入力トランスフォーメーションのすぐ下と各出力トランスフォーメーションのすぐ上に式トランスフォーメーションが作成されます。式トランスフォーメーションには次のような名前が付けられます。

Expr_<InputOrOutputTransformationName>

式トランスフォーメーションには、パススルーポートが含まれます。

出力トランスフォーメーション

マプレットをエクスポートしてターゲットを出力トランスフォーメーションに変換する場合は、ターゲットごとに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には次のような名前が付けられます。

<MappletInstanceName>_<TargetName>

エクスポートプロセスでは、以下のマプレットオブジェクトの名前がエクスポート XML ファイルで変更されます。

マプレットの入力トランスフォーメーションおよび出力トランスフォーメーション

マプレットの入力トランスフォーメーションおよび出力トランスフォーメーションには次のような名前が付けられます。

<TransformationName>_<InputOrOutputGroupName>

マプレットポート

マプレットポートの名前は次のように変更されます。

<PortName>_<GroupName>

パラメータを含むマッピングのエクスポート

パラメータを含むマッピングやマプレットをエクスポートして、PowerCenter にインポートできます。

パラメータを含むマッピングやマプレットをエクスポートする場合、パラメータは PowerCenter にインポートするときにデフォルト値に解決されます。インポートでは、パラメータを含むすべての SQL 式を解決できません。

システムパラメータは、対応する PowerCenter システムパラメータに解決されます。PowerCenter に対応するシステムパラメータがない場合、システムパラメータ参照は、PowerCenter にインポートした後もマッピング内に残ります。マッピングを編集し、参照を変更する必要があります。

マッピング出力を PowerCenter にエクスポートすることはできません。マッピングにマッピング出力が含まれている場合、マッピングをインポートしても、PowerCenter では無効になります。

PowerCenter へのエクスポートのオプション

オブジェクトを PowerCenter で使用するためにエクスポートするときは、エクスポートオプションを指定する必要があります。

以下の表に、エクスポートのオプションを示します。

オプション	説明
プロジェクト	オブジェクトのエクスポート元のモデルリポジトリ内のプロジェクト。
ターゲットリリース	PowerCenter のリリースバージョン。
選択したオブジェクトをファイルにエクスポート	PowerCenter XML ファイルにオブジェクトをエクスポートします。このオプションを選択する場合は、エクスポート XML ファイルの名前と場所を指定します。

オプション	説明
選択したオブジェクトを PowerCenter リポジトリにエクスポート	<p>PowerCenter リポジトリにオブジェクトをエクスポートします。このオプションを選択する場合は、PowerCenter リポジトリに関する以下の接続の詳細を指定します。</p> <ul style="list-style-type: none"> - ホスト名。PowerCenter ドメインゲートウェイのホスト名です。 - ポート番号。PowerCenter ドメインゲートウェイの HTTP ポート番号。 - 認証タイプ。次のうちいずれかの値を選択します: Kerberos シングルサインオン、ネイティブ、または LDAP。 - 認証タイプ。次のうちいずれかの値を選択します: ネイティブまたは LDAP。 - ユーザー名。リポジトリユーザー名です。 - パスワード。リポジトリユーザー名のパスワード。 <p>注: 認証タイプがネイティブまたは LDAP の場合は、ユーザー名とパスワードを指定します。</p> <ul style="list-style-type: none"> - セキュリティドメイン。認証タイプが LDAP の場合は、LDAP セキュリティドメイン名を指定します。存在しない場合は、「Native」と入力します。 - リポジトリ名。PowerCenter リポジトリ名です。
リポジトリフォルダーに送る	PowerCenter リポジトリ内の指定したフォルダーにオブジェクトをエクスポートします。
制御ファイルを使用	指定した <i>pmrep</i> 制御ファイルを使用して PowerCenter リポジトリにオブジェクトをエクスポートします。
エクスポートしたマッピングを PowerCenter マプレットに変換	Developer ツールのマッピングを PowerCenter マプレットに変換します。Developer ツールにより、マッピングのソースおよびターゲットとして使用されるデータオブジェクトが PowerCenter マプレットの入力トランスフォーメーションおよび出力トランスフォーメーションに変換されます。
ターゲットマプレットの変換	マプレットのターゲットとして使用されるデータオブジェクトを PowerCenter マプレットの出力トランスフォーメーションに変換します。PowerCenter マプレットにターゲットを含めることはできません。エクスポートオブジェクトにターゲットを含むマプレットが含まれる場合にこのオプションを選択しないと、エクスポートプロセスは失敗します。
参照データのエクスポート	エクスポートするオブジェクトのトランスフォーメーションで使用する参照テーブルデータをエクスポートします。
参照データの場所	Developer ツールがエクスポートする参照テーブルデータの場所。参照テーブルデータは 1 つ以上のディクショナリファイルとしてエクスポートされます。Developer ツールをホストするマシンで、ディレクトリへのパスを入力します。
コードページ	PowerCenter リポジトリのコードページ。

PowerCenter へのオブジェクトのエクスポート

PowerCenter にマッピング、マップレット、または論理データオブジェクトモデルをエクスポートするときに、オブジェクトをファイルまたは PowerCenter リポジトリにエクスポートできます。

オブジェクトをエクスポートする前に、互換性レベルを適切な PowerCenter のリリースに設定します。オブジェクトがその PowerCenter のリリースとの互換性を持つことを検証します。

1. **【ファイル】** > **【エクスポート】** をクリックします。
【エクスポート】 ダイアログボックスが表示されます。
2. **【Informatica】** > **【PowerCenter】** を選択します。
3. **【次へ】** をクリックします。
【エクスポート先 PowerCenter】 ダイアログボックスが表示されます。
4. オブジェクトのエクスポート元のモデルリポジトリ内のプロジェクトを選択します。
5. オブジェクトのエクスポート先の PowerCenter リリースを選択します。
6. オブジェクトのエクスポート先を選択します。オブジェクトは、PowerCenter リポジトリの XML ファイルにエクスポートできます。
 - オブジェクトをファイルにエクスポートするには、XML ファイルの名前と場所を指定します。
 - オブジェクトを PowerCenter リポジトリにエクスポートするには、**【参照】** をクリックしてリポジトリへの接続の詳細を指定します。
7. PowerCenter リポジトリにエクスポートする場合は、PowerCenter リポジトリ内のフォルダーか、PowerCenter へのオブジェクトのインポート方法を定義する *pmrep* 制御ファイルを選択します。
8. **【エクスポートしたマッピングを PowerCenter マップレットに変換】** を選択し、PowerCenter で Developer ツールのマッピングをマップレットに変換します。
9. **【ターゲットマップレットの変換】** を選択し、マップレットでターゲットとして使用されるデータオブジェクトを PowerCenter マップレットの出力トランスフォーメーションに変換します。
10. **【参照データのエクスポート】** を選択し、エクスポートするオブジェクトのトランスフォーメーションで使用する参照テーブルデータをエクスポートします。
11. 参照データをエクスポートする場合は、Developer ツールがエクスポートする参照テーブルデータの場所を指定します。
12. PowerCenter リポジトリのコードページを選択します。
13. **【次へ】** をクリックします。
エクスポートするオブジェクトを選択するように求められます。
14. エクスポートするオブジェクトを選択して、**【完了】** をクリックします。
オブジェクトが選択した場所にエクスポートされます。

オブジェクトをファイルにエクスポートする場合は、そのファイルから PowerCenter リポジトリにオブジェクトをインポートできます。

参照テーブルデータをエクスポートする場合は、Informatica サービスをホストしているマシンの PowerCenter のディレクトリ構造に参照データファイルをコピーします。参照データファイルの場所は、モデルリポジトリ内の参照テーブルオブジェクトの場所に対応している必要があります。

例えば、参照データファイルを以下の場所にコピーします。

<PowerCenter インストールディレクトリ>\services\<モデルリポジトリプロジェクト名>\<フォルダー名>

エクスポートの制限

モデルリポジトリオブジェクトを PowerCenter にエクスポートすると、一部のモデルリポジトリオブジェクトが PowerCenter リポジトリにエクスポートされない場合があります。PowerCenter で有効でない任意のオブジェクトを含むマッピングやマップレットをエクスポートすることはできません。

以下のオブジェクトは PowerCenter にエクスポートできません。

名前が長いオブジェクト

オブジェクト名が 80 文字を超える場合、PowerCenter ユーザーはマッピング、マップレット、またはマッピングかマップレット内のオブジェクトをインポートできません。

動的ポートが含まれているマッピングまたはマップレット

動的ポートが含まれているマッピングまたはマップレットはエクスポートできません。

データプロセッサトランスフォーメーションが含まれているマッピングまたはマップレット

データプロセッサトランスフォーメーションが含まれているマッピングまたはマップレットはエクスポートできません。

システムパラメータを使用しているディシジョントランスフォーメーションを含むマッピングまたはマップレット

ディシジョントランスフォーメーションのトランスフォーメーションスクリプトにシステムパラメータが含まれている場合、そのトランスフォーメーションを含むマッピングまたはマップレットをエクスポートすることはできません。エクスポート操作では、システムパラメータを PowerCenter が使用できる値に変換することはできません。システムパラメータを使用しているディシジョントランスフォーメーションを含むマッピングまたはマップレットをエクスポートするには、事前にそれらのシステムパラメータを適切な値で置換してください。

マッピング出力を返すマッピングまたはマップレット

マッピングまたはマップレットでマッピング出力が返される場合、PowerCenter ユーザーはマッピングまたはマップレットをインポートできません。

特定の結合条件を持つジョイナトランスフォーメーションが含まれているマッピングまたはマップレット

PowerCenter で有効でない結合条件を持つジョイナトランスフォーメーションが含まれているマッピングおよびマップレットはエクスポートできません。PowerCenter では、ユーザーはマスタソースと明細ソースが等しいことに基づいて結合条件を定義します。Developer ツールでは、その他の結合条件を定義できます。例えば、マスタソースと明細ソースが等しいことまたは等しくないことに基づいて結合条件を定義できます。トランスフォーメーション式が含まれている結合条件を定義できます。また、ジョイナトランスフォーメーションで交差結合が実行されるようにする $1=1$ などの結合条件も定義できます。

これらのタイプの結合条件は、PowerCenter では無効です。したがって、これらのタイプの結合条件を持つジョイナトランスフォーメーションが含まれているマッピングまたはマップレットを PowerCenter にエクスポートすることはできません。

ポート名が変更されたルックアップトランスフォーメーションが含まれているマッピングまたはマップレット

PowerCenter 統合サービスは、トランスフォーメーションのルックアップポートおよびルックアップ条件に基づいて、ルックアップソースに対してクエリを実行します。したがって、ルックアップトランスフォーメーションのポート名はルックアップソースのカラム名と一致している必要があります。

特定のカスタム SQL クエリを使用するルックアップトランスフォーメーションが含まれているマッピングまたはマップレット

Developer ツールでは、ルックアップトランスフォーメーション内の SQL クエリ構文を検証するために、PowerCenter とは異なるルールを使用します。Developer ツールで作成されるカスタム SQL クエリで AS キーワードまたは計算済みフィールドを使用する場合、そのクエリは PowerCenter で有効ではありません。SQL クエリを使用するルックアップトランスフォーメーションがマッピングまたはマップレットに含

まれていて、そのクエリで AS キーワードまたは計算済みフィールドが使用されている場合、そのマッピングまたはマプレットを PowerCenter にエクスポートすることはできません。

PowerCenter で使用できないソースが含まれているマッピングまたはマプレット

PowerCenter で使用できないソースが含まれているマッピングまたはマプレットをエクスポートする場合、マッピングまたはマプレットのエクスポートは失敗します。

以下のソースを持つマッピングまたはマプレットはエクスポートできません。

- 複合型ファイルデータオブジェクト
- DataSift
- Web コンテンツ-Kapow Katalyst

ポートを連結するマプレット

複数グループ入力トランスフォーメーションが含まれているマプレットをエクスポートする場合に、異なる入力グループのポートが同じダウストリームトランスフォーメーションに接続されていると、エクスポートプロセスは失敗します。

未接続のルックアップトランスフォーメーションを持つネストされたマプレット

未接続のルックアップトランスフォーメーションを持つ別のマプレットが含まれているあらゆるタイプのマッピングまたはマプレットをエクスポートする場合、エクスポートプロセスは失敗します。

SAP ソースが含まれているマッピング

SAP ソースが含まれているマッピングをエクスポートすると、SAP ソースなしでマッピングがエクスポートされます。このマッピングを PowerCenter リポジトリにインポートすると、PowerCenter クライアントによってマッピングがソースなしでインポートされます。出力ウィンドウに、マッピングが無効であることを伝えるメッセージが表示されます。PowerCenter で SAP ソースを手動で作成してマッピングに追加する必要があります。

Timestamp with Time Zone または Timestamp with Local Time Zone を含むマッピング

Timestamp with Time Zone または Timestamp with Local Time Zone データ型のデータを含むマッピングを Developer tool からインポートすると、PowerCenter Client がマッピングの変換に失敗します。

PowerCenter へのエクスポートに関するルールおよびガイドライン

Developer ツールと PowerCenter の違いにより、Developer ツールのオブジェクトの中には PowerCenter との互換性がないものもあります。

PowerCenter にオブジェクトをエクスポートする場合には、以下のルールおよびガイドラインに従います。

PowerCenter のリリースを確認する。

Developer ツールからエクスポートするオブジェクトがターゲットの PowerCenter リリースと互換性があることを確認します。

オブジェクト名が一意であることを確認する。

オブジェクトを PowerCenter リポジトリにエクスポートする場合、PowerCenter オブジェクトの名前がエクスポートされるオブジェクトと同じときは、PowerCenter オブジェクトが上書きされます。

コードページに互換性があることを確認する。

Developer ツールと PowerCenter で互換性のないコードページが使用されている場合、エクスポートプロセスは失敗します。

精度モードを確認する。

デフォルトでは、Developer ツールは高精度が有効になっている状態でマッピングおよびマプレットを実行しますが、PowerCenter は高精度が無効になっている状態でセッションを実行します。Developer ツールのマッピングと PowerCenter のセッションを異なる精度モードで実行すると、異なる結果になる場合があります。異なる結果にならないようにするには、オブジェクトを同じ精度モードで実行します。

参照データをコピーする。

参照テーブルを使用するトランスフォーメーションとともにマッピングまたはマプレットをエクスポートする場合、その参照テーブルを、PowerCenter 統合サービスがアクセスできるディレクトリにコピーする必要があります。参照テーブルは、INFA_CONTENT 環境変数で定義されたディレクトリにコピーします。INFA_CONTENT が設定されていない場合は、参照テーブルを次の PowerCenter サービスディレクトリにコピーします。

```
$INFA_HOME\services\<Developer ツールのプロジェクト名>\<Developer ツールのフォルダー名>
```

PowerCenter へのエクスポートのトラブルシューティング

名前が長いオブジェクトが含まれているマプレットをエクスポートするときにエクスポートプロセスが失敗する。

マプレットをエクスポートするとき、またはマッピングをマプレットとしてエクスポートするときに、エクスポートプロセスによってマプレット内のいくつかのオブジェクトが作成または名前変更されます。エクスポートプロセスでは、式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エクスポート XML ファイルに作成されます。また、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、およびマプレットポートの名前が変更されます。

式トランスフォーメーションの名前を生成するために、入力トランスフォーメーションおよび出力トランスフォーメーションの名前に文字が付加されます。マプレットをエクスポートしてターゲットを出力トランスフォーメーションに変換する場合は、マプレットのインスタンス名とターゲット名を組み合わせで出力トランスフォーメーション名が生成されます。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、およびマプレットポートの名前が変更されるとき、オブジェクト名にグループ名が付加されます。

既存のオブジェクトの名前が長い場合は、エクスポートされるオブジェクトの名前が、エクスポート XML ファイルまたは PowerCenter リポジトリにおけるオブジェクト名の制限である 80 文字を超えることがあります。オブジェクト名が 80 文字を超えると、内部エラーが発生してエクスポートプロセスが失敗します。

マプレットをエクスポートする場合に内部エラーが返されたときは、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、ターゲット、およびポートの名前を確認します。名前が長い場合は短くします。

第 10 章

PowerCenter からのインポート

この章では、以下の項目について説明します。

- [PowerCenter からのインポートの概要, 201 ページ](#)
- [オーバーライドプロパティ, 202 ページ](#)
- [競合の解決, 202 ページ](#)
- [インポートサマリ, 202 ページ](#)
- [データ型変換, 203 ページ](#)
- [トランスフォーメーションの変換, 203 ページ](#)
- [パラメータを含むマッピングのインポート, 211 ページ](#)
- [PowerCenter リポジトリ接続パラメータ, 212 ページ](#)
- [接続の割り当て, 213 ページ](#)
- [PowerCenter からのオブジェクトのインポート, 214 ページ](#)
- [インポートの制限, 215 ページ](#)
- [インポートパフォーマンス, 216 ページ](#)

PowerCenter からのインポートの概要

PowerCenter リポジトリからモデルリポジトリにオブジェクトをインポートすることができます。インポートプロセスでは、PowerCenter リポジトリオブジェクトが検証されてモデルリポジトリオブジェクトに変換された後、インポートが実行されます。

PowerCenter からオブジェクトをインポートする場合は、インポートするオブジェクトおよびモデルリポジトリ内のターゲット場所を選択します。インポートプロセスには、インポート中にオブジェクト名の競合を解決するためのオプションがあります。

また、モデルリポジトリの接続を PowerCenter のオブジェクトに割り当てることもできます。複数の PowerCenter オブジェクトに同時に単一の接続を割り当てることができます。

パラメータを含むマッピングをインポートできます。再利用可能なトランスフォーメーションを含むマッピングをインポートする場合、インポートプロセスで PowerCenter マッピングパラメータがインポートされ、そのパラメータのバインド先となるトランスフォーメーションレベルのパラメータが生成されます。マッピングに再利用不可のトランスフォーメーションが含まれている場合、入力プロセスでマッピングレベルのパラメータが作成されます。

インポートプロセスの完了後、インポートサマリを表示できます。

PowerCenter から PowerCenter Express にオブジェクトをインポートするには、PowerCenter Express クライアントをホストするマシンに Metadata Converter をインストールする必要があります。

オーバーライドプロパティ

インポートプロセス中に PowerCenter オブジェクトのオーバーライドプロパティを保持するか無視するかを選択できます。デフォルトでは、インポートプロセスは PowerCenter オブジェクトのオーバーライドプロパティを保持します。

オーバーライドプロパティを保持すると、PowerCenter オブジェクトの再利用不可能なトランスフォーメーションまたは再利用可能なデータオブジェクトがインポートプロセスで作成されます。PowerCenter マッピングがソースおよびターゲットプロパティをオーバーライドすると、PowerCenter マッピングと同じオーバーライドプロパティ値を持つデータオブジェクトが作成されます。インポートプロセスは PowerCenter オブジェクトの名前に番号を付与して、データオブジェクトが作成します。

競合の解決

PowerCenter からオブジェクトをインポートするときにモデルリポジトリ内に同じ名前のオブジェクトが存在する場合の、オブジェクト名の競合を解決することができます。

以下の競合解決オプションから選択できます。

ターゲットのオブジェクトの名前を変更する

デフォルトの命名規則で PowerCenter リポジトリオブジェクトの名前を変更し、インポートします。オブジェクトの名前を変更するオプションは、デフォルトの競合解決オプションです。

ターゲットのオブジェクトを上書きする

モデルリポジトリオブジェクトのオブジェクトを PowerCenter リポジトリのオブジェクトで置き換えます。

ターゲットのオブジェクトを再利用する

モデルリポジトリのオブジェクトをマッピングで再利用します。

重要: モデルリポジトリでは、競合を解決するためにマッピングとマップレットを区別することはありません。例えば、インポートしたマップレットと同じ名前のマッピングがリポジトリに含まれる場合、競合を解決するように要求されます。オブジェクトを置き換えることを選択した場合、インポートプロセスによってマッピングがマップレットで置き換えられます。

インポートサマリ

PowerCenter オブジェクトをモデルリポジトリにインポートした後、インポートプロセスによってインポートサマリが作成されます。

変換エラーがある場合は、インポートサマリをファイルに保存できます。インポートサマリには、インポートのステータス、変換されなかったオブジェクトの数、インポート後に無効となったオブジェクトの数、変換エラーが含まれます。インポート後のオブジェクトを Developer ツールで検証して、検証エラーを表示することもできます。

データ型変換

一部の PowerCenter データ型はモデルリポジトリでは無効です。無効なデータ型の PowerCenter オブジェクトをインポートすると、モデルリポジトリの有効な対応するデータ型に変換されます。

以下の表に、インポートプロセスで、対応するモデルリポジトリデータに変換される PowerCenter リポジトリデータ型を示します。

PowerCenter リポジトリデータ型	モデルリポジトリデータ型
Real	Double
Small Int	Integer
Nstring	String
Ntext	Text

トランスフォーメーションの変換

PowerCenter トランスフォーメーションは、互換性に基づいて変換されます。トランスフォーメーションの中にはモデルリポジトリと互換ではないものがあります。制約付きのその他のインポート。

制約付きでインポートするか、またはインポートに失敗した PowerCenter トランスフォーメーションについて、以下の表で説明します。

PowerCenter トランスフォーメーション	インポートアクション
アグリゲータ	制限付きでインポートされる。
データマスキング	インポートが失敗する。
エクスターナルプロシージャ	インポートが失敗する。
HTTP	インポートが失敗する。
ID 解決	インポートが失敗する。
Java	制限付きでインポートされる。
ジョイナ	制限付きでインポートされる。
ルックアップ	制限付きでインポートされる。
ノーマライザ	制限付きでインポートされる。
ランク	制限付きでインポートされる。
シーケンスジェネレータ	制限付きでインポートされる。
ソータ	制限付きでインポートされる。

PowerCenter トランスフォーメーション	インポートアクション
ソース修飾子	制限付きでインポートされる。ソースおよびソース修飾子トランスフォーメーションが 1 つのデータオブジェクトとして完全にインポートされる。
ストアドプロシージャ	インポートが失敗する。
トランザクションコントロール	インポートが失敗する。
SQL	制限付きでインポートされる。
共有体	制限付きでインポートされる。
構造化されていないデータ	インポートが失敗する。
XML パーサー	インポートが失敗する。
XML ジェネレータ	インポートが失敗する。

トランスフォーメーションのプロパティの制限

一部の PowerCenter 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プロパティに基づき、制限付きでインポートされます。

インポートプロセスは、特定のトランスフォーメーションプロパティの互換性に基づいて、以下のいずれかのアクションを実行する場合があります。

- 無視。トランスフォーメーションプロパティを無視して、オブジェクトをインポートします。
- 内部的に変換。オブジェクトをトランスフォーメーションプロパティとともにインポートしますが、プロパティは Developer ツールで公開されません。
- インポートの失敗。オブジェクトのインポートに失敗し、マッピングは無効になります。

アグリゲータトランスフォーメーション

以下の表に、アグリゲータトランスフォーメーションのプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
トランスフォーメーション範囲	無視。

Java トランスフォーメーション

Java トランスフォーメーションでは、ポートは入力ポートまたは出力ポートのどちらかである必要があります。Java トランスフォーメーションに入力ポートと出力ポート両方がある場合、インポートは失敗します。

以下の表に、Java トランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
クラス名	無視。
関数識別子	無視。
トランザクションの生成	無視。
入力ブロック	無視。
パーティション化可能	無視。
言語	無視。
モジュール識別子	無視。
出力は確定的	無視。
出力が再現可能	無視。
パーティションごとに 1 つのスレッドを要求します	無視。
実行時位置	無視。
アップデートストラテジトランスフォーメーション	無視。

ジョイナトランスフォーメーション

以下の表に、ジョイナトランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
マスタでの NULL の順序付け	内部的に変換。
NULL の詳細な順序付け	内部的に変換。
トランスフォーメーション範囲	内部的に変換。

ルックアップトランスフォーメーション

以下の表に、ルックアップトランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
キャッシュファイル名プレフィックス	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップキャッシュの初期化	無視。
ルックアップキャッシュディレクトリ名	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップキャッシュが有効	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップデータのキャッシュサイズ	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップのインデックスキャッシュサイズ	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップソースは静的です	無視。
ルックアップ SQL オーバライド	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はカスタム SQL クエリにインポート。
ルックアップソースフィルタ	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップキャッシュの事前作成	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
ルックアップソースからの再キャッシュ	スタンドアロントランスフォーメーションとして変換された場合は無視、マッピング内で変換された場合はインポート。
古い場合に再キャッシュ	無視。
サブ秒の精度	無視。
動的キャッシュの同期	無視。

ノーマライザトランスフォーメーション

ノーマライザトランスフォーメーションを Developer tool にインポートする場合、1つの入力グループおよび少なくとも1つの出力グループを持つノーマライザトランスフォーメーションをインポートします。

マッピングの一部ではないノーマライザトランスフォーメーションをインポートする場合、Developer tool はノーマライザトランスフォーメーションの入力グループにすべての入力ポートを置きます。Developer tool は出力ポートのノーマライザトランスフォーメーションルールに基づいてデフォルトの出力グループを作成します。インポートするノーマライザトランスフォーメーションに出力ポートがない場合、Developer tool はインポート済みのノーマライザトランスフォーメーションにデフォルトの出力グループを作成します。

ノーマライザトランスフォーメーションがマッピングの一部である場合、Developer tool はマッピングのダウンストリームトランスフォーメーションまたはターゲットへのリンクに基づいて複数の出力グループを作成することがあります。複数グループのトランスフォーメーションからターゲットへのリンクに関するルールおよびガイドラインの詳細については、『Developer トランスフォーメーションガイド』を参照してください。

再利用可能なノーマライザトランスフォーメーションを含むマッピングをインポートする場合、Developer tool はトランスフォーメーションを再利用可能としてインポートします。また Developer tool は、マッピング内の再利用可能なノーマライザトランスフォーメーションインスタンスを再利用不可能なトランスフォーメーションインスタンスで置換します。Developer tool は再利用不可能なノーマライザトランスフォーメーションからダウンストリームトランスフォーメーションおよびターゲットへの新しいリンクを生成します。

PowerCenter で、ノーマライザトランスフォーメーションにキーポートが少なくとも 1 つ生成されます。Developer tool では、ノーマライザトランスフォーメーションに生成されたキーポートは含まれません。ノーマライザトランスフォーメーションを PowerCenter からインポートする場合、Developer tool は生成されたキーポートを無視します。

以下の表に、ノーマライザトランスフォーメーションのプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
リセット	無視。
リスタート	無視。

リンクトランスフォーメーション

以下の表に、リンクトランスフォーメーションプロパティのインポートアクションについて示します。

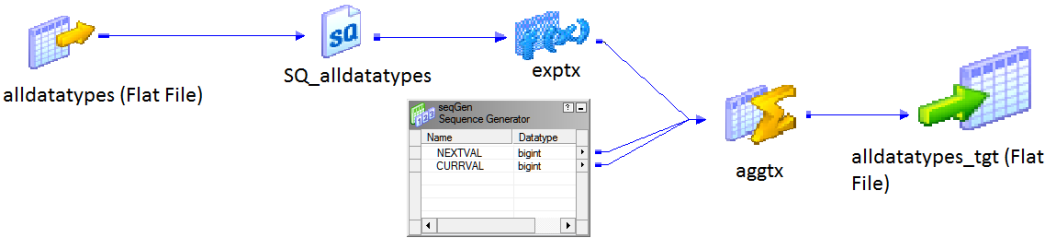
トランスフォーメーションのプロパティ	インポートアクション
トランスフォーメーション範囲	無視。

シーケンスジェネレータトランスフォーメーション

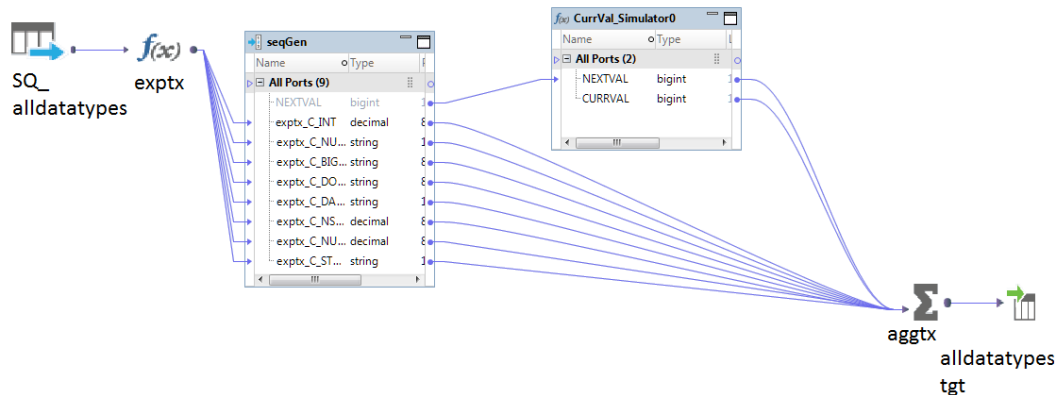
PowerCenter のシーケンスジェネレータトランスフォーメーションには 2 つのポート、CURRVAL および NEXTVAL があります。Developer tool では、シーケンスジェネレータトランスフォーメーションには出力ポート NEXTVAL のみがあります。マッピング内のシーケンスジェネレータトランスフォーメーションをインポートする場合、Developer tool は式トランスフォーメーションを作成して CURRVAL ポートの値を設定し、ダウンストリームトランスフォーメーションに渡します。

例えば、PowerCenter にシーケンスジェネレータ、アグリゲータ、および式トランスフォーメーションを持つマッピングがあるとします。ソースデータは、式トランスフォーメーション、アグリゲータトランスフォーメーション、ターゲットの順序で渡されます。シーケンスジェネレータトランスフォーメーションは、シーケンス番号を CURRVAL ポートを持つ各行に追加します。

次の図に、シーケンスジェネレータトランスフォーメーション、アグリゲータトランスフォーメーション、および式トランスフォーメーションを持つ PowerCenter マッピングを示します。



次の図に、Developer tool にインポートされたマッピングを示します。



マッピングをインポートする場合、Developer tool は NEXTVAL データをシーケンスジェネレータトランスフォーメーションから式トランスフォーメーションに渡して CURRVAL 値を設定します。

Developer tool は、マッピング内の再利用可能なシーケンスジェネレータトランスフォーメーションを再利用可能なシーケンスデータオブジェクトを持つ再利用不可能なトランスフォーメーションとしてインポートします。マッピングの一部ではない再利用可能なシーケンスジェネレータトランスフォーメーションをインポートする場合、Developer tool は再利用可能なシーケンスデータオブジェクトを作成します。

以下の表に、シーケンスジェネレータトランスフォーメーションのプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
現在の値	無視。
キャッシュされる値の数	無視。

ソータートランスフォーメーション

以下の表に、ソータートランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
トランスフォーメーション範囲	無視。

ソース修飾子トランスフォーメーション

ソース修飾子トランスフォーメーションプロパティへのインポートアクションを以下の表に示します。

トランスフォーメーションのプロパティ	インポートアクション
ソートするポート数	無視。

SQL トランスフォーメーション

以下の表に、SQL トランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
自動コミット	無視。
クラス名	無視。
接続タイプ	動的接続オブジェクトまたは完全な動的接続情報に設定されている場合、インポートに失敗する。
データベースタイプ	Sybase、Informix、または Teradata のインポートに失敗する。
関数識別子	無視。
トランザクションの生成	無視。
入力ブロック	無視。
パーティション化可能	無視。
言語	無視。
プール内の最大接続数	無視。
モジュール識別子	無視。
出力は確定的	無視。
出力が再現可能	無視。

トランスフォーメーションのプロパティ	インポートアクション
パーティションごとに1つのスレッドを要求します	無視。
実行時位置	無視。
SQL モード	スクリプトモードのインポートは失敗する。
トランスフォーメーション範囲	無視。
DB 接続エラーを重大として扱う	内部的に変換。
アップデートストラテジトランスフォーメーション	無視。
接続プールの使用	無視。

共有体トランスフォーメーション

以下の表に、共有体トランスフォーメーションプロパティのインポートアクションについて示します。

トランスフォーメーションのプロパティ	インポートアクション
クラス名	無視。
関数識別子	無視。
トランザクションの生成	無視。
入力ブロック	無視。
パーティション化可能	無視。
言語	無視。
モジュール識別子	無視。
出力は確定的	無視。
出力が再現可能	無視。
パーティションごとに1つのスレッドを要求します	無視。
実行時位置	無視。

トランスフォーメーションのプロパティ	インポートアクション
トランスフォーメーション範囲	無視。
アップデートストラテジトランスフォーメーション	無視。

パラメータを含むマッピングのインポート

パラメータを含む PowerCenter マッピングまたはマプレットをインポートできます。

PowerCenter マッピングまたはマプレットにパラメータが含まれている場合、インポートプロセスで PowerCenter マッピングパラメータと、そのパラメータを参照する再利用可能なトランスフォーメーション間にパラメータバインディングが作成されます。インポートプロセスでは、トランスフォーメーションレベルのパラメータが生成されます。

PowerCenter では、パラメータの初期値は不要です。初期値を指定しないで PowerCenter パラメータをインポートすると、インポートプロセスでパラメータのデータ型に基づいてパラメータのデフォルト値が割り当てられます。

変数が含まれているマッピングをインポートしても、インポートプロセスで変数は変換されません。マッピングの変換に成功することもあります。そのマッピングはモデルリポジトリでは無効になります。マッピングを変更して、モデルリポジトリの変数の代わりにパラメータを使用できます。

一部の PowerCenter マッピングパラメータでは、IsExprVar プロパティが有効になっています。このプロパティは、式を解析する前に PowerCenter 統合サービスでパラメータを展開する必要があることを示します。IsExprVar プロパティは、モデルリポジトリでは無効です。このパラメータプロパティが有効になっているマッピングをインポートする場合、変換には成功しますが、マッピングは無効になります。

パラメータ化されたソース所有者名プロパティが含まれる Netezza および Teradata オブジェクトをインポートした場合、インポートプロセスによってプロパティが変換されません。インポートプロセスでは、Teradata マッピング内のパラメータ化されたターゲットテーブル名も無視されます。

システム定義パラメータを含むマッピングのインポート

一部のシステム定義パラメータを含む PowerCenter マッピングまたはマプレットをインポートできます。パラメータがモデルリポジトリで有効なシステム定義パラメータと一致する場合、インポートプロセスでパラメータがインポートされます。

インポートプロセスでは、システム定義パラメータと、そのパラメータを参照する再利用可能なトランスフォーメーション間にパラメータバインディングが作成されます。

PowerCenter マッピングに、モデルリポジトリ内に相当するシステム定義パラメータがないシステム定義パラメータがある場合、変換は失敗しません。インポートプロセスで、そのパラメータ名のマッピングプロパティがプロパティ値としてコピーされます。ただし、インポートされたマッピングは有効ではありません。ユーザー定義パラメータを作成してプロパティ値を置き換えるか、マッピングロジックを変更することができます。

次のシステム定義パラメータをインポートできます。

- \$PMMappingName
- \$PMIntegrationServiceName

- \$PMRepositoryUserName
- \$SESSSTARTTIME
- \$SYSDATE

次の PowerCenter システム定義パラメータはインポートできません。

- \$PMFolderName
- \$PMRepositoryServiceName
- \$PMSessionName
- \$PMSessionRunMode
- \$PMTAB_ALL_DATA_TYPES@TableName
- \$PMTGT_ALL_DATA_TYPES@TableName
- \$PMWorkflowName
- \$PMWorkflowRunId
- \$PMWorkflowRunInstanceName

PowerCenter リポジトリ接続パラメータ

PowerCenter リポジトリからオブジェクトをインポートする際には、接続パラメータをリポジトリに指定する必要があります。Developer ツールはこのインポートパラメータを使用して、PowerCenter リポジトリに接続します。

次の表は、各インポートパラメータとその説明です。

パラメータ	説明
ホスト名	PowerCenter ドメインゲートウェイのホスト名です。
ポート番号	PowerCenter ドメインゲートウェイの HTTP ポート番号です。
リリース番号	PowerCenter のリリースバージョンです。
認証タイプ	PowerCenter リポジトリへの接続に必要なユーザー認証のタイプです。次のうちいずれかの値を選択します: Kerberos シングルサインオン、ネイティブ、または LDAP。 PowerCenter リポジトリへの接続に必要なユーザー認証のタイプです。次のうちいずれかの値を選択します: ネイティブまたは LDAP。 注: 認証タイプがネイティブまたは LDAP の場合は、ユーザー名とパスワードを指定します。
ユーザー名	PowerCenter リポジトリのユーザー名です。
パスワード	PowerCenter リポジトリのユーザー名のパスワードです。
セキュリティドメイン	認証タイプが LDAP の場合は、LDAP セキュリティドメイン名を指定します。それ以外の場合は、Native と入力します。

パラメータ	説明
リポジトリ名	PowerCenter リポジトリ名です。
コードページ	PowerCenter リポジトリのコードページです。

接続の割り当て

PowerCenter からデータソースおよびその他のオブジェクトをインポートする場合、データソース接続タイプを割り当てることができます。

例えば、Oracle データベースに接続する PowerCenter のソース定義、ターゲット定義、またはルックアップトランスフォーメーションを作成できます。Developer tool でこれらのオブジェクトをインポートする場合、**【接続の割り当て】** ダイアログボックスを使用して、各オブジェクトの接続タイプを指定します。

PowerCenter リポジトリオブジェクトの接続タイプを選択する場合、次のいずれかの方法を使用して、接続を割り当てます。

単一の接続を複数の PowerCenter オブジェクトに同時に割り当てます。

割り当てられた接続を持たないすべてのソース、すべてのターゲット、すべてのルックアップトランスフォーメーション、またはすべてのオブジェクトに単一の接続を割り当てることができます。または、単一の接続を、特定の名前のパターンと一致する名前を持つすべてのオブジェクトに割り当てることができます。**【選択】** リストからオプションを選択し、**【接続の割り当て】** をクリックします。

単一の接続を、複数の異なるオブジェクトタイプの PowerCenter オブジェクトに割り当てます。

【カスタム】 オプションを **【選択】** リストで選択し、複数の PowerCenter オブジェクトを選択して、**【接続を割り当て】** をクリックします。

PowerCenter オブジェクトに接続を割り当てます。

PowerCenter オブジェクトを選択し、**【接続名】** カラムの **【開く】** ボタンをクリックします。

元の接続タイプとは異なるオブジェクトに接続を割り当てることができます。この操作は、Developer tool で元の接続タイプがサポートされていない場合などに行います。元の接続タイプとは異なる接続タイプを割り当てると、Developer tool に接続タイプの不一致の警告が表示されます。

警告を無視して続行するように選択すると、インポートに成功し、インポートしたデータソースに新しい接続が割り当てられます。選択した接続のメタデータが、インポートしたデータソースのスキーマと一致している場合、ソースまたはターゲットが有効になります。

注: Developer tool でサポートされていない接続タイプで作成されたルックアップトランスフォーメーションの場合、元の接続タイプに関する情報が使用できないため、Developer tool に接続の不一致に関する警告が表示されません。

PowerCenter からのオブジェクトのインポート

PowerCenter リポジトリからモデルリポジトリにオブジェクトをインポートすることができます。

PowerCenter からオブジェクトをインポートする前に、ターゲットモデルリポジトリに接続します。

1. **【ファイル】 > 【インポート】** を選択します。

【インポート】ダイアログボックスが表示されます。

2. **【Informatica】 > 【PowerCenter】** を選択します。

3. **【次へ】** をクリックします。

【PowerCenter からのインポート】ダイアログボックスが表示されます。

4. PowerCenter リポジトリの接続パラメータを指定します。

5. **【テスト接続】** をクリックします。

PowerCenter リポジトリへの接続がテストされます。

6. PowerCenter リポジトリへの接続が成功したら、**【OK】** をクリックします。**【次へ】** をクリックします。

PowerCenter リポジトリのフォルダーが表示され、インポートするオブジェクトを選択するように求めるメッセージが表示されます。

7. インポートするオブジェクトを 1 つ以上選択します。

8. **【次へ】** をクリックします。

9. モデルリポジトリ内でオブジェクトのインポート先の場所を選択します。

10. オブジェクト名の競合に対する競合解決オプションを選択します。ターゲットモデルリポジトリで、オブジェクトの名前の変更、置換、または再利用を選択します。

- デフォルトの命名規則を使用している PowerCenter リポジトリオブジェクトの名前を変更してからモデルリポジトリにインポートするには、**【ターゲットのオブジェクトの名前を変更する】** オプションを選択します。オブジェクトの名前を変更するオプションは、デフォルトの競合解決オプションです。
- PowerCenter リポジトリオブジェクトを持つモデルリポジトリオブジェクトを置換するには、**【ターゲットのオブジェクトを上書きする】** オプションを選択します。
- PowerCenter オブジェクトをインポートするのではなく、マッピング内のモデルリポジトリのオブジェクトを再利用するには、**【ターゲットのオブジェクトを再利用する】** オプションを選択します。

11. **【次へ】** をクリックします。

PowerCenter オブジェクトと依存オブジェクトが表示されます。

12. 再利用可能な PowerCenter ソース、ターゲット、トランスフォーメーションのオーバーライドプロパティを無視するには、**【オーバーライドされたプロパティを無視する】** をクリックします。デフォルトでは、プロセスで上書きプロパティが保持されます。

13. IBM DB2 オブジェクトをインポートする場合は、DB2 オブジェクトタイプを選択します。以下のいずれかのオブジェクトタイプを選択できます。LOW、z/OS、i5/OS。

14. **【次へ】** をクリックします。

15. PowerCenter リポジトリオブジェクトに対するモデルリポジトリの接続の詳細を指定します。

16. **【接続の選択】** ダイアログボックスが開き接続を選択し、**【OK】** をクリックします。

17. **【次へ】** をクリックします。

Developer ツールによってインポートの概要が作成され、PowerCenter オブジェクトとインポートされる依存オブジェクトが一覧表示されます。

18. **【変換チェック】** をクリックし、オブジェクトが有効なモデルリポジトリオブジェクトとしてインポートできるかどうかを検証します。

変換チェックのサマリレポートに、変換チェックの結果が表示されます。

19. **【OK】** をクリックします。**【完了】** をクリックします。

インポート中、進捗情報が表示されます。PowerCenter オブジェクトおよび依存オブジェクトがモデルリポジトリにインポートされ、最終的なインポートサマリが生成されます。

20. 変換エラーがある場合、**【保存】** をクリックして、インポートサマリを保存するファイル名を指定します。

インポートの制限

PowerCenter オブジェクトをインポートするときには次の制限が適用されます。

ソースおよびターゲット

- PowerCenter の 9.1.0 以前のリリースからソースまたはターゲットをインポートする場合、インポートプロセスでは、オブジェクトに関連付けられている接続タイプが有効かを検証できません。
- PowerCenter リポジトリのバージョンが 9.5.0 以前の場合、IBM DB2 のソースデータベース名またはターゲット名の先頭には、DB2 タイプを設定するために "DB2" を付けます。
- フラットファイルソースの行の区切り文字が無効な場合、インポートプロセスによってデフォルト値に変更されます。
- PowerCenter から Teradata ソースをインポートした場合、インポートプロセスで次のプロパティが無視されます。
 - カラムの [非 NULL] プロパティ
 - 出力は確定的
 - 出力が再現可能
 - ソートするポート数
 - 外部キー
- PowerCenter から Teradata ターゲットをインポートした場合、インポートプロセスで次のプロパティが無視されます。
 - カラムの [非 NULL] プロパティ
 - 更新オーバーライド
 - ターゲットテーブルプレフィックス
 - 外部キー
- PowerCenter から Netezza ソースをインポートした場合、インポートプロセスで次のプロパティが無視されます。
 - 外部キー
 - 出力は確定的
 - 出力が再現可能
 - ソートするポート数
- PowerCenter から Netezza ターゲットをインポートした場合、インポートプロセスで次のプロパティが無視されます。
 - 外部キー
 - 更新オーバーライド

トランスフォーメーション

- トランスフォーメーションに含まれる式は、最大 4,000 文字とします。
- SQL トランスフォーメーションまたはルックアップトランスフォーメーションのデータベースタイプは、インポートプロセス中に ODBC に変換されます。
- トランスフォーメーションのデータキャッシュサイズまたはインデックスキャッシュサイズを無効な値に設定すると、インポートプロセスによってその値は Auto に変更されます。

マッピング

- マッピングに含めるパイプラインは 1 つだけにします。

インポートパフォーマンス

68MB を超えるマッピングをインポートする場合は、コマンドラインから行うと最適なパフォーマンスが得られます。

ヒント: 次のコマンドラインオプションを使用できます: ImportFromPC

第 11 章

パフォーマンスのチューニング

この章では、以下の項目について説明します。

- [パフォーマンスのチューニングの概要, 217 ページ](#)
- [最適化方式, 218 ページ](#)
- [最適化レベル, 222 ページ](#)
- [Developer ツールのマッピングに対する最適化レベルの設定, 223 ページ](#)
- [デプロイ済みのマッピングに対する最適化レベルの設定, 224 ページ](#)

パフォーマンスのチューニングの概要

データ統合サービスでは、マッピングを最適化してパフォーマンスを向上させます。

データ統合サービスは次の最適化を実行できます。

データをフィルタリングして、処理行数を削減する。

データ統合サービスは処理するデータ量を削減するため最適化方式を適用します。マッピングの実行時に最適化レベルを選択することができ、それによってデータ統合サービスでマッピングに適用できる最適化方式が決まります。例えば、データ統合サービスでは、初期選択最適化を使用してフィルタをソースに近づけることができます。また、プッシュダウンの最適化を使用してトランスフォーメーションロジックをデータベースにプッシュできます。コストベースの最適化方式を使用して結合の処理順序を変更できます。

データ統合サービスでは、マッピングに複数の最適化方式を同時に適用できます。例えば、ノーマル最適化レベルを選択すると、データ統合サービスは初期プロジェクション、述部最適化、初期選択、ブランチ刈り込み、またはプッシュイン最適化方式を適用します。

パーティション化戦略を判断して、並列処理を最大化する。

パーティション化オプションがある場合、データ統合サービスは、マッピングの並列処理を最大化できます。データ統合サービスはマッピングのパーティション化戦略を動的に判断します。パーティション化戦略には、パーティションポイントの場所、各パイプラインステージの最適なパーティション数、各パーティションポイントでデータを最適に再分散するパーティション化タイプが含まれます。パーティション化の詳細については、[「パーティション化されたマッピングの概要」 \(ページ 249\)](#)を参照してください。

データ統合サービスは処理するデータ量を削減するため最適化方式を適用します。マッピングの実行時に最適化レベルを選択することができ、それによってデータ統合サービスでマッピングに適用できる最適化方式が決まります。例えば、データ統合サービスでは、初期選択最適化を使用してフィルタをソースに近づけることができます。また、プッシュダウンの最適化を使用してトランスフォーメーションロジックをデータベースにプッシュできます。コストベースの最適化方式を使用して結合の処理順序を変更できます。

データ統合サービスでは、マッピングに複数の最適化方式を同時に適用できます。例えば、ノーマル最適化レベルを選択すると、データ統合サービスは初期プロジェクション、述部最適化、初期選択、ブランチ刈り込み、またはプッシュイン最適化方式を適用します。

マッピングで、リレーショナルリソース、論理データオブジェクト、物理データオブジェクト、および仮想テーブルに制約を設定し、不要な行をフィルタリングすることもできます。データ統合サービスは、制約を処理することによってマッピングパフォーマンスを向上させることができます。

最適化方式

各種の最適化方式を適用することによって、マッピングで処理される行数が削減されます。マッピングの最適化レベルを設定して、適用される最適化方式を制限できます。

データ統合サービスでは、以下の最適化方式を適用できます。

- プッシュダウンの最適化
- 初期プロジェクション最適化
- 初期選択最適化
- ブランチ刈り込み最適化
- 最適化にプッシュイン
- 述部最適化
- グローバル述部最適化
- コストベース最適化
- データシップ結合最適化
- 準結合最適化

データ統合サービスでは、マッピングに複数の最適化方式を同時に適用できます。例えば、ノーマル最適化レベルを選択すると、データ統合サービスは、初期プロジェクション最適化、述部最適化、グローバル述部最適化、ブランチ刈り込み最適化、および初期選択最適化またはプッシュイン最適化方式を適用します。

初期プロジェクション最適化方法

データ統合サービスで初期プロジェクション最適化方式を適用すると、未使用のポートが特定され、それらのポート間のリンクが削除されます。

初期プロジェクション最適化方式では、トランスフォーメーション間を移動するデータの量を少なくすることによってパフォーマンスを向上させます。データ統合サービスは、マッピングを処理するときに、マッピング内のすべての接続ポートのデータをトランスフォーメーション間で移動します。大きく複雑なマッピング、またはネストされたマプレットを使用するマッピングには、データをターゲットに提供しないポートがあります。データ統合サービスにより、データをターゲットに提供しないポートが特定されます。未使用のポートが特定されたら、すべての未使用のポート間のリンクがマッピングから削除されます。

すべてのリンクが削除されるわけではありません。例えば、以下のリンクは削除されません。

- 副次効果のあるトランスフォーメーションに接続されているリンク。
- ABORT()関数または ERROR()関数の呼び出し、メールの送信、またはストアードプロシージャの呼び出しを実行するトランスフォーメーションに接続されているリンク。

トランスフォーメーション内のすべてのポートが未使用であると判断されたら、データが最も少ないポートへのリンクを除くすべてのトランスフォーメーションリンクが削除されます。未使用のトランスフォーメーションはマッピングから削除されません。

この最適化方式は、Developer ツールではデフォルトで有効になっています。

初期選択最適化方式

Data Integration Service で初期選択最適化方式を適用すると、マッピングでフィルタトランスフォーメーションの分割、移動、削除が実行されます。フィルタはマッピングのソースに近づくように移動されます。

フィルタ条件が論理積の場合、フィルタトランスフォーメーションは分割されることがあります。例えば、フィルタ条件「A>100 AND B<50」は、2つのより単純な条件「A>100」および「B<50」に分割されることがあります。フィルタを分割する場合、Data Integration Service は、簡略化されたフィルタをマッピングパイプラインの上に移動してソースに近づけます。分割されたフィルタはパイプラインで個別に移動されます。

初期選択最適化方式は、最適化レベルをノーマルまたは完全に設定した場合に Developer tool でデフォルトで有効になります。フィルタトランスフォーメーションより前に出てくるトランスフォーメーションに副次作用がある場合、データ統合サービスは初期選択最適化を無視します。データ統合サービスは、SQL 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、Java トランスフォーメーションに副次作用があるかを判断できません。これらのトランスフォーメーションに副次作用がない場合は、これらのトランスフォーメーションの初期選択最適化を設定できます。

初期選択最適化でパフォーマンスが向上しない場合は無効にしてください。データ統合サービスでは、デフォルトでこの最適化方式が有効化されます。

ブランチ刈り込み最適化方式

データ統合サービスでは、マッピングにおいてターゲットに行をまったく渡さないトランスフォーメーションにブランチ刈り込み最適化方式を適用できます。

データ行に関してフィルタ条件が FALSE に評価されると、データ統合サービスはフィルタトランスフォーメーションを削除することがあります。例えば、マッピングに、2つのリレーショナルソースのデータをフィルタリングする2つのフィルタトランスフォーメーションが存在するとします。一方のフィルタトランスフォーメーションには「Country=US」というフィルタ条件が設定されており、もう一方のフィルタトランスフォーメーションには「Country=Canada」というフィルタ条件が設定されているとします。共有体トランスフォーメーションは、これら2つのリレーショナルソースを結合するもので、「Country=US」というフィルタ条件が設定されているとします。データ統合サービスは、フィルタ条件「Country=Canada」が設定されたフィルタトランスフォーメーションをマッピングから削除することがあります。

ブランチ刈り込み最適化方式は、最適化レベルをノーマルまたは完全に設定した場合に Developer ツールでデフォルトで有効になります。この最適化でパフォーマンスが向上しない場合は、最適化レベルを最小またはなしに設定してブランチ刈り込みを無効にすることができます。

述部最適化方式

データ統合サービスで述部最適化方式を適用すると、マッピングで生成された述部式が調べられ、マッピングのパフォーマンス向上のために式を簡略化するか書き直すことができるかどうか判断されます。

データ統合サービスは、マッピングを実行するときに、マッピングソースに対するクエリを生成し、マッピングロジックおよびマッピング内のトランスフォーメーションに基づいてクエリ結果に対して操作を実行します。クエリおよび操作には、多くの場合、述部式が含まれます。述部式は、データが満たす必要がある条件を表します。述部式の例としては、フィルタトランスフォーメーションとジョイナトランスフォーメーションのフィルタ条件と結合条件があります。

また、述部最適化方式では、マッピングのパフォーマンス向上のために、マッピングで述部式をできる限り早期に適用するように試行されます。

データ統合サービスでは、既存の述部式からリレーションを推測し、新しい述部式を作成します。例えば、マッピングに結合条件「A=B」を持つジョイナトランスフォーメーションおよびフィルタ条件「A>5」を持つフィルタトランスフォーメーションが含まれているとします。データ統合サービスは、「B>5」を結合条件に追加できます。

データ統合サービスでは、述部最適化方式と初期選択最適化方式の両方をマッピングに適用できる場合、それらの方式が適用されます。例えば、データ統合サービスで述部最適化方式を使用して新しいフィルタ条件が作成されるとき、初期選択方式を使用してマッピングのアップストリームへの条件の移動も試行されます。両方の最適化方式を適用すると、一方の方式のみを適用したときと比べてマッピングのパフォーマンスが向上します。

データ統合サービスで述部最適化方式が適用されるのは、それによってパフォーマンスが向上する場合です。適用することでマッピングの結果が変わったりマッピングのパフォーマンスが低下したりする場合は、この方式は適用されません。データ統合サービスでは、デフォルトでこの最適化方式が適用されます。

コストベースの最適化方式

コストベースの最適化では、データ統合サービスによって、マッピングが評価されて意味的に同等のマッピングが生成され、最適なパフォーマンスでマッピングが実行されます。コストベースの最適化では、隣接する内部結合や完全な外部結合の操作を実行するマッピングの実行時間が短縮されます。

意味的に同等のマッピングとは、同じ関数を実行して同じ結果になるマッピングです。意味的に同等のマッピングを生成するために、データ統合サービスでは、元のマッピングがフラグメントに分割されます。次に、最適化できるマッピングのフラグメントが特定されます。

最適化中に、フラグメント内のトランスフォーメーションの追加、削除、または順序変更が行われる場合があります。最適化したフラグメントが元のフラグメントと同じ結果になることが検証され、最適化したフラグメントを使用する代替マッピングが形成されます。

データ統合サービスは、ソートされたマージ結合のパフォーマンスが、ネストされたループ結合のパフォーマンスよりも高いと判断した場合に、ソートされたマージ結合を適用することもできます。ソートされたマージ結合では、結合を実行する前に、ソート順を使用して2つのデータセットを整列します。ネストされたループ結合では、ネストされたループを使用して2つのデータセットを結合します。データをソートするためのコストが、ネストされたループ結合を処理するためのコストより低い場合、データ統合サービスは、ソースのソート情報を使用したり、ソータートランスフォーメーションを作成したりできます。

元のマッピングと意味的に同等のすべてのマッピングまたはほぼすべてのマッピングが生成されます。プロファイリング統計またはデータベース統計を使用して、元のマッピングおよび各代替マッピングのコストを計算します。次に、最も早く実行するマッピングを特定します。最適な代替マッピングに対して検証チェックが実行され、そのマッピングが有効で元のマッピングと同じ結果になることが確認されます。

最適な代替マッピングがメモリにキャッシュされます。マッピングを実行するときに、代替マッピングが取得されて元のマッピングの代わりに実行されます。

この方式は、Developer tool ではデフォルトで無効になっています。

データシップ結合最適化方式

データシップ結合最適化方式では、大きいデータセットに隣接する小さいデータセットを特定し、結合処理の時間を短縮しようとします。データ統合サービスは、2つのテーブルのサイズが大幅に異なる場合、データシップ結合最適化方式を適用しようとします。

例えば、データ統合サービスは、データシップ結合最適化方式を適用して、10,000 行のマスタテーブルと 1,000,000 行の詳細テーブルを結合できます。データ統合サービスは、データシップ結合を実行するために、大きい詳細テーブルを含むデータベースに一時ステージングテーブルを作成します。次に、データ統合サービスは、小さいマスタテーブルを一時テーブルにコピーし、一時テーブルのデータと大きい詳細テーブルのデー

タを結合します。データ統合サービスが結合操作を実行すると、データベースでジョイナトランスフォーメーションロジックが処理されます。

データシップ結合最適化方式を適用する前に、データ統合サービスは分析を実行し、データシップ結合最適化が可能か、また実行する価値があると考えられるかを判断します。分析によって、この方式でパフォーマンスが向上する可能性が高いと判断されたら、この方式がマッピングに適用されます。その後、マッピングが再分析されて、データシップ結合最適化を行う機会が他にもあるかどうか判断されます。必要に応じて、最適化がさらに実行されます。

この方式は、Developer ツールではデフォルトで無効になっています。

データシップ結合でパフォーマンスを向上させるための要件

データシップ結合最適化方式では、パフォーマンスが常に向上するとは限りません。データシップ結合最適化によるマッピングのパフォーマンスに影響する要因を次に示します。

- ジョイナトランスフォーメーションのマスタソースの行が明細ソースよりも大幅に少なくなければならない。
- 明細ソースが最適化の正当性を保証できるほど大幅に大きくなければならない。明細ソースが十分に大きくない場合、データ統合サービスは、データシップ結合最適化方式を適用せずに、マスタソースおよび明細ソースからすべてのデータを読み取る方が速いと判断します。

データシップ結合最適化に関するルールとガイドライン

データ統合サービスでは、次の要件を満たすジョイナトランスフォーメーションにデータシップ結合最適化を適用できます。

- 結合タイプがノーマル、マスタ外部、または明細外部である。
- 明細パイプラインがリレーショナルソースから生成されている。
- ジョイナトランスフォーメーション範囲が [すべての入力] である（マッピングでターゲットベースのコミットが使用される場合）。
- マスタパイプラインと明細パイプラインでトランスフォーメーションが共有されていない。
- 明細ソースとジョイナトランスフォーメーションの間のブランチがマッピングに含まれていない。
- 結合の詳細サイドを含むデータベースが Unicode エンコーディングをサポートしない IBM DB2 データベースの場合、データ統合サービスはデータシップ結合最適化方式の適用に失敗します。

準結合最適化方式

準結合最適化方式では、マッピングで結合操作を変更することで、ソースから抽出されるデータの量の削減が試行されます。

データ統合サービスでは、一方の入力グループに他方よりも多くの行が含まれている場合、結合条件に基づいて、小さい方のグループに一致するものがない行が大きい方のグループに多数含まれているときに、準結合最適化方式がジョイナトランスフォーメーションに適用されます。データ統合サービスは、小さい方のグループから行を読み取り、大きい方のグループで一致する行を見つけて結合操作を実行することで、1つの結合オペランドのデータセットのサイズを小さくしようとします。データセットのサイズを小さくすると、データ統合サービスで大きい方のグループソースから不要な行が読み取られなくなるため、マッピングのパフォーマンスが向上します。データ統合サービスによって、結合条件が大きい方のグループソースに移動され、小さい方のグループと一致する行のみが読み取られます。

準結合最適化方式を適用する前に、データ統合サービスは分析を実行し、準結合最適化が可能か、また実行する価値があると考えられるかを判断します。分析によって、この方式でパフォーマンスが向上する可能性が高いと判断されたら、この方式がマッピングに適用されます。その後、マッピングが再分析されて、準結合最適化を行う機会が他にもあるかどうか判断されます。必要に応じて、最適化がさらに実行されます。

この方式は、Developer ツールではデフォルトで無効になっています。

準結合最適化でパフォーマンスを向上させるための要件

準結合最適化方式では、パフォーマンスが常に向上するとは限りません。準結合最適化によるマッピングのパフォーマンスに影響する要因を次に示します。

- ジョイナトランスフォーメーションのマスタソースの行が明細ソースよりも大幅に少なくなければならない。
- 明細ソースが最適化の正当性を保証できるだけの大きさでなければならない。準結合最適化方式を適用すると、マッピング処理にオーバーヘッド時間が加わります。明細ソースが小さい場合は、準結合最適化の適用にかかる時間が、明細ソース内のすべての行の処理にかかる時間を超える可能性があります。
- Data Integration Service で、通常の結合操作と準結合操作にかかる時間を正確に比較できるように、ジョイナトランスフォーメーションのソース行数の統計を取得できなければならない。

準結合最適化に関するルールとガイドライン

Data Integration Service では、以下の要件を満たすジョイナトランスフォーメーションに準結合最適化を適用できます。

- 結合タイプがノーマル、マスタ外部、または明細外部である。ジョイナトランスフォーメーションでは、完全外部結合は実行できません。
- 明細パイプラインがリレーショナルソースから生成されている。
- 結合条件が有効なソート-マージ-結合条件である。つまり、各句が1つのマスタポートと1つの明細ポートの等式である必要があります。複数の句がある場合は、AND で結合する必要があります。
- ジョイナトランスフォーメーション範囲が [すべての入力] である（マッピングでターゲットベースのコミットが使用されない場合）。
- マスタパイプラインと明細パイプラインでトランスフォーメーションが共有されていない。
- 明細ソースとジョイナトランスフォーメーションの間のブランチがマッピングに含まれていない。

最適化されたマッピングの表示

最適化されたマッピングを表示して、最適化方式がマッピングにどのように影響するかを判断できます。

- ▶ エディタの空の領域を右クリックし、**最適化されたマッピングの表示** をクリックします。
データ統合サービスにより最適化されたマッピングが生成されます。

最適化レベル

設定した最適化レベルに基づいて、データ統合サービスによりマッピングが最適化されます。マッピングでノーマル以外の最適化レベルを使用する場合は、最適化レベルを設定します。デフォルトでは、各マッピングでノーマルの最適化レベルが使用されます。

以下の最適化レベルのいずれかを選択できます。

なし

データ統合サービスは最適化は適用されません。

最小

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。

ノーマル

データ統合サービスは、初期プロジェクション、初期選択、ブランチ刈り込み、プッシュイン、グローバル述部、述部の最適化方式を適用します。ノーマルがデフォルトの最適化レベルです。

完全

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、ブランチ刈り込み、述部、プッシュイン、準結合、データシップ結合の最適化方式を適用します。

マッピングを Developer tool の **【実行】** メニューまたはマッピングエディタから実行した場合、ノーマルの最適化レベルが適用されます。マッピングを **【実行】** メニューから実行した場合は、マッピング設定に従って最適化レベルが適用されます。マッピングをコマンドラインから実行した場合は、アプリケーションで設定したマッピングのデプロイメントのプロパティに従って最適化レベルが適用されます。

注: データ統合サービスはプッシュダウンの最適化方式に最適化レベルを適用しません。マッピングランタイムプロパティでマッピングのプッシュダウンの最適化を設定できます。

関連項目：

- [「プッシュダウンの最適化の概要」 \(ページ 225\)](#)

Developer ツールのマッピングに対する最適化レベルの設定

マッピングを **【実行】** メニューまたはマッピングエディタから実行するとき、マッピングは Developer ツールによってノーマル最適化レベルで実行されます。マッピングを異なる最適化レベルで実行するには、マッピングを **【実行設定】** ダイアログボックスから実行します。

1. マッピングを開きます。
2. **【実行】** > **【実行ダイアログを開く】** を選択します。
【実行設定】 ダイアログボックスが表示されます。
3. 適用する最適化レベルが含まれているマッピング設定を選択するか、マッピング設定を作成します。
4. **【詳細】** タブをクリックします。
5. オプティマイザレベルを変更します。
6. **【適用】** をクリックします。
7. **【実行】** をクリックしてマッピングを実行します。

選択したマッピング設定内の最適化レベルでマッピングが実行されます。

デプロイ済みのマッピングに対する最適化レベルの設定

アプリケーションのマッピングのデプロイメントのプロパティを変更して、コマンドラインから実行するマッピングの最適化レベルを設定します。

マッピングがアプリケーションに含まれている必要があります。

1. マッピングを含むアプリケーションを開きます。
2. **【詳細】** タブをクリックします。
3. 最適化レベルを選択します。
4. アプリケーションを保存します。

最適化レベルを変更した後、アプリケーションを再デプロイする必要があります。

第 12 章

プッシュダウンの最適化

この章では、以下の項目について説明します。

- [プッシュダウンの最適化の概要, 225 ページ](#)
- [プッシュダウンタイプ, 226 ページ](#)
- [トランスフォーメーションロジックのプッシュダウン, 228 ページ](#)
- [ソースへのプッシュダウンの最適化, 229 ページ](#)
- [プッシュダウンの最適化の式, 233 ページ](#)
- [データ統合サービスとソースの出力の比較, 248 ページ](#)

プッシュダウンの最適化の概要

データ統合サービスでプッシュダウンの最適化を適用すると、トランスフォーメーションロジックがソースにプッシュされます。データ統合サービスはトランスフォーメーションロジックを SQL クエリに変換し、その SQL クエリをデータベースに送信します。トランスフォーメーションを処理する SQL クエリはソースデータベースで実行されます。

ソースデータベースがデータ統合サービスよりも高速にトランスフォーメーションロジックを処理できる場合、プッシュダウンの最適化により、マッピングのパフォーマンスが向上します。データ統合サービスがソースから読み取るデータも少なくなります。

データ統合サービスがソースデータベースにプッシュするトランスフォーメーションロジックの量は、データベース、トランスフォーメーションロジック、およびマッピング設定によって決まります。データ統合サービスは、データベースにプッシュできないすべてのトランスフォーメーションロジックを処理します。

プッシュダウンの最適化をマッピングに設定すると、ソースからターゲットまで、またはソースデータベースにプッシュできないダウンストリームトランスフォーメーションに達するまで、最適化されたマッピングが分析されます。データ統合サービスでは、トランスフォーメーションロジックがプッシュダウンされたソースごとに SELECT 文を生成して実行します。次に、Data Integration Service はこの SQL クエリの結果が読み込み、マッピングの残りのトランスフォーメーションを処理します。

関連項目：

- [「最適化レベル」 \(ページ 222\)](#)

プッシュダウンタイプ

マッピングのランタイムプロパティでプッシュダウンタイプを選択した場合、データ統合サービスはプッシュダウンの最適化をマッピングに適用します。

以下のプッシュダウンタイプを選択することができます。

- なし。マッピングのプッシュダウンタイプを選択しません。
- ソース。データ統合サービスは、ソースデータベースに、できるだけ多くのトランスフォーメーションロジックのプッシュダウンを試みます。
- 全体。データ統合サービスは、トランスフォーメーションロジック全体をソースデータベースにプッシュします。

プッシュダウンタイプの文字列パラメータを作成し、以下のパラメータ値を使用することもできます。

- フル
- ソース
- なし

完全なプッシュダウンの最適化

データ統合サービスは、完全なプッシュダウンの最適化を適用するときに、マッピングのすべてのトランスフォーメーションロジックをソースデータベースにプッシュします。マッピングランタイムプロパティで完全なプッシュダウンを設定できます。

完全なプッシュダウンの最適化は、ソースとターゲットが同じデータベースにある場合や、アグリゲータトランスフォーメーションやフィルタトランスフォーメーションなどのトランスフォーメーションをソースデータベースで処理してデータの移動量を減らす場合に最適です。例えば、マッピングに Teradata ソースと Teradata ターゲットが含まれている場合、完全なプッシュダウンの最適化を設定して、処理するすべてのトランスフォーメーションロジックを Teradata ソースデータベースから Teradata ターゲットデータベースにプッシュします。

完全なプッシュダウン用にアップデートストラテジトランスフォーメーションを含むマッピングを設定する場合、マッピングのプッシュダウン互換性を判断する必要があります。

次の場合、データ統合サービスはアップデートストラテジトランスフォーメーションを含むマッピングをプッシュダウンできます。

- アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションが、異なるキーを持つ複数の行を受信する場合。
- アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションが、並べ替えできる同じキーを持つ複数の行を受信する場合。

次の場合、データ統合サービスはアップデートストラテジトランスフォーメーションを含むマッピングをプッシュダウンできません。

- アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションが、並べ替えできない同じキーを持つ複数の行を受信する場合。

マッピングでプッシュダウン互換性パラメータを使用することもできます。次のパラメータ値を使用できます。

- noMultipleRowsWithSameKeyOnTarget
- reorderAllowedForMultipleRowsWithSameKey
- reorderNotAllowedForRowsWithSameKey

データ統合サービスで完全なプッシュダウンの最適化を使用できるソースは次のとおりです。

- Oracle
- IBM DB2
- Microsoft SQL Server
- Teradata
- Netezza
- Greenplum
- SAP HANA

完全なプッシュダウンの最適化に関するルールおよびガイドライン

完全なプッシュダウンの最適化を設定する場合は、以下のルールとガイドラインを考慮してください。

- データ統合サービスは、マッピング内のすべてのトランスフォーメーションロジックを IBM DB2、Oracle、Microsoft SQL Server、および ODBC ソース（Teradata、Greenplum、Netezza、SAP HANA など）にプッシュできます。
- アップデートストラテジトランスフォーメーションを含むマッピングに対して完全なプッシュダウンの最適化を設定する場合、「更新または挿入」ストラテジは Oracle と Teradata でのみ使用できます。

ソースプッシュダウン

データ統合サービスは、ソースプッシュダウンを適用するときにソースからターゲットへのマッピングを分析します。これを行わないと、ダウンストリームトランスフォーメーションに達するまで、ソースデータベースにプッシュできません。

データ統合サービスは、データベースにプッシュできる各トランスフォーメーションのトランスフォーメーションロジックに基づいて、SELECT 文を生成、実行します。次に、統合サービスではこの SQL クエリの結果が読み込まれ、残りのトランスフォーメーションが処理されます。

ソースとターゲットが異なるデータベースに存在している場合、ソースプッシュダウンを使用するようにマッピングを設定できます。例えば、マッピングに Teradata ソースと Oracle ターゲットが含まれている場合、ソースプッシュダウンを設定して、処理する一部のトランスフォーメーションロジックを Teradata ソースにプッシュできます。

プッシュダウンの設定

マッピングランタイムプロパティで、プッシュダウンの最適化のためのマッピングを設定できます。

1. マッピングを開きます。
2. **【プロパティ】** タブで、**【ランタイム】** を選択します。
3. プッシュダウンタイプを選択するか、プッシュダウンパラメータを割り当てます。
 - **なし**。データ統合サービスはマッピングロジックをソースデータベースにプッシュダウンしません。
 - **完全**。データ統合サービスはマッピングロジック全体をソースデータベースにプッシュダウンします。

- **ソース。**データ統合サービスは、ターゲットを除くすべてのマッピングロジックをソースデータベースにプッシュダウンします。
 - **パラメータの割り当て。**プッシュダウンタイプに設定したパラメータを選択するか、新しいパラメータを作成して、[OK] をクリックします。
4. 完全なプッシュダウンの最適化を選択していて、マッピングにアップデートストラテジトランスフォーメーションが含まれている場合は、必要に応じて、プッシュダウン互換性オプションを選択するか、プッシュダウン互換性パラメータを割り当てることができます。
- **複数の行に同じキーがない。**アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションは、同じキーを持つ複数の行を受け取ります。データ統合サービスは、アップデートストラテジトランスフォーメーションをプッシュダウンできます。
 - **同じキーの複数の行を並べ替えることができる。**アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションは、並べ替えが可能な同じキーの複数の行を受け取ります。データ統合サービスは、アップデートストラテジトランスフォーメーションをプッシュダウンできます。
 - **同じキーの複数の行を並べ替えることができない。**アップデートストラテジトランスフォーメーションに接続されているターゲットトランスフォーメーションは、並べ替えが不可能な同じキーの複数の行を受け取ります。データ統合サービスは、アップデートストラテジトランスフォーメーションをプッシュダウンできません。
 - **パラメータの割り当て。**プッシュダウン互換性に設定したパラメータを選択するか、パラメータを作成して、[OK] をクリックします。

トランスフォーメーションロジックのプッシュダウン

データ統合サービスは、プッシュダウンの最適化を使用してトランスフォーメーションロジックをソースデータベースにプッシュします。データ統合サービスがソースデータベースにプッシュするトランスフォーメーションロジックの量は、データベース、トランスフォーメーションロジック、およびマッピング設定によって決まります。データ統合サービスは、データベースにプッシュできないすべてのトランスフォーメーションロジックを処理します。

データ統合サービスは、以下のトランスフォーメーションロジックをソースデータベースにプッシュできます。

- アグリゲータ
- 式
- フィルタ
- ジョイナ
- ルックアップ
- ソータ
- 共有体

データ統合サービスは、以下の場合にソースにトランスフォーメーションロジックをプッシュできません。

- ソースがバイナリデータ型のカラムを含む場合。
- ソースは、フィルタ条件、または式またはジョイナトランスフォーメーションロジックのユーザー定義ジョインを含む、カスタマイズされたデータオブジェクトです。

- ソースは、別のデータベース管理システムにあるか、ジョイナまたは共有体トランスフォーメーションロジックに別の接続を使用しています。

データ統合サービスは、ソースにプッシュできないマッピングロジックを処理します。

ソースへのプッシュダウンの最適化

データ統合サービスはトランスフォーメーションロジックを別のソース、例えばリレーショナルソースやデータベース固有の ODBC ドライバを使用するソースにプッシュできます。Data Integration Service がプッシュするトランスフォーメーションロジックのタイプは、ソースタイプによって異なります。

Data Integration Service は、トランスフォーメーションロジックを次のタイプのソースにプッシュできます。

- リレーショナルソース
- ネイティブのデータベースドライバを使用するソース
- PowerExchange の非リレーショナルソース
- データベース固有の ODBC ドライバを使用するソース
- SAP ソース

リレーショナルソースへのプッシュダウンの最適化

データ統合サービスは、ネイティブドライバまたはデータベース固有の ODBC ドライバを使用して、トランスフォーメーションロジックをリレーショナルソースにプッシュできます。

データ統合サービスは、アグリゲータ、式、フィルタ、ジョイナ、ソータ、および共有体トランスフォーメーションロジックを以下のリレーショナルソースにプッシュできます。

- Greenplum
- Hive
- IBM DB2
- Microsoft SQL Server
- Oracle
- SAP HANA
- Sybase
- Teradata

リレーショナルソースにアグリゲータトランスフォーメーションロジックをプッシュすると、パススルーポートがグループ別ポートの場合はパススルーポートが有効になります。トランスフォーメーション言語には、アグリゲータトランスフォーメーションで使用できる集計関数が含まれています。

以下の表では、IBM DB2 リレーショナルソースで有効な集計関数を示しています。

集計関数	DB2-LUW	DB2i	DB2z/os
AVG	○	○	○
COUNT	○	○	○

集計関数	DB2-LUW	DB2i	DB2z/os
FIRST	×	×	×
LAST	×	×	×
MAX	○	○	○
MEDIAN	×	×	×
MIN	○	○	○
PERCENTILE	×	×	×
STDDEV	○	○	○
SUM	○	○	○
VARIANCE	○	○	○

以下の表では、Greenplum、Hive、MSSQL、Oracle、Sybase、および Teradata リレーショナルソースで有効な集計関数を示しています。

集計関数	Greenplum	Hive	MSSQL	Oracle	Sybase	Teradata
AVG	○	○	○	○	○	○
COUNT	○	○	○	○	○	○
FIRST	×	×	×	×	×	×
LAST	×	×	×	×	×	×
MAX	○	○	○	○	○	○
MEDIAN	×	×	×	○	×	×
MIN	○	○	○	○	○	○
PERCENTILE	×	×	×	×	×	×
STDDEV	○	○	○	○	×	○
SUM	○	○	○	○	○	○
VARIANCE	○	○	○	○	×	○

以下の表では、MSSQL、Oracle、および Sybase リレーショナルソースで有効な集計関数を示しています。

集計関数	MSSQL	Oracle	Sybase
AVG	○	○	○
COUNT	○	○	○
FIRST	×	×	×
LAST	×	×	×
MAX	○	○	○
MEDIAN	×	○	×
MIN	○	○	○
PERCENTILE	×	×	×
STDDEV	○	○	×
SUM	○	○	○
VARIANCE	○	○	×

リレーショナルソースには NULL 値を扱うためのデフォルト設定があります。デフォルトでは、一部のデータベースは NULL 値を他の値よりも低く扱い、一部のデータベースは NULL 値を他の値よりも高く扱います。ソースにデフォルトの NULL の順序付けがあるときは、ソータートランスフォーメーションロジックをリレーショナルソースにプッシュして正確な結果を得ることができます。

重複しない出力行にソータートランスフォーメーションを設定すると、大文字と小文字を区別するソートを有効にしてトランスフォーメーションロジックを DB2、Sybase、および Oracle のソースにプッシュする必要があります。

データ統合サービスは、Decimal データ型を含む関数を Hive ソースにプッシュすることができません。

ネイティブソースへのプッシュダウンの最適化

ネイティブドライバを使用してトランスフォーメーションロジックをリレーショナルソースにプッシュするときに、データ統合サービスはネイティブデータベース SQL を使用する SQL 文を生成します。

データ統合サービスは、アグリゲータ、式、フィルタ、ジョイナ、ソータ、および共有体の各トランスフォーメーションロジックを、次のネイティブソースにプッシュできます。

- IBM DB2 for Linux、UNIX、および Windows (「DB2 for LUW」)
- Microsoft SQL Server。データ統合サービスは、Windows 上で実行される場合、Microsoft SQL Server へのネイティブ接続を使用できます。
- Oracle

データ統合サービスは、フィルタトランスフォーメーションロジックを次のネイティブソースにプッシュできます。

- IBM DB2 for i5/OS
- IBM DB2 for z/OS

PowerExchange 非リレーショナルソースへのプッシュダウンの最適化

z/OS システム上の PowerExchange 非リレーショナルデータソースの場合、Data Integration Service はフィルタトランスフォーメーションロジックを PowerExchange にプッシュします。PowerExchange は、ソースが処理できるクエリにロジックを変換します。

Data Integration Service は、フィルタトランスフォーメーションロジックを次のタイプの非リレーショナルソースにプッシュできます。

- IBM IMS
- シーケンシャルデータセット
- VSAM

ODBC ソースへのプッシュダウンの最適化

データ統合サービスは、トランスフォーメーションロジックを、データベース固有の ODBC ドライバを使用するデータベースにプッシュできます。[その他] として ODBC プロバイダを選択した場合、データ統合サービスは、トランスフォーメーションロジックをソースにプッシュできません。

データベース固有の ODBC ドライバを使用してソースに接続する場合、データ統合サービスはネイティブデータベース SQL を使用して SQL 文を生成します。

ODBC プロバイダは、ODBC 接続オブジェクトで指定できます。

以下の ODBC 接続タイプオブジェクトに対して、特定の ODBC プロバイダを設定できます。

- Greenplum
- Microsoft SQL Server
- Netezza
- SAP HANA
- Sybase ASE
- Teradata

SAP ソースへのプッシュダウンの最適化

Data Integration Service は、カラム名、演算子、リテラル文字列を含む式について、フィルタトランスフォーメーションロジックを SAP ソースにプッシュできます。Data Integration Service がトランスフォーメーションロジックを SAP にプッシュする際、式のリテラル文字列は SAP のデータ型に変換されます。

Data Integration Service は、TO_DATE が DATS、TIMS、または ACCP データ型の文字列を以下のいずれかの日付フォーマットに変換する場合、TO_DATE 関数を含むフィルタトランスフォーメーションロジックをプッシュできます。

- 'MM/DD/YYYY'
- 'YYYY/MM/DD'
- 'YYYY-MM-DD HH24:MI:SS'
- 'YYYY/MM/DD HH24:MI:SS'
- 'MM/DD/YYYY HH24:MI:SS'

Data Integration Service は、TO_DATE 関数が DATS、TIMS、または ACCP 以外のデータ型に適用される場合、または、Data Integration Services が SAP にプッシュできないフォーマットに TO_DATE が文字列が変換

される場合、そのトランスフォーメーションロジックを処理します。Data Integration Service は、TO_DATE 以外の Informatica 関数を含むトランスフォーメーションロジックを処理します。Data Integration Service は、TO_DATE 以外の Informatica 関数を含むトランスフォーメーションロジックを処理します。

フィルタトランスフォーメーション式には、AND または OR で区切って複数の条件を指定できます。条件が複数の SAP テーブルに適用される場合、Data Integration Service は、SAP データオブジェクトで Open SQL ABAP ジョイン構文が使用されているときに、トランスフォーメーションロジックを SAP にプッシュできます。SAP データオブジェクトの読み取り操作で Select 構文モードを設定します。

SAP データ型の例外

ソースがフィルタトランスフォーメーションロジックを処理できず、トランスフォーメーション式に以下のデータ型が含まれる場合、データ統合サービスがフィルタトランスフォーメーションロジックを処理します。

- RAW
- LRAW
- LCHR

プッシュダウンの最適化の式

トランスフォーメーションにソースがサポートする演算子と関数が含まれる場合、Data Integration Service はトランスフォーメーションロジックをソースデータベースにプッシュできます。Data Integration Service は、データベース内の対応する演算子および関数を判断することによって、トランスフォーメーション式をクエリに変換します。対応する演算子または関数が存在しない場合、Data Integration Service がトランスフォーメーションロジックを処理します。

ソースで ODBC 接続が使用されていて、ODBC 接続オブジェクト内にデータベース固有の ODBC プロバイダを設定した場合、Data Integration Service はそのソースをネイティブソースタイプと見なします。

関数

Informatica 関数は、z/OS の非リレーショナルソースには使用できません。以下の表では、IBM DB2 ソースのプッシュダウンの最適化に使用できる Informatica 関数を示しています。

関数	DB2 for i5/OS ¹	DB2 for LUW	DB2 for z/OS ¹
ABORT ()	×	×	×
ABS()	×	○	×
ADD_TO_DATE()	○	○	○
AES_DECRYPT()	×	×	×
AES_ENCRYPT()	×	×	×
ASCII()	○	○	○
AVG()	○	○	○
CEIL()	○	○	○

関数	DB2 for i5/OS ¹	DB2 for LUW	DB2 for z/OS ¹
CHOOSE()	×	×	×
CHR()	×	○	×
CHRCODE()	×	○	○
COMPRESS()	×	×	×
CONCAT()	○	○	○
COS()	○	○	○
COSH()	○	○	○
COUNT()	○	○	○
CRC32()	×	×	×
CREATE_TIMESTAMP_TZ()	×	×	×
CUME()	×	×	×
DATE_COMPARE()	○	○	○
DATE_DIFF()	×	×	×
DECODE()	×	○	×
DECODE_BASE64()	×	×	×
DECOMPRESS()	×	×	×
ENCODE_BASE64()	×	×	×
ERROR()	×	×	×
EXP()	×	○	×
FIRST()	×	×	×
FLOOR()	×	○	×
FV()	×	×	×
GET_DATE_PART()	○	○	○
GET_TIMESTAMP()	×	×	×
GET_TIMEZONE()	×	×	×
GREATEST()	×	×	×
IIF()	×	○	×

関数	DB2 for i5/OS ¹	DB2 for LUW	DB2 for z/OS ¹
IN()	×	○	×
INDEXOF()	×	×	×
INITCAP()	×	×	×
INSTR()	○	○	○
IS_DATE()	×	×	×
IS_NUMBER()	×	×	×
IS_SPACES()	×	×	×
ISNULL()	○	○	○
LAST()	×	×	×
LAST_DAY()	×	×	×
LEAST()	×	×	×
LENGTH()	○	○	○
LN()	○	○	○
LOG()	○	○	○
LOWER()	○	○	○
LPAD()	×	×	×
LTRIM()	○	○	○
MAKE_DATE_TIME()	×	×	×
MAX()	○	○	○
MD5()	×	×	×
MEDIAN()	×	×	×
METAPHONE()	×	×	×
MIN()	○	○	○
MOD()	○	○	○
MOVINGAVG()	×	×	×
MOVINGSUM()	×	×	×
NPER()	×	×	×

関数	DB2 for i5/OS ¹	DB2 for LUW	DB2 for z/OS ¹
PERCENTILE()	×	×	×
PMT()	×	×	×
POWER()	○	○	○
PV()	×	×	×
RAND()	×	×	×
RATE()	×	×	×
REG_EXTRACT()	×	×	×
REG_MATCH()	×	×	×
REG_REPLACE	×	×	×
REPLACECHR()	×	×	×
REPLACESTR()	×	×	×
REVERSE()	×	×	×
ROUND(DATE)	×	×	○
ROUND(NUMBER)	○	○	○
RPAD()	×	×	×
RTRIM()	○	○	○
SET_DATE_PART()	×	×	×
SIGN()	○	○	○
SIN()	○	○	○
SINH()	○	○	○
SOUNDEX()	×	○ ¹	×
SQRT()	×	○	×
STDDEV()	○	○	○
SUBSTR()	○	○	○
SUM()	○	○	○
SYSTIMESTAMP()	○	○	○
TAN()	○	○	○

関数	DB2 for i5/OS ¹	DB2 for LUW	DB2 for z/OS ¹
TANH()	○	○	○
TO_BIGINT	○	○	○
TO_CHAR(DATE)	○	○	○
TO_CHAR(NUMBER)	○	○ ²	○
TO_DATE()	○	○	○
TO_DECIMAL()	○	○ ³	○
TO_DECIMAL38()	×	×	×
TO_FLOAT()	○	○	○
TO_INTEGER()	○	○	○
TO_TIMESTAMP_TZ()	×	×	×
TRUNC(DATE)	×	×	×
TRUNC(NUMBER)	○	○	○
UPPER()	○	○	○
VARIANCE()	○	○	○

¹ データ統合サービスは、フィルタトランスフォーメーションロジックに含まれる場合にのみこれらの関数をプッシュできます。

² この関数が decimal または float の引数を取り、フィルタトランスフォーメーションロジックに含まれる場合にのみ、データ統合サービスはこの関数をプッシュできます。

³ この関数が string の引数を取り、フィルタトランスフォーメーションロジックに含まれる場合にのみ、データ統合サービスはこの関数をプッシュできます。

次の表では、Greenplum、Hive、Microsoft SQL Server、Netezza、Oracle、SAP、SAP HANA、Sybase ASE、および Teradata ソースのプッシュダウンの最適化に使用できる Informatica 関数を示しています。

関数	Greenplum	Hive	Microsoft SQL Server	Netezza	Oracle	SAP ¹	SAP HANA	Sybase ASE	Teradata
ABORT ()	×	×	×	×	×	×	×	×	×
ABS()	○	○	○	○	○	×	○	○	○
ADD_TO_DATE()	○	×	○	○	○	×	×	○	○
AES_DECRYPT()	×	×	×	×	×	×	×	×	×
AES_ENCRYPT()	×	×	×	×	×	×	×	×	×
ASCII()	○	×	○	○	○	×	×	○	×

関数	Greenplum	Hive	Microsoft SQL Server	Netezza	Oracle	SAP ↓	SAP HANA	Sybase ASE	Teradata
AVG()	○	○	○	○	○	×	○	○	○
CEIL()	○	○	○	○	○	×	○	○	○
CHOOSE()	×	×	×	×	×	×	×	×	×
CHR()	○	×	○	○	○	×	○	○	×
CHRCODE()	○	×	○	○	○	×	○	○	×
COMPRESS()	×	×	×	×	×	×	×	×	×
CONCAT()	○	○	○	○	○	×	○	○	○
COS()	○	○	○	○	○	×	○	○	○
COSH()	○	×	○	○	○	×	○	○	○
COUNT()	○	○	○	○	○	×	○	○	○
CRC32()	×	×	×	×	×	×	×	×	×
CREATE_TIMESTAMP_TZ()	×	×	×	×	○	×	×	×	×
CUME()	×	×	○	×	×	×	×	×	×
DATE_COMPARE()	○	×	○	○	○	×	○	○	○
DATE_DIFF()	×	×	×	×	×	×	○	×	×
DECODE()	○	○	○	○	○	×	○	○	○
DECODE_BASE64()	×	×	×	×	×	×	×	×	×
DECOMPRESS()	×	×	×	×	×	×	×	×	×
ENCODE_BASE64()	×	×	×	×	×	×	×	×	×
ERROR()	×	×	×	×	×	×	×	×	×
EXP()	○	○	○	○	○	×	×	○	○
FIRST()	×	×	×	×	×	×	×	×	×
FLOOR()	○	○	○	○	○	×	○	○	○
FV()	×	×	×	×	×	×	×	×	×
GET_DATE_PART()	○	×	○	○	○	×	○	○	○
GET_TIMESTAMP()	×	×	×	×	○	×	×	×	×

関数	Greenplum	Hive	Microsoft SQL Server	Netezza	Oracle	SAP 1	SAP HANA	Sybase ASE	Teradata
GET_TIMEZONE()	×	×	×	×	×	×	×	×	×
GREATEST()	×	×	×	×	○	×	×	×	×
IIF()	○	○	○	○	○	×	○	○	○
IN()	×	×	○	×	○	×	×	○	○
INDEXOF()	×	×	×	×	×	×	×	×	×
INITCAP()	○	×	×	○	○	×	×	×	×
INSTR()	×	×	○	○	○	×	×	○	○
IS_DATE()	×	×	×	×	×	×	×	×	×
IS_NUMBER()	×	×	×	×	×	×	×	×	×
IS_SPACES()	×	×	×	×	×	×	×	×	×
ISNULL()	×	○	○	×	○	×	○	○	○
LAST()	×	×	×	×	×	×	×	×	×
LAST_DAY()	×	×	×	○	○	×	○	×	×
LEAST()	×	×	×	×	○	×	×	×	×
LENGTH()	○	○	○	○	○	×	○	○	○
LN()	○	○	○	×	○	×	○	○	○
LOG()	○	○	○	○	○	×	○	○	○
LOWER()	○	○	○	○	○	×	○	○	○
LPAD()	○	○	×	○	○	×	○	×	×
LTRIM()	○	○	○	○	○	×	○	×	○
MAKE_DATE_TIME()	×	×	×	×	×	×	×	×	×
MAX()	○	○	○	○	○	×	○	○	○
MD5()	×	×	×	×	×	×	×	×	×
MEDIAN()	×	×	×	×	○	×	×	×	×
METAPHONE()	×	×	×	×	×	×	×	×	×
MIN()	○	○	○	○	○	×	○	○	○

関数	Greenplum	Hive	Microsoft SQL Server	Netezza	Oracle	SAP 1	SAP HANA	Sybase ASE	Teradata
MOD()	○	○	○	○	○	×	○	○	○
MOVINGAVG()	×	×	×	×	×	×	×	×	×
MOVINGSUM()	×	×	×	×	×	×	×	×	×
NPER()	×	×	×	×	×	×	×	×	×
PERCENTILE()	×	×	×	×	×	×	×	×	×
PMT()	×	×	×	×	×	×	×	×	×
POWER()	○	○	○	○	○	×	○	○	○
PV()	×	×	×	×	×	×	×	×	×
RAND()	×	×	×	×	×	×	×	×	×
RATE()	×	×	×	×	×	×	×	×	×
REG_EXTRACT()	×	×	×	×	×	×	×	×	×
REG_MATCH()	×	×	×	×	×	×	×	×	×
REG_REPLACE	×	×	×	×	×	×	×	×	×
REPLACECHR()	×	×	×	×	×	×	×	×	×
REPLACESTR()	×	×	×	×	×	×	×	×	×
REVERSE()	×	×	×	×	×	×	×	×	×
ROUND(DATE)	×	×	×	×	○	×	×	×	×
ROUND(NUMBER)	○	○	○	○	○	×	○	○	○
RPAD()	○	○	×	○	○	×	○	×	×
RTRIM()	○	○	○	○	○	×	○	○	○
SET_DATE_PART()	×	×	×	×	×	×	×	×	×
SIGN()	○	×	○	○	○	×	○	○	○
SIN()	○	○	○	○	○	×	○	○	○
SINH()	○	×	○	○	○	×	○	○	○
SOUNDEX()	×	×	○	×	○	×	×	○	×
SQRT()	○	○	○	○	○	×	○	○	○

関数	Greenplum	Hive	Microsoft SQL Server	Netezza	Oracle	SAP ¹	SAP HANA	Sybase ASE	Teradata
STDDEV()	○	×	○	○	○	×	○	○	○
SUBSTR()	○	○	○	○	○	×	○	○	○
SUM()	○	○	○	○	○	×	○	○	○
SYSTIMESTAMP()	○	×	○	○	○	×	○ ²	○	×
TAN()	○	×	○	○	○	×	○	○	○
TANH()	○	×	○	○	○	×	×	○	○
TO_BIGINT	○	○	○	○	○	×	○	○	○
TO_CHAR(DATE)	○	×	○	○	○	×	○	○	○
TO_CHAR(NUMBER)	○	×	○	○	○	×	○	○	○
TO_DATE()	○	×	○	○	○	○	○	○	○
TO_DECIMAL()	○	×	○	○	○	×	○	○	○
TO_DECIMAL38()	○	○	○	○	○	○	○	○	○
TO_FLOAT()	○	×	○	○	○	×	○	○	○
TO_INTEGER()	○	○	○	○	○	×	○	○	○
TO_TIMESTAMP_TZ()	×	×	×	×	○	×	×	×	×
TRUNC(DATE)	○	×	×	○	○	×	○	×	×
TRUNC(NUMBER)	○	×	○	○	○	×	○	○	○
UPPER()	○	○	○	○	○	×	○	○	○
VARIANCE()	○	×	○	○	○	×	×	○	○

¹. データ統合サービスは、これらの関数がフィルタトランスフォーメーションロジックに含まれる場合にのみ、これらをソースにプッシュできます。

². SYSTIMESTAMP()は SS 引数のみをサポートしています。

以下の表に、各 Informatica 関数をプッシュダウンの最適化に使用できるかどうかを示します。

関数	DB2 for LUW	Microsoft SQL Server	ODBC	Oracle	Sybase ASE
ABS()	○	○	○	○	○
ADD_TO_DATE()	○	○	×	○	○

関数	DB2 for LUW	Microsoft SQL Server	ODBC	Oracle	Sybase ASE
ASCII()	○	○	×	○	○
CEIL()	○	○	×	○	○
CHR()	○	○	×	○	○
CONCAT()	○	○	×	○	○
COS()	○	○	○	○	○
COSH()	○	○	×	○	○
DATE_COMPARE()	○	○	○	○	○
DECODE()	○	○	○	○	○
EXP()	○	○	○	なし	○
FLOOR()	×	○	×	○	○
GET_DATE_PART()	○	○	×	○	○
IIF()	○	○	○	×	○
IN()	×	○	○	×	○
INITCAP()	×	×	×	○	×
INSTR()	○	○	×	○	○
ISNULL()	○	○	○	○	○
LAST_DAY()	×	×	×	○	なし
LENGTH()	○	○	×	○	○
LN()	○	×	×	○	○
LOG()	○	○	×	○	○
LOOKUP()	×	×	○	×	×
LOWER()	○	○	○	○	○
LPAD()	×	×	×	○	なし
LTRIM()	×	○	×	○	○
MOD()	×	○	×	○	○
POWER()	×	○	×	○	○
ROUND(DATE)	×	×	×	○	なし

関数	DB2 for LUW	Microsoft SQL Server	ODBC	Oracle	Sybase ASE
ROUND(NUMBER)	○	○	×	○	○
RPAD()	×	×	×	○	×
RTRIM()	○	○	×	○	○
SIGN()	○	○	×	○	○
SIN()	○	○	○	○	○
SINH()	○	○	×	○	○
SOUNDEX()	○ ¹	○	×	○	○
SQRT()	○	○	○	○	○
SUBSTR()	○	○	×	○	○
SYSDATE()	○	○	×	○	○
SYSTIMESTAMP()	○	○	×	○	○
TAN()	○	○	○	○	○
TANH()	○	○	×	○	○
TO_BIGINT	○	○	×	○	○
TO_CHAR(DATE)	○	○	×	○	○
TO_CHAR(NUMBER)	○ ²	○	×	○	○
TO_DATE()	○	○	×	○	○
TO_DECIMAL()	○ ³	○	×	○	○
TO_FLOAT()	○	○	×	○	○
TO_INTEGER()	○	○	×	○	○
TRUNC(DATE)	×	×	×	○	×
TRUNC(NUMBER)	○	○	×	○	○
UPPER()	○	○	○	○	○

¹ データ統合サービスは、フィルタトランスフォーメーションロジックに含まれる場合にのみこれらの関数をプッシュできます。

² この関数が decimal または float の引数を取り、フィルタトランスフォーメーションロジックに含まれる場合にのみ、データ統合サービスはこの関数をプッシュできます。

³ この関数が string の引数を取り、フィルタトランスフォーメーションロジックに含まれる場合にのみ、データ統合サービスはこの関数をプッシュできます。

Hive 関数の例外

特定の条件下において、データ統合サービスは、サポートされている関数を Hive ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、データ統合サービスは、Hive ソースのトランスフォーメーションロジックを処理します。

- 2 番目の引数として、LTRIM にスペースが含まれる。
- 2 番目の引数として、RTRIM にスペースが含まれる。

以下の関数を date データ型とともに使用する場合、データ統合サービスでは Hive ソースのトランスフォーメーションロジックを処理できません。

- CONCAT
- MAX
- MIN
- ROUND
- TO_BIGINIT
- TO_INTEGER

IBM DB2 関数の例外

特定の条件下において、Data Integration Service は、サポートされている関数を IBM DB2 for i5/OS、DB2 for LUW、および DB2 for z/OS ソースにプッシュできません。特定の条件下において、Data Integration Service は、サポートされている関数を IBM DB2 for LUW ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、Data Integration Service は、IBM DB2 ソースのトランスフォーメーションロジックを処理します。

- ADD_TO_DATE または GET_DATE_PART が、ミリ秒またはナノ秒の精度で結果を返す場合。
- LTRIM が 2 つ以上の引数を含む場合。
- RTRIM が 2 つ以上の引数を含む場合。
- TO_BIGINT が、文字列を DB2 for LUW ソースの Bigint 値に変換される場合。
- TO_CHAR が日付を文字列値に変換し、DB2 でサポートされていない形式を指定する場合。
- TO_DATE が文字列値を日付に変換し、DB2 でサポートされていない形式を指定する場合。
- TO_DECIMAL が scale 引数なしで文字列を decimal 値に変換する場合。
- TO_FLOAT が文字列を倍精度浮動小数点数に変換する場合。
- TO_INTEGER が文字列を DB2 for LUW ソースの整数値に変換する場合。

Microsoft SQL Server 関数の例外

特定の条件下において、Data Integration Service はサポートされている関数を Microsoft SQL Server ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、Data Integration Service は、Microsoft SQL Server ソースのトランスフォーメーションロジックを処理します。

- IN が CaseFlag 引数を含む場合。
- INSTR が 4 つ以上の引数を含む場合。
- LTRIM が 2 つ以上の引数を含む場合。

- RTRIM が 2 つ以上の引数を含む場合。
- TO_BIGINT が 2 つ以上の引数を含む場合。
- TO_INTEGER が 2 つ以上の引数を含む場合。

Netezza 関数の例外

特定の条件下において、データ統合サービスは、サポートされている関数を Netezza ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、データ統合サービスは、Netezza ソースのトランスフォーメーションロジックを処理します。

- SYSTIMESTAMP は、YYYY-MM-DD HH24:MI:SS.US 形式の日付を含みます。
- TO_CHAR(DATE)と TO_DATE()は、サブ秒精度の YYYY-MM-DD HH24:MI:SS.US 形式の日付を含みます。

Oracle 関数の例外

特定の条件下において、Data Integration Service は、サポートされている関数を Oracle ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、Data Integration Service は、Oracle ソースのトランスフォーメーションロジックを処理します。

- ADD_TO_DATE または GET_DATE_PART が、サブ秒の精度で結果を返す場合。
- ROUND が、秒またはサブ秒に値を丸める場合。
- SYSTIMESTAMP がミリ秒の精度で日付と時刻を返す場合。
- TRUNC が秒またはサブ秒を切り詰める場合。

ODBC 関数の例外

IN 関数の CaseFlag 引数がゼロ以外の値の場合、データ統合サービスは、ODBC のトランスフォーメーションロジックを処理します。

注: ODBC 接続オブジェクトのプロパティがデータベース固有の ODBC プロバイダを含む場合、データ統合サービスはソースをネイティブソースタイプと見なします。

接続オブジェクトで ODBC プロバイダを **【その他】** に指定した場合、データ統合サービスは EXP()関数を Teradata ソースにプッシュできません。EXP()関数をプッシュするには、ODBC プロバイダを **【Teradata】** に設定してください。

Sybase ASE 関数の例外

特定の条件下において、Data Integration Service は、サポートされている関数を Sybase ASE ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、Data Integration Service は、Sybase ASE ソースのトランスフォーメーションロジックを処理します。

- IN が CaseFlag 引数を含む場合。
- INSTR が 3 つ以上の引数を含む場合。
- LTRIM が 2 つ以上の引数を含む場合。
- RTRIM が 2 つ以上の引数を含む場合。

- TO_BIGINT が 2 つ以上の引数を含む場合。
- TO_INTEGER が 2 つ以上の引数を含む場合。
- TRUNC(Numbers)が 2 つ以上の引数を含む場合。

Teradata 関数の例外

特定の条件下において、データ統合サービスは、サポートされている関数を Teradata ソースにプッシュできません。

サポートされている関数が以下のロジックで式に含まれる場合、データ統合サービスは、Teradata ソースのトランスフォーメーションロジックを処理します。

- ADD_TO_DATE が、YEAR および MONTH 以外の属性を含む場合。
- IN が CaseFlag 引数を含む場合。
- INSTR が 3 つ以上の引数を含む場合。
- LTRIM が 2 つ以上の引数を含む場合。
- ROUND が 2 つ以上の引数を含む場合。
- RTRIM が 2 つ以上の引数を含む場合。

演算子

以下の表に、Informatica の演算子を使用できるかどうかをソースタイプ別に示します。各カラムには、データ統合サービスが演算子をソースにプッシュできるかどうかを示されます。

注: 非リレーショナルソースは、z/OS 上の IMS、VSAM、およびシーケンシャルデータセットです。

演算子	DB2 for LUW	DB2 for i5/OS または z/OS	Greenplum	Hive	Microsoft SQL Server	非リレーショナル	Oracle	SAP*	SAP HANA	Sybase ASE	Teradata
+ - *	○	○	○	○	○	○	○	×	○	○	○
/	○	○	○	○	○	×	○	×	○	○	○
%	○	○	○	○	○	×	○	×	○	○	○
	○	○	○	○	○	×	○	×	○	○	○
= > < >= <=	○	○	○	○	○	○	○	○	○	○	○

演算子	DB2 for LUW	DB2 for i5/OS または z/OS	Greenplum	Hive	Microsoft SQL Server	非リレーショナル	Oracle	SAP*	SAP HANA	Sybase ASE	Teradata
<>	○	○	○	○	○	×	○	○	○	○	○
!=	○	○	○	○	○	○	○	○	○	○	○
^=	○	○	○	○	○	×	○	○	○	○	○
AND OR	○	○	○	○	○	○	○	○	○	○	○
NOT	○	○	○	○	○	×	○	×	○	○	○

*データ統合サービスは、フィルタトランスフォーメーションロジックに含まれる場合にのみこれらの演算子をプッシュできます。

演算子	DB2 for LUW	Microsoft SQL Server	ODBC	Oracle	Sybase ASE
+ - *	○	○	○	○	○
/	○	○	○	○	○
%	○	○	×	○	○
	○	○	×	○	○
= > < >= <=	○	○	○	○	○
<>	○	○	○	○	○
!=	○	○	○	○	○
^=	○	○	○	○	○
AND OR	○	○	○	○	○
NOT	○	○	○	○	○

データ統合サービスとソースの出力の比較

データ統合サービスとソースでは、同じトランスフォーメーションロジックを処理した場合でも、異なる結果になる場合があります。データ統合サービスがトランスフォーメーションロジックをソースにプッシュした場合、トランスフォーメーションロジックの出力は異なる可能性があります。

トランスフォーメーションロジックの出力が異なる可能性があるのは、次のような場合です。

大文字小文字の区別

大文字小文字の区別の取り扱い方は、データ統合サービスとデータベースとで異なる場合があります。例えば、データ統合サービスは大文字小文字を区別するクエリが使用し、データベースは使用しない場合があります。フィルタトランスフォーメーションでフィルタ条件として `IIF(col_varchar2 = 'CA', TRUE, FALSE)` が使用されているとします。'CA' に一致する行を返すデータベースが必要です。ただし、大文字と小文字を区別しないデータベースにこのトランスフォーメーションロジックをプッシュすると、値 'Ca'、'ca'、'cA'、および 'CA' に一致する行が返されます。

数値が文字値に変換される

同じ数値を文字値に変換するにしても、データ統合サービスとデータベースとで変換形式が異なる可能性があります。データベースでは、数値を許容できない文字形式に変換されてしまう可能性があります。例えば、テーブルに数値 1234567890 が含まれているとします。データ統合サービスでこの数を文字値に変換した場合、文字「1234567890」が挿入されます。ただし、この数はデータベースで、「1.2E9」に変換されます。これら 2 つの文字列は同じ値を示しています。

TO_CHAR 関数および TO_DATE 関数の日付フォーマット

データ統合サービスは、TO_CHAR 関数または TO_DATE 関数をデータベースにプッシュする場合、関数で日付フォーマットが使用されます。日付または時刻の値を比較するには、TO_DATE 関数を使用します。TO_CHAR を使用して日付または時刻の値を比較する場合、1 桁の月、日、または 1 時間などの値に、スペースまたは先頭のゼロが追加されます。データベースがスペースまたは先頭のゼロを追加すると、データベースの比較結果はデータ統合サービスの結果と異なる可能性があります。

精度

データ統合サービスとデータベースとで、特定のデータ型の精度が異なる場合があります。トランスフォーメーションデータ型が使用するデフォルトの数値精度は、ネイティブデータ型とは異なる場合があります。データベースで使用されている精度がデータ統合サービスとは異なる場合、結果が変わる可能性があります。

SYSTIMESTAMP 関数

SYSTIMESTAMP を使用した場合、データ統合サービスはサービスプロセスを実行するノードの現在の日付と時刻を返します。しかし、トランスフォーメーションロジックをデータベースに渡した場合、データベースは、データベースのホストマシンの現在の日付と時刻を返します。データベースをホストしているマシンのタイムゾーンが、データ統合サービスを実行するマシンのタイムゾーンと同じでない場合、結果に相違が生じることがあります。

SYSTIMESTAMP を IBM DB2 または Sybase ASE データベースにプッシュし、SYSTIMESTAMP のフォーマットを指定した場合、データベースではそのフォーマットは無視され、完全なタイムスタンプが返されます。

LTRIM、RTRIM、または SOUNDEX 関数

LTRIM、RTRIM、または SOUNDEX をデータベースにプッシュした場合、引数 (') はそのデータベースでは NULL として扱われますが、データ統合サービスではスペースとして扱われます。

Oracle ソース上の LAST_DAY 関数

LAST_DAY を Oracle にプッシュした場合、秒までの日付が Oracle によって返されます。入力日付にサブ秒が含まれている場合、Oracle は日付を秒まで切り詰めます。

第 13 章

パーティション化されたマッピング

この章では、以下の項目について説明します。

- [パーティション化されたマッピングの概要, 249 ページ](#)
- [各パイプラインステージごとに 1 つのスレッド, 250 ページ](#)
- [各パイプラインステージごとに複数のスレッド, 251 ページ](#)
- [パーティション化されたフラットファイルソース, 253 ページ](#)
- [パーティション化されたリレーショナルソース, 254 ページ](#)
- [パーティション化されたフラットファイルターゲット, 256 ページ](#)
- [パーティション化されたリレーショナルターゲット, 260 ページ](#)
- [パーティション化されたトランスフォーメーション, 262 ページ](#)
- [パーティション化されたマッピングにおける順序の維持, 265 ページ](#)
- [マッピングの最大並行処理のオーバーライド, 266 ページ](#)
- [パーティション化されたマッピングのトラブルシューティング, 270 ページ](#)

パーティション化されたマッピングの概要

パーティション化オプションがある場合は、管理者はデータ統合サービスによりマッピング実行時の並列処理を最大化することができます。並列処理を最大化すると、データ統合サービスによって基になるデータが動的にパーティションに分割され、すべてのパーティションが同時に処理されます。

マッピングが大規模なデータセットを処理する場合、または複雑な計算を実行するトランスフォーメーションを含む場合は、マッピングの処理に時間がかかり、データのスループットが低下する可能性があります。これらのマッピングでパーティション化を有効にする場合、データ統合サービスは追加のスレッドを使用してマッピングを処理し、これによりパフォーマンスが最適化される可能性があります。

パーティション化を有効にするには、管理者と開発者が以下のタスクを実行します。

管理者は、データ統合サービスの最大並行処理を、Administrator ツールで 2 以上の値に設定します。

最大並行処理は、単一のパイプラインステージを処理する並行スレッドの最大数を決定します。管理者は、マッピングを実行するノードで使用可能な CPU の数に基づいて【最大並行処理】プロパティ値を増やします。

必要に応じて、開発者は、マッピングの並行処理の最大値を Developer ツールで変更できます。

デフォルトでは、各マッピングの【最大並行処理】プロパティは【自動】に設定されています。各マッピングは、データ統合サービスに定義されている最大並行処理値を使用します。

開発者は、マッピングランタイムプロパティ内の最大並行処理値を変更することにより、特定のマッピングの最大値を定義できます。データ統合サービスとマッピングで設定されている最大並行処理の整数値が異なる場合、データ統合サービスではこれらの最小値が使用されます。

マッピングでパーティション化が無効になっている場合、データ統合サービスはマッピングをパイプラインステージに区切り、1つのスレッドを使用して各ステージを処理します。

マッピングでパーティション化が有効になっている場合、データ統合サービスは、複数のスレッドを使用して各マッピングパイプラインステージを処理します。

データ統合サービスでは、入力および出力として物理データを持つマッピングにパーティションを作成できます。データ統合サービスは複数のパーティションを使用してマッピング実行中に次のアクションを完了できます。

- フラットファイル、IBM DB2 for LUW、または Oracle ソースからの読み取り。
- トランスフォーメーションの実行。
- フラットファイル、IBM DB2 for LUW、または Oracle ターゲットへの書き込み。

各パイプラインステージごとに1つのスレッド

最大並列処理数が1に設定されている場合、パーティション化は無効になっています。データ統合サービスは、マッピングを複数のパイプラインステージに分割し、1つのスレッドを使用して各ステージを処理します。

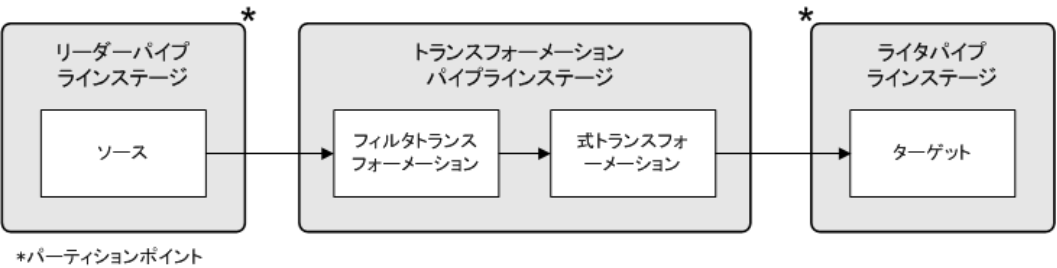
各マッピングには、1つ以上のパイプラインが含まれています。パイプラインは、読み取りトランスフォーメーションと、読み取りトランスフォーメーションからデータを受信するすべてのトランスフォーメーションで構成されています。データ統合サービスは、マッピングパイプラインをパイプラインステージに分けてから、各パイプラインステージの抽出、トランスフォーメーション、およびロードを並行して実行します。

パーティションポイントはパイプラインの境界をマークし、パイプラインをステージに分割します。どのマッピングパイプラインに対しても、データ統合サービスは、読み取りトランスフォーメーションの後、および書き込みトランスフォーメーションが複数のパイプラインステージを作成する前に、パーティションポイントを追加します。

各パイプラインステージは以下のいずれかのスレッドを実行します。

- データ統合サービスがソースからデータを抽出する方法を制御する読み取りスレッド。
- データ統合サービスがパイプラインのデータを処理する方法を制御するトランスフォーメーションスレッド。
- データ統合サービスがデータをターゲットにロードする方法を制御する書き込みスレッド。

以下の図に、読み取りパイプラインステージ、トランスフォーメーションパイプラインステージ、および書き込みパイプラインステージに区切られたマッピングを示します。



パイプラインには3つのステージが含まれるため、データ統合サービスでは3セットの行を同時に処理してマッピングのパフォーマンスを最適化することができます。例えば、読み取りスレッドが3番目の行セットを処理している間、トランスフォーメーションスレッドは2番目の行セットを処理し、書き込みスレッドは1番目の行セットを処理します。

以下の表に、複数のスレッドが3セットの行を同時に処理する方法を示します。

読み取りスレッド	トランスフォーメーションスレッド	書き込みスレッド
行セット 1	-	-
行セット 2	行セット 1	-
行セット 3	行セット 2	行セット 1
行セット 4	行セット 3	行セット 2
行セット n	行セット (n-1)	行セット (n-2)

マッピングパイプラインに複雑な計算を実行するトランスフォーメーションが含まれている場合、トランスフォーメーションパイプラインステージの処理には時間がかかる可能性があります。パフォーマンスを最適化するには、一部のトランスフォーメーションが追加のトランスフォーメーションパイプラインステージを作成する前に、データ統合サービスがパーティションポイントを追加します。

各パイプラインステージごとに複数のスレッド

最大並列処理が2以上の値に設定されている場合、パーティション化が有効になっています。データ統合サービスはマッピングをパイプラインステージに分割し、複数のスレッドを使用して各ステージを処理します。任意のパイプラインステージにおけるスレッドの数は、そのステージにおけるパーティションの数と一致します。

並行処理を最大にすると、データ統合サービスは実行時に以下のタスクを動的に実行します。

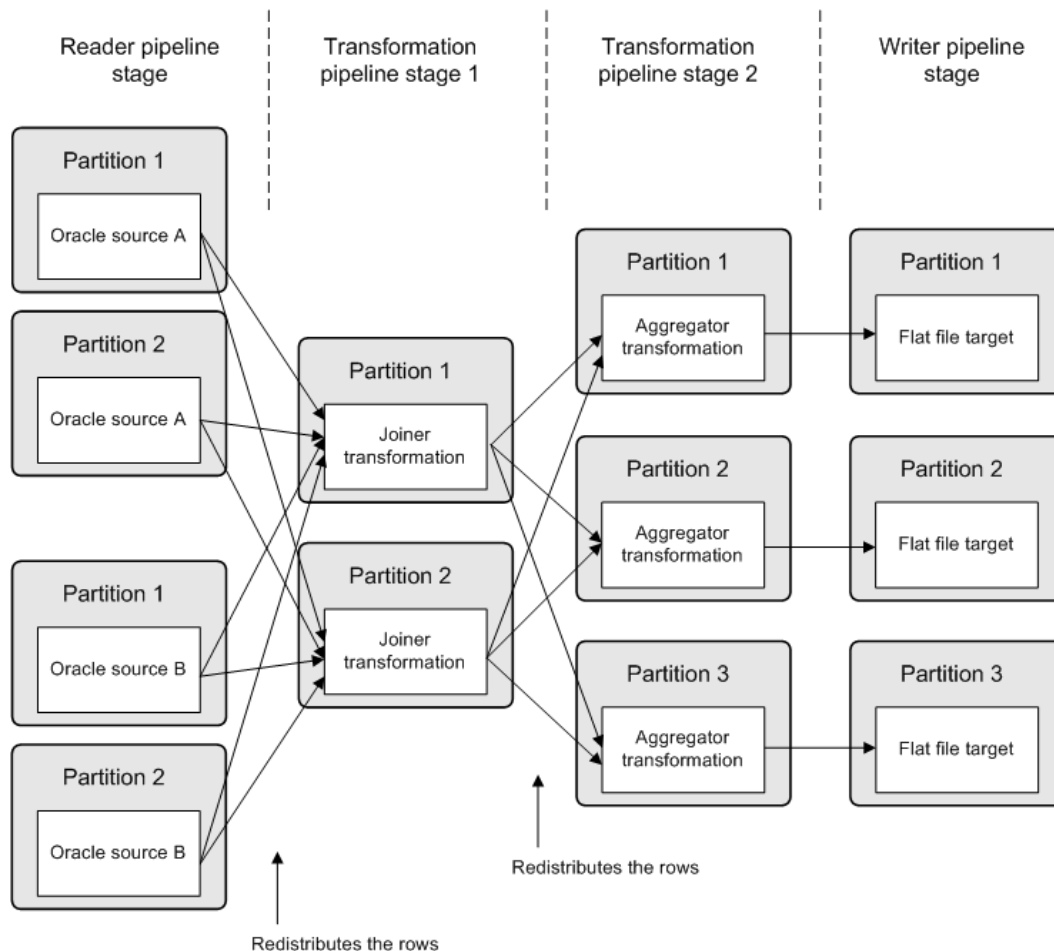
データをパーティションに分割する。

データ統合サービスは基になるデータを動的にパーティションに分割し、各パーティションを同時に実行します。データ統合サービスは、各パイプラインステージの最適なスレッド数を判断します。1つのパイプラインステージで使用されるスレッドの数は、最大並列処理数を超えることはできません。データ統合サービスは、パイプラインステージごとに異なる数のスレッドを使用できます。

パーティションポイント全体にわたってデータを再分散する。

データ統合サービスは、トランスフォーメーション要件に基づいて、パーティションポイント全体にわたってデータを再分配する最善の方法を動的に決定します。

以下の図に、パイプラインステージごとに複数のパーティションにデータを分配するマッピングの例を示します。



上図では、データ統合サービスの最大並列処理の値は3です。マッピングの最大並列処理は自動です。データ統合サービスは、マッピングを4つのパイプラインステージに分け、合計12個のスレッドを使用してマッピングを実行します。データ統合サービスは、各パイプラインステージで以下のタスクを実行します。

- 読み取りパイプラインステージでは、データ統合サービスは、Oracle データベースシステムに問い合わせ、ソース A とソース B の両方のソーステーブルに2つのデータベースパーティションが存在することを確認します。データ統合サービスは、各データベースパーティションについて1つの読み取りスレッドを使用します。
- 最初のトランスフォーメーションパイプラインステージで、データ統合サービスは、データを再配分して結合条件を満たす行を2つのスレッドに分割します。
- 2番目のトランスフォーメーションパイプラインステージで、データ統合サービスは、このアグリゲータトランスフォーメーションには3つのスレッドが最適であると判断します。そして、データを再配分して、集計式を満たす行を3つのスレッドに分割します。
- 書き込みパイプラインステージで、データ統合サービスは、ターゲットパーティションポイント全体にわたって行を再配分する必要がありません。単一パーティション内のすべての行は、ターゲットパーティションポイントを通じて、そのパーティション内に留まります。

パーティション化されたフラットファイルソース

パーティション化が有効になっているマッピングでフラットファイルソースを読み取る場合、データ統合サービスは、複数のスレッドを使用してファイルソースを読み取ることができます。

データ統合サービスは、以下のフラットファイルソースタイプ用のパーティションを作成できます。

- 直接ファイル
- 間接ファイル
- ファイルのディレクトリ
- コマンド
- Hadoop 分散ファイルシステム (HDFS) のファイルまたはファイルのディレクトリ

データ統合サービスで複数のスレッドを使用してファイルソースを読み取る場合、そのソースに対して複数の同時接続が作成されます。デフォルトでは、データ統合サービスによって行の順番が保持されることはありません。ファイルやファイルリストの行を順番に読み取ることがないためです。複数のスレッドで1つのファイルソースを読み取る場合に行順序を保持するには、同時読み取りのパーティション化を設定します。

データ統合サービスで複数のスレッドを使用して直接ファイルを読み取る場合は、複数の読み取りスレッドが作成され、ファイルが同時に読み取られます。

データ統合サービスで、複数のスレッドを使用して間接ファイルまたはファイルのディレクトリを読み取る場合は、複数の読み取りスレッドが作成され、リストまたはディレクトリのファイルが同時に読み取られます。データ統合サービスで複数のスレッドを使用して1つのファイルが読み取られる場合もあります。また、データ統合サービスで1つのスレッドを使用して、リストやディレクトリの複数のファイルが読み取られる場合もあります。

同時読み込みのパーティション化

複数のスレッドが単一のファイルソースから読み取りを行う場合に、行の順序を保持するには、フラットファイルデータオブジェクトの **【同時読み込みのパーティション化】** プロパティを設定して順序を保持します。

フラットファイルデータオブジェクトの **【詳細】** プロパティの **【同時読み取りのパーティション化】** プロパティを設定します。 **【ランタイム: 読み取り】** セクションでプロパティを見つけます。

【同時読み込みのパーティション化】 プロパティの以下のいずれかのオプションを選択します。

スループットの最適化

複数のパーティションが1つのファイルソースから読み取る場合、データ統合サービスでは行の順番が保持されません。このオプションは、複数のパーティションが1つのファイルソースから読み込む順序が重要でない場合に使用します。

デフォルトのオプションです。

相対順序を維持する

各パーティションが読み取る入力行のソート順が保持されます。

以下の表に、2つのパーティションが読み取る10行を含むファイルソースのソート順の例を示します。

パーティション	読み込まれる行
Partition #1	1、3、5、8、9
Partition #2	2、4、6、7、10

絶対順序を維持する

すべてのパーティションが読み取るすべての入力行のソート順が保持されます。バッチトランスフォーマーションを含むパススルーマッピングでは、ターゲットに書き込まれる行の順序は、入力行と同じ順序になります。

以下の表に、2つのパーティションが読み取る10行を含むファイルソースのソート順の例を示します。

パーティション	読み込まれる行
Partition #1	1、2、3、4、5
Partition #2	6、7、8、9、10

パーティション化されたリレーショナルソース

パーティション化が有効になっているマッピングで IBM DB2 for LUW または Oracle ソースから読み取りを行う場合、データ統合サービスは複数のスレッドを使用してリレーショナルソースを読み取ることができます。データ統合サービスは、各スレッド用にデータベースへの個別の接続を作成します。

注: マッピングが DB2 for LUW または Oracle 以外のリレーショナルソースから読み取りを行う場合、データ統合サービスは1つのスレッドを使用してソースから読み取りを行います。データ統合サービスは、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。

データ統合サービスでは DB2 for LUW または Oracle データベースシステムに対してパーティション情報のクエリを実行します。ソーステーブルでデータベースのパーティション化がサポートされている場合、データ統合サービスでは、複数のスレッドを使用してパーティション化されたデータを、データベースの対応するノードから読み取ることができます。データ統合サービスでは、各 reader スレッドに対して SQL クエリが生成されます。

データ統合サービスが使用する reader スレッドの数は以下の状況によって異なります。

データベースパーティションの数は、最大並行処理値以下です。

データ統合サービスは、各データベースパーティションについて1つの読み取りスレッドを使用します。データ統合サービスは1つのデータベースパーティションを各 reader スレッドに分散します。

複合パーティション化を使用する Oracle ソースの場合、データ統合サービスは各データベースのサブパーティションに1つの読み取りスレッドを使用します。例えば、Oracle ソースに3つのパーティションが含まれ、各パーティションに2つのサブパーティションが含まれる場合、データ統合サービスは6つの読み取りスレッドを使用します。

データベースパーティションの数が並列処理の最大値を超えている。

データ統合サービスは、最大並行処理値で定義されている数の読み取りスレッドを使用します。データ統合サービスは複数のデータベースパーティションを一部の reader スレッドに分散します。例えば、DB2 for LUW ソースに5つのデータベースパーティションがあり、最大並行処理値が3に設定されているとします。データ統合サービスは3つの読み取りスレッドを使用します。データ統合サービスは、2つのデータベースパーティションを、1番目の reader スレッドと2番目の reader スレッドに分散します。サービスは1つのデータベースパーティションを3番目の reader スレッドに分散します。

データベースパーティションがない。

データ統合サービスは1つのスレッドを使用してソースから読み取りを行います。データ統合サービスは、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。

パーティション化のリレーショナル接続タイプ

データ統合サービスでは複数のスレッドを使用して、データベースへの接続で使用する接続タイプに基づいて、DB2 for LUW または Oracle のリレーショナルソースを読み取ることができます。

DB2 for LUW または Oracle のデータベースに接続するには、以下のどの接続タイプでも使用できます。

- DB2 for LUW 接続または Oracle 接続
- JDBC 接続
- ODBC 接続

複数のスレッドを使用して DB2 for LUW または Oracle のリレーショナルソースを読み取るには、リレーショナルデータオブジェクトが DB2 for LUW または Oracle の接続を使用する必要があります。

DB2 for LUW または Oracle のリレーショナルデータオブジェクトが JDBC または ODBC 接続を使用する場合、データ統合サービスはスレッドを 1 つ使用してソースを読み取ります。データ統合サービスは、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。

パーティション化されたリレーショナルソースの SQL クエリ

データ統合サービスは、複数のスレッドを使用してリレーショナルソースを読み取る際に、読み取りスレッドごとに SQL クエリを生成します。

データベースソースのデータベースパーティションの方が並列処理の最大値よりも多い場合、データ統合サービスは reader スレッド全体にわたってデータを分散します。データ統合サービスは、複数のデータベースパーティションから読み取る SQL クエリを生成できます。Oracle ソースにサブパーティションが含まれる場合、データ統合サービスは複数のデータベースサブパーティションから読み取る SQL クエリを生成できます。

DB2 for LUW または Oracle ソースの例

最大並行処理値は 3 で、リレーショナルソースには 5 つのデータベースパーティションがあります。データ統合サービスでデータベースパーティションに対して SQL クエリが実行されている場合、最初と 2 番目の reader スレッドは 2 つのデータベースパーティションからデータを受け取ります。3 番目の reader スレッドは、1 つのデータベースパーティションからデータを受け取ります。この例では、読み取りトランスフォーメーション内の単純クエリで個別選択オプションが有効になっていません。

DB2 for LUW ソースを使用する場合は、データ統合サービスで、最初の reader スレッドに対して以下の文と同様の SQL 文が生成されます。

```
SELECT <column list> FROM <table name>  
WHERE (nodenumber(<column 1>)=0 OR nodenumber(<column 1>) = 3)
```

Oracle ソースを使用する場合は、データ統合サービスで、最初の reader スレッドに対して以下の文と同様の SQL 文が生成されます。

```
SELECT <column list> FROM <table name> PARTITION <database_partition1 name> UNION ALL  
SELECT <column list> FROM <table name> PARTITION <database_partition4 name> UNION ALL
```

Oracle ソースとサブパーティションの例

Oracle ソースには、1 から 5 の 5 つのパーティションと、各パーティションに a と b の 2 つのサブパーティションがあります。最大並行処理値は 3 です。最初の reader スレッドは、4 つのデータベースサブパーティションからデータを受け取ります。2 番目と 3 番目の reader スレッドは、3 つのデータベースサブパーティションからデータを受け取ります。この例では、読み取りトランスフォーメーション内の単純クエリで個別選択オプションが有効になっていません。

データ統合サービスで、最初の reader スレッドに対して、以下の文と同様の SQL 文が生成されます。

```
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_a name> UNION ALL  
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_b name> UNION ALL
```



```
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_a name> UNION ALL  
SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_b name> UNION ALL
```

リレーショナルソースパーティションに関するルールおよびガイドライン

リレーショナルソースからの読み取りを実行するマッピングのパーティション化を有効にする場合は、以下のルールおよびガイドラインを考慮してください。

- データ統合サービスは、1 スレッドを使用してソースを読み取りますが、以下の状況では、複数のスレッドを使用して残りのマッピングパイプラインステージを処理できます。
 - マッピングが DB2 for LUW または Oracle 以外のリレーショナルソースから読み取る。
 - マッピングが JDBC または ODBC 接続を使用して DB2 for LUW または Oracle ソースから読み取る。
 - マッピングがトランスフォーメーションロジックをソースデータベースにプッシュする。
 - 読み取りトランスフォーメーションで単純クエリを使用して、ソートの基準となるポートを選択するか、ユーザー定義の結合を設定する。
 - 読み取りトランスフォーメーションで詳細クエリを使用して、カスタム SQL クエリを作成する。
- 読み取りトランスフォーメーションで単純クエリを使用してヒントを作成、異なる値を選択、またはソースフィルタを入力する場合は、データ統合サービスで複数のスレッドを使用してソースを読み取ることができます。データ統合サービスで、ヒント、異なる値、またはソースフィルタが、各パーティション用に生成された SQL クエリに追加されます。

パーティション化されたフラットファイルターゲット

パーティション化が有効になっているマッピングがフラットファイルターゲットに書き込む場合、データ統合サービスは複数のスレッドを使用してファイルターゲットに書き込むことができます。

データ統合サービスは、Hadoop 分散ファイルシステム（HDFS）にフラットファイルまたはファイルのパーティションを作成できます。

ファイル出力タイプまたはコマンド出力タイプを持つようにフラットファイルデータオブジェクトを設定できます。フラットファイルデータオブジェクトがファイル出力タイプである場合、データ統合サービスはターゲットデータをフラットファイルに書き込みます。複数のスレッドがフラットファイルターゲットに書き込むと、各スレッドは個別のファイルに対してターゲット出力を書き込みます。各スレッドは以下の形式でファイルに名前を付けます。

```
<output_file_name><partition_number>.out
```

例えば、3 つのスレッドが MyOutput1.out、MyOutput2.out、および MyOutput3.out という名前のファイルに書き込むとします。

パフォーマンスが最適化されるように複数の出力ファイルディレクトリを設定したり、単一のマージファイルに書き込むようにフラットファイルデータオブジェクトを設定したりできます。

フラットファイルデータオブジェクトがコマンド出力タイプである場合、データ統合サービスはターゲットデータをフラットファイルまたはマージファイルの代わりにコマンドまたはマージコマンドに出力します。複数のパーティションがフラットファイルターゲットに書き込む場合は、単一のパーティションのターゲットデータを処理するか、すべてのターゲットパーティションのマージデータを処理するようにコマンドを設定できます。

パーティション化されたファイルターゲットに対応するための出力ファイルディレクトリの最適化

デフォルトでは、フラットファイルデータオブジェクトの出力タイプがファイルの場合、各スレッドがターゲット出力を個々のファイルに書き込みます。複数のスレッドで1つのファイルターゲットに書き込む場合に最適なパフォーマンスを実現するには、複数の出力ファイルディレクトリを設定します。

複数のスレッドが単一のディレクトリに書き込む場合、入力/出力 (I/O) 競合によりマッピングにボトルネックが発生することがあります。I/O 競合は、複数のスレッドがファイルシステムに同時にデータを書き込む場合に発生する可能性があります。

複数のディレクトリを設定すると、ラウンドロビン方式で各スレッドに対して出力ディレクトリが割り当てられます。例えば、directoryA と directoryB をターゲットディレクトリとして使用するようフラットファイルデータオブジェクトを設定したとします。データ統合サービスがファイルターゲットへの書き込みに4つのスレッドを使用する場合、1つ目と3つ目の書き込みスレッドはターゲットファイルを directoryA に書き込みます。2つ目と4つ目の書き込みスレッドは、ターゲットファイルを directoryB に書き込みます。

データ統合サービスがターゲットへの書き込みに複数のスレッドを使用しない場合は、出力ファイルは、最初の一覧表示されたディレクトリに書き込まれます。

フラットファイルデータオブジェクトの **【詳細】** プロパティで、出力ファイルディレクトリを設定します。**【ランタイム: 書き込み】** セクションで、**【出力ファイルディレクトリ】** プロパティを見つけます。デフォルトでは、このプロパティは、データ統合サービスに対して定義されているシステムパラメータ値を使用するように設定されます。データ統合サービスの **【ターゲットディレクトリ】** プロパティ用に管理者がセミコロンで区切られた複数のディレクトリを入力した場合は、デフォルトの TargetDir システムパラメータ値を使用してください。

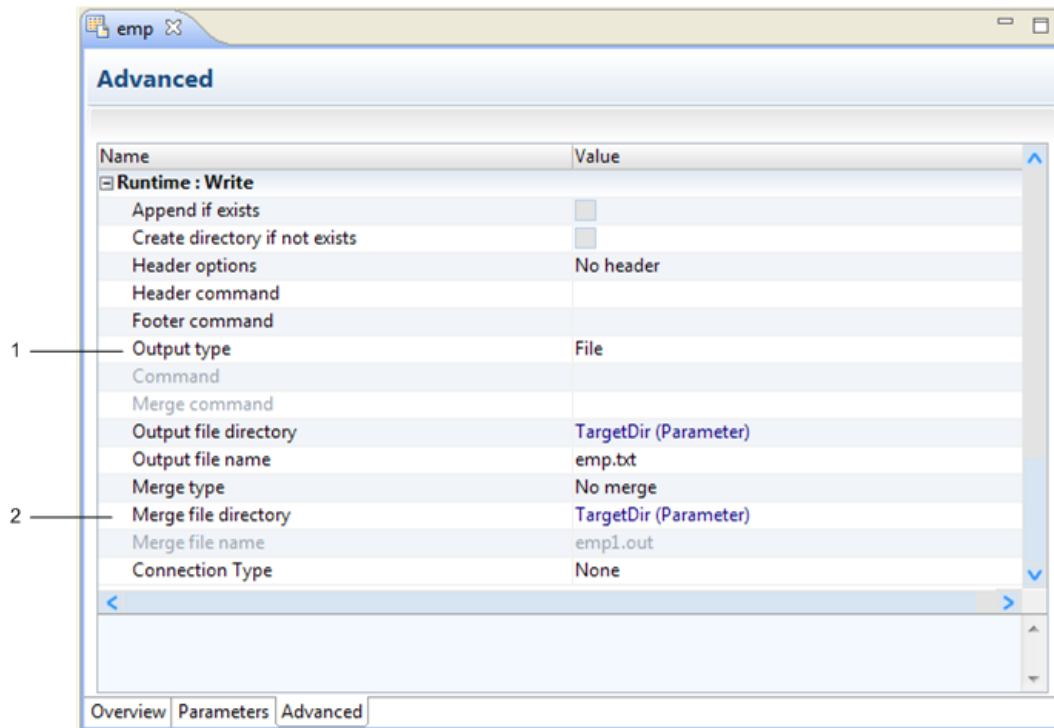
異なる値を入力し、フラットファイルデータオブジェクトに固有の出力ファイルディレクトリを複数設定できます。プロパティ、またはプロパティに割り当てられたユーザー定義のパラメータ用に、セミコロンで区切られた複数のディレクトリを入力します。

パーティション化されたファイルターゲットのマージオプション

デフォルトでは、フラットファイルデータオブジェクトの出力タイプがファイルの場合、各スレッドがターゲット出力を個々のファイルに書き込みます。パーティションのターゲットデータはマージできます。ターゲットデータをマージすると、データ統合サービスはすべてのターゲットパーティションに対して1つのマージファイルを作成します。

マージオプションは、フラットファイルデータオブジェクトの **【詳細】** プロパティで設定します。**【ランタイム: 書き込み】** セクションで、マージプロパティを見つけます。

以下の図に、フラットファイルデータオブジェクトの「詳細」プロパティのマージオプションを示します。



1. ファイル出力タイプ
2. マージオプション

【マージタイプ】 プロパティには、以下のいずれかのオプションを選択します。

マージなし

データ統合サービスは、ターゲット出力を各パーティションの個別のファイルに同時に書き込みます。

デフォルトのオプションです。

シーケンシャル

データ統合サービスは各パーティションに対して出力ファイルを作成してから、それらを1つのマージファイルにマージします。データ統合サービスは、出力ファイル名と出力ファイルディレクトリ値を使用して個々のターゲットファイルを作成します。データ統合サービスは、各書き込みスレッドの完了順に、各パーティションの出力データをマージファイルに順次追加します。例えば、パーティション2の書き込みスレッドがパーティション1のスレッドよりも前に完了した場合、データ統合サービスは、パーティション2、パーティション1の順序でデータをマージファイルに追加します。

ファイルリスト

データ統合サービスは、各パーティションに対してターゲットファイルを作成し、個々のファイルのパスを含むファイルリストを作成します。データ統合サービスは、出力ファイル名と出力ファイルディレクトリ値を使用して個々のターゲットファイルを作成します。ターゲットファイルをマージディレクトリまたはマージディレクトリの下ディレクトリに書き込んだ場合、ファイルリストには相対パスが含まれます。それ以外の場合は、ファイルリストに絶対パスが含まれます。別のマッピングでターゲットファイルをソースファイルとして使用する場合は、このファイルをソースファイルとして使用します。

同時

データ統合サービスは、すべてのターゲットパーティションのデータをマージファイルに同時に書き込みます。パーティションごとに中間ファイルが作成されることはありません。データ統合サービスはすべて

のパーティションのデータをマージファイルに同時に書き込むため、マージファイル内のデータの順序が連続していない可能性があります。

フラットファイルデータオブジェクトを設定してターゲットデータをマージする場合、オプションで **【マージファイルディレクトリ】** および **【マージファイル名】** プロパティのデフォルト値を編集できます。

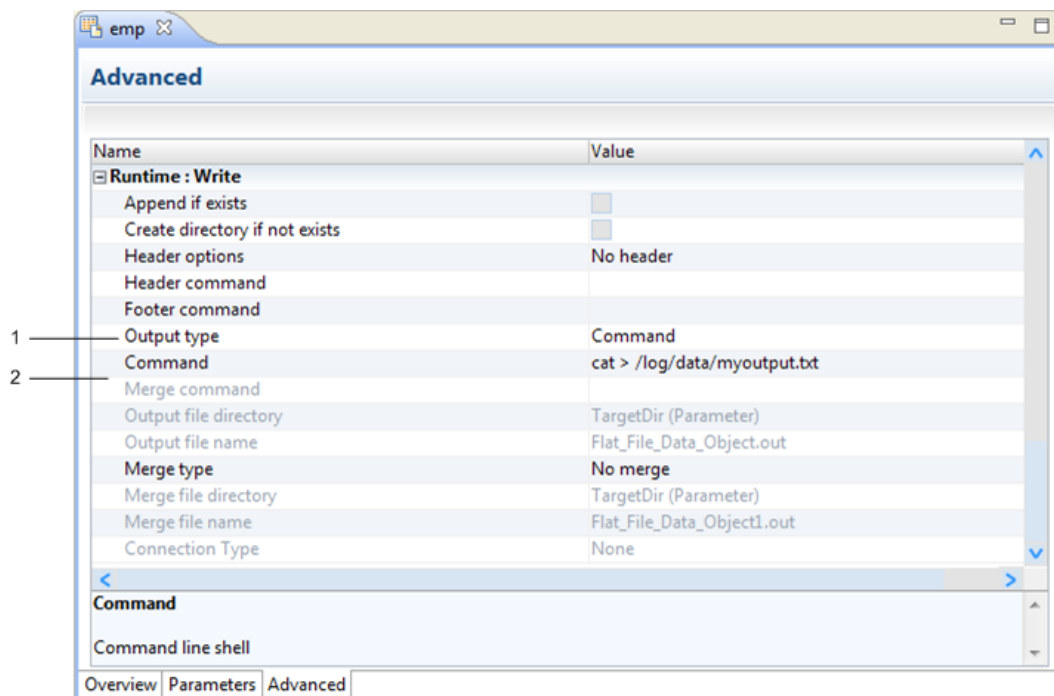
フラットファイルデータオブジェクトを設定してターゲットデータをマージし、データ統合サービスがターゲット用のパーティションを作成しない場合、データ統合サービスはマージオプションを無視します。サービスは **【出力ファイル名】** および **【出力ファイルディレクトリ】** プロパティで定義されたファイルに書き込みます。

パーティション化されたファイルターゲットのコマンド

フラットファイルデータオブジェクトの出力タイプがコマンドの場合、コマンドを使用して、1つのパーティションのターゲットデータの処理や、マッピング内のすべてのターゲットパーティションのマージデータの処理が行えます。データ統合サービスはデータを、フラットファイルやマージファイルではなく、コマンドまたはマージコマンドに送信します。

フラットファイルデータオブジェクトの **【詳細】** プロパティで、パーティションのデータを処理するコマンドを設定します。**【ランタイム: 書き込み】** セクションで、コマンドプロパティを見つけます。

以下の図に、コマンドを使用して1つのパーティションのターゲットデータを処理するように設定されたフラットファイルデータオブジェクトを示します。



1. コマンド出力タイプ
2. コマンドのオプション

UNIX では、有効な任意の UNIX コマンドまたはシェルスクリプトを使用します。Windows では、有効な任意の DOS またはバッチファイルを使用します。

コマンドを使用して次のタイプのターゲットデータを処理できます。

単一パーティションのデータ

各 writer スレッドが個別に実行するコマンドを入力します。各スレッドは、マッピングが実行されるときにターゲットデータをコマンドに送信します。各スレッドは、異なるデータセットで同じコマンドを実行します。

コマンドを入力するときは、マッピングを実行するオペレーティングシステムを考慮する必要があります。例えば、コマンド `cat > /log/data/myoutput.txt` を入力すると、複数のスレッドが同じファイルに書き込むため、オペレーティングシステムのエラーが発生する可能性があります。コマンド `cat >> /log/data/myoutput.txt` を入力すると、複数のスレッドがデータを同じファイルに付加するため、オペレーティングシステムエラーが発生する可能性は少なくなります。

単一パーティションのターゲットデータをコマンドに送信するには、**【出力タイプ】** プロパティで **【コマンド】** を選択し、**【マージタイプ】** プロパティで **【マージなし】** を選択します。**【コマンド】** プロパティのコマンドを入力します。

すべてのパーティションのデータのマージ

すべての writer スレッドのデータのマージを処理するマージコマンドを入力します。データ統合サービスは、マージデータを処理するためのコマンドに、コンカレントマージタイプを使用する必要があります。マッピングを実行するときに、各スレッドは同時にターゲットデータをマージコマンドに送信します。マージコマンドは、すべてのデータで 1 回実行します。コマンドでは、ターゲットデータの順序が維持されない場合があります。

すべてのパーティションのマージデータをマージコマンドに送信するには、**【出力タイプ】** プロパティで **【コマンド】** を選択し、**【マージタイプ】** プロパティで **【コンカレント】** を選択します。**【マージコマンド】** プロパティのコマンドを入力します。

パーティション化されたリレーショナルターゲット

パーティション化が有効になっているマッピングが IBM DB2 for LUW または Oracle ターゲットへ書き込みを行う場合、データ統合サービスは複数スレッドを使用してリレーショナルターゲットに書き込むことができます。データ統合サービスは、各スレッド用にデータベースへの個別の接続を作成します。

注: マッピングが DB2 for LUW または Oracle 以外のリレーショナルターゲットに書き込む場合、データ統合サービスは 1 つのスレッドを使用してターゲットに書き込みます。データ統合サービスは、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。

ターゲットが Oracle である場合、データ統合サービスは最大並行処理値で定義されている数の書き込みスレッドを使用します。Oracle リレーショナルテーブルにパーティションがある場合、このデータベースはデータを正しいパーティションにルーティングします。

ターゲットが DB2 for LUW の場合、データ統合サービスはパーティション情報に DB2 for LUW システムを必要とします。データ統合サービスはパーティション化されたデータをターゲットデータベース内の対応するノードにロードします。

データ統合サービスが DB2 for LUW ターゲットに使用する writer スレッドの数は、以下の状況によって異なります。

データベースパーティションの数は、最大並行処理値以下です。

データ統合サービスは、データベースパーティションごとに 1 つの書き込みスレッドを使用します。各 writer スレッドは 1 つのデータベースパーティションに書き込みます。

データベースパーティションの数が並列処理の最大値を超えている。

データ統合サービスは、最大並行処理値で定義されている数の書き込みスレッドを使用します。データ統合サービスは複数のデータベースパーティションを一部の writer スレッドに分散します。例えば、DB2 for LUW ターゲットに 5 つのデータベースパーティションがあり、最大並行処理値が 3 に設定されているとします。データ統合サービスは 3 つの書き込みスレッドを使用します。データ統合サービスは、2 つのデータベースパーティションを、1 番目の writer スレッドと 2 番目の writer スレッドに分散します。サービスは 1 つのデータベースパーティションを 3 番目の writer スレッドに分散します。

データベースパーティションがない。

データ統合サービスは、最大並行処理値で定義されている数の書き込みスレッドを使用します。

ただし、ターゲットにデータベースパーティションがある場合は、ロードパフォーマンスを最適化することができます。この場合、各 writer スレッドは、データベースパーティションを含む DB2 for LUW ノードに接続します。すべてのスレッドが単一のマスターノードに接続するのではなく、writer スレッドが別々の DB2 for LUW ノードに接続するため、パフォーマンスは向上します。

パーティション化のリレーショナル接続タイプ

データ統合サービスは複数のスレッドを使用して、データベースへの接続で使用する接続タイプに基づいて、DB2 for LUW または Oracle のリレーショナルターゲットに書き込むことができます。

DB2 for LUW または Oracle のデータベースに接続するには、以下のどの接続タイプでも使用できます。

- DB2 for LUW 接続または Oracle 接続
- JDBC 接続
- ODBC 接続

複数のスレッドを使用して DB2 for LUW または Oracle のリレーショナルターゲットに書き込むには、リレーショナルデータオブジェクトが DB2 for LUW 接続または Oracle 接続を使用する必要があります。

DB2 for LUW または Oracle のリレーショナルデータオブジェクトが JDBC 接続または ODBC 接続を使用する場合、データ統合サービスは 1 つのスレッドを使用してターゲットに書き込みます。データ統合サービスは、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。

リレーショナルターゲットパーティションに関するルールおよびガイドライン

リレーショナルターゲットへの書き込みを行うマッピングのパーティション化を有効にする場合は、以下のルールとガイドラインを考慮してください。

- データ統合サービスは、1 つのスレッドを使用してターゲットに書き込みますが、以下の状況では、残りのマッピングパイプラインステージ用に複数のスレッドを使用できます。
 - マッピングが DB2 for LUW や Oracle 以外のリレーショナルターゲットに書き込む。
 - マッピングが JDBC または ODBC 接続を使用して DB2 for LUW または Oracle のターゲットに書き込む。
- DB2 for LUW ターゲットテーブルパーティションキーが小数のカラムである場合は、マッピングの高精度を有効にします。パーティションキーが小数のカラムであり、マッピングに対して高精度を有効にしない場合は、データ統合サービスでセッションが失敗することがあります。

パーティション化されたトランスフォーメーション

パーティション化が有効になっているマッピングにパーティション化をサポートする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複数のスレッドを使用してそのトランスフォーメーションを実行できます。

データ統合サービスはトランスフォーメーションにパーティションポイントを追加する必要があるかどうか決定し、そのトランスフォーメーションのパイプラインステージの最適なスレッド数を決定します。また、データ統合サービスはパーティションポイントでデータを再分散する必要があるかどうかを決定します。例えば、データ統合サービスはアグリゲータトランスフォーメーションでデータを再分散して集計式の行をグループ化することがあります。

一部のトランスフォーメーションは、パーティション化をサポートしません。パーティション化が有効になっているマッピングにパーティション化をサポートしない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1つのスレッドを使用してそのトランスフォーメーションを実行します。データ統合サービスは、複数のスレッドを使用して残りのマッピングパイプラインステージを実行できます。

以下のトランスフォーメーションは、パーティション化をサポートしません。

- 関連付け
- 統合
- 例外
- 一致（フィールド一致分析に設定されている場合）
- REST Web サービスコンシューマ
- 未接続のルックアップ
- Web サービスコンシューマ

パーティション化されたトランスフォーメーションに関する制限

パーティション化をサポートしている一部のトランスフォーメーションには、特有の設定が必要です。パーティション化が有効になっているマッピングにサポートされていない設定がある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1つのスレッドを使用してそのトランスフォーメーションを実行します。データ統合サービスは複数のスレッドを使用して残りのマッピングパイプラインステージを処理します。

以下のトランスフォーメーションには、パーティション化をサポートする特有の設定が必要です。

- アグリゲータトランスフォーメーションには、グループ化ポートを含める必要があります。アグリゲータトランスフォーメーションにパススルーポートを含めることはできません。アグリゲータトランスフォーメーションに、行ごとに現在の合計と平均を計算する数値関数を含めることはできません。
- 式トランスフォーメーションに、以下のタイプの関数または変数を含めることはできません。
 - 行ごとに現在の合計と平均を計算する数値関数。
 - 複数のスレッドでトランスフォーメーションが処理された場合に、異なる結果を返す可能性がある特殊関数。
 - 前の行の値に依存するローカル変数。
- ディジション、Java、および SQL トランスフォーメーションでは、**【パーティション化可能】** プロパティを有効にする必要があります。
- ジョイナトランスフォーメーションには、等価演算子を使用する結合条件が含まれていることが必要です。結合条件に複数の等価条件が含まれる場合、各条件は AND 演算子を使用して結合する必要があります。
- ランクトランスフォーメーションには、グループ化ポートを含める必要があります。

トランスフォーメーションのキャッシュのパーティション化

キャッシュのパーティション化では、各パーティションごとに、アグリゲータ、ジョイナ、ランク、ルックアップ、またはソータの各トランスフォーメーションを処理する個別のキャッシュが作成されます。キャッシュのパーティション化中、各パーティションは異なるデータを別々のキャッシュに保存します。各キャッシュには、そのパーティションが必要とする行が含まれます。

各スレッドが個々のキャッシュに並行してクエリを実行するため、キャッシュをパーティション化することでマッピングのパフォーマンスが最適化されます。データ統合サービスがマッピング用のパーティションを作成する場合、データ統合サービスは、パーティション化されたアグリゲータ、ジョイナ、ランク、およびソータートランスフォーメーションに対して常にキャッシュのパーティション化を使用します。データ統合サービスは、パーティション化されたルックアップトランスフォーメーションに対しては、キャッシュのパーティション化を使用する場合があります。

データ統合サービスは、以下の状況で、接続されたルックアップトランスフォーメーションに対してキャッシュのパーティション化を使用します。

- ルックアップ条件に等価演算子だけが含まれる。
- 接続されたルックアップトランスフォーメーションがリレーショナルテーブルのデータを検索する場合、データベースは大文字と小文字を区別して比較するように設定されています。

例えば、ルックアップ条件に文字列ポートが含まれ、データベースが大文字と小文字を区別して比較するように設定されていない場合、データ統合サービスはキャッシュのパーティション化を使用しません。

データ統合サービスがルックアップトランスフォーメーションに対してキャッシュのパーティション化を使用しない場合、ルックアップトランスフォーメーションを実行するすべてのスレッドは同じキャッシュを共有します。各スレッドは同じキャッシュに連続してクエリを実行します。

注: データ統合サービスは、1つのスレッドを使用して未接続のルックアップトランスフォーメーションを実行するため、未接続のルックアップトランスフォーメーションにキャッシュのパーティション化を使用しません。

パーティション化されたキャッシュのキャッシュサイズ

データ統合サービスはアグリゲータ、ジョイナ、ランク、ルックアップ、およびソータートランスフォーメーションに対してキャッシュのパーティション化を使用する場合、パーティション全体にキャッシュサイズを分割します。

キャッシュサイズは、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。キャッシュサイズはバイト数で入力するか、**【自動】**を選択してデータ統合サービスによって実行時に自動で決定されるようにすることができます。

サイズを数値で指定した場合、データ統合サービスは実行時にトランスフォーメーションスレッド数でキャッシュサイズを分割します。例えば、トランスフォーメーションキャッシュサイズを 2,000,000 バイトに設定したとします。データ統合サービスは 4つのスレッドを使用してトランスフォーメーションを実行します。各スレッドが最大 500,000 バイトをキャッシュサイズとして使用するよう、キャッシュサイズ値が分割されます。

【自動】を選択すると、データ統合サービスはトランスフォーメーションのキャッシュサイズを実行時に決定します。次に、そのキャッシュサイズをトランスフォーメーションスレッドの数で分割します。

パーティション化に対応するためのキャッシュディレクトリの最適化

アグリゲータ、ジョイナ、ランク、およびソータートランスフォーメーションでキャッシュのパーティション化を行うときのパフォーマンスを最適化するため、複数のキャッシュディレクトリを設定します。

データ統合サービスがキャッシュのパーティション化を使用し、オーバーフローした値をキャッシュファイルに格納する必要がある場合、トランスフォーメーションスレッドはキャッシュディレクトリに書き込みを行います。複数のスレッドが単一のディレクトリに書き込む場合、入力/出力 (I/O) 競合によりマッピングにボト

ルネックが発生することがあります。I/O 競合は、複数のスレッドがファイルシステムに同時にデータを書き込む場合に発生する可能性があります。

複数のキャッシュディレクトリを設定すると、データ統合サービスは各トランスフォーメーションスレッドのキャッシュディレクトリをラウンドロビン方式で決定します。例えば、ディレクトリ A とディレクトリ B をキャッシュディレクトリとして使用するアグリゲータトランスフォーメーションを設定するとします。このアグリゲータトランスフォーメーションを実行するためにデータ統合サービスが 4 つのスレッドを使用する場合、最初と 3 つ目のトランスフォーメーションスレッドはオーバーフローした値を `directoryA` のキャッシュファイルに格納します。2 つ目と 4 つ目のトランスフォーメーションスレッドは、オーバーフローした値をディレクトリ B のキャッシュファイルに格納します。

データ統合サービスがアグリゲータ、ジョイナ、ランク、またはソータートランスフォーメーションでキャッシュのパーティション化を使用しない場合は、オーバーフローした値は最初にリストされたディレクトリのキャッシュファイルに格納されます。

注: ルックアップトランスフォーメーションでは、1 つのキャッシュディレクトリしか使用できません。

アグリゲータ、ジョイナ、またはランクトランスフォーメーションの [詳細] プロパティの **【キャッシュディレクトリ】** プロパティで、キャッシュディレクトリを設定します。ソータートランスフォーメーションの [詳細] プロパティの **【作業ディレクトリ】** プロパティで、キャッシュディレクトリを設定します。デフォルトでは、**【キャッシュディレクトリ】** プロパティと **【作業ディレクトリ】** プロパティは、データ統合サービスに定義されているシステムパラメータ値を使用するように設定されます。データ統合サービスの **【キャッシュディレクトリ】** または **【一時ディレクトリ】** プロパティ用に管理者がセミコロンで区切られた複数のディレクトリを入力した場合は、デフォルトの `CacheDir` または `TempDir` システムパラメータ値を使用してください。

異なる値を入力し、トランスフォーメーションに固有のキャッシュディレクトリを複数設定できます。プロパティ、またはプロパティに割り当てられたユーザー定義のパラメータ用に、セミコロンで区切られた複数のディレクトリを入力します。

トランスフォーメーションのためのパーティション化の無効化

パーティション化されたディシジョン、Java、または SQL トランスフォーメーションは、各マッピング実行で同じ結果を返さないことがあります。これらのトランスフォーメーションのパーティション化を無効にすることで、データ統合サービスが 1 つのスレッドを使用してトランスフォーメーションを処理するようにできます。データ統合サービスは複数のスレッドを使用して残りのマッピングパイプラインステージを処理します。

Java または SQL トランスフォーメーションでは、**【パーティション化可能】** 詳細プロパティがデフォルトで選択されます。この詳細プロパティをクリアし、トランスフォーメーションのパーティション化を無効にします。

ディシジョントランスフォーメーションでは、**【パーティション化可能】** 詳細プロパティはデフォルトでクリアされます。この詳細プロパティを選択して、トランスフォーメーションのパーティション化を有効にします。

トランスフォーメーションのためにパーティション化を無効にする理由は、トランスフォーメーションのタイプによって異なります。

ディシジョントランスフォーメーション

数値関数を使用するディシジョントランスフォーメーションのパーティション化を無効にする場合があります。数値関数の `CUME`、`MOVINGSUM`、および `MOVINGAVG` は、行単位で現在の合計および平均を計算します。パーティション化されたディシジョントランスフォーメーションにこれらのいずれかの関数が含まれる場合、各スレッドは関数を個別に処理します。各関数は、すべてのデータではなく、データのサブセットを使用して結果を計算します。したがって、`CUME`、`MOVINGSUM`、または `MOVINGAVG` 関数を使用するパーティション化されたトランスフォーメーションは、マッピングを実行するたびに、同じ計算結果を返さない場合があります。

Java トランスフォーメーション

Java コードで Java トランスフォーメーションを 1 つのスレッドで処理する必要がある場合に、Java トランスフォーメーションのパーティション化を無効にします。

SQL トランスフォーメーション

SQL クエリで SQL トランスフォーメーションを 1 つのスレッドで処理する必要がある場合に、SQL トランスフォーメーションのパーティション化を無効にします。または、SQL トランスフォーメーションのパーティション化を無効にすることで、データベースへの接続を 1 つだけに限定することもできます。

パーティション化されたマッピングにおける順序の維持

ソート済みフラットファイルソース、ソート済みリレーショナルソース、またはソータートランスフォーメーションを使用したマッピングにおける順序を確立できます。マッピングにパーティションポイントを追加する場合、データ統合サービスがデータを再分散し、マッピング内ですでに確立されていた順序が失われることがあります。パーティション化されたマッピングで順序を保持するには、一部のトランスフォーメーションと書き込みトランスフォーメーションで行順序が保持されるように指定する必要があります。

以下のマッピングオブジェクトについては、入力データの行順序を維持するように指定できます。

- 式トランスフォーメーション
- Java トランスフォーメーション
- シーケンスジェネレータトランスフォーメーション
- SQL トランスフォーメーション
- 書き込みトランスフォーメーション[かきこみとらんすふおーめーしょん]

例えば、ソート順で書き込まれているデータに依存するデータベーストリガがリレーショナルターゲットに含まれる場合、行順序を保持するように書き込みトランスフォーメーションを設定します。

行順序を保持するように書き込みトランスフォーメーションを設定すると、データ統合サービスは 1 つのスレッドを使用してターゲットに書き込みます。ソート済みの入力を使用するアグリゲータトランスフォーメーションが書き込みトランスフォーメーションの前にある場合、データ統合サービスは 1 つのスレッドを使用してアグリゲータトランスフォーメーションとターゲットの両方を処理します。

他のすべてのトランスフォーメーションで行順序を保持するように設定するとき、データ統合サービスは順序を保持するためのトランスフォーメーションパイプラインステージの最適なスレッド数を判断します。

行順序を保持するためにトランスフォーメーションの設定に使用する方法は、以下のオブジェクトタイプによって異なります。

式、シーケンスジェネレータ、または SQL トランスフォーメーション

式、シーケンスジェネレータ、または SQL トランスフォーメーションの **【詳細】** プロパティで、**【行順序を保持】** プロパティを選択します。

Java トランスフォーメーション

Java トランスフォーメーションの **【詳細】** プロパティで、**【ステートレス】** プロパティを選択します。

書き込みトランスフォーメーション[かきこみとらんすふおーめーしょん]

書き込みトランスフォーメーションの **【詳細】** プロパティで、**【行順序を保持】** プロパティを選択します。

安定ソートの維持

パーティション化されたマッピングで順序を維持する場合、データ統合サービスは安定ソートを実行しません。データ統合サービスは、ソートキーに基づいて行の順序を維持します。ただし、複数の行でソートキーの値が等しい場合、それらの行の相対順序は入力と出力とで一致しないことがあります。

例えば、パーティションが有効化されたマッピングで、次のデータを含むソート済みのフラットファイルソースを読み取るとします。

```
Order_ID,Item_ID,Item,Quantity,Price
45,000468,ItemD,5,0.56
45,123456,ItemA,5,3.04
41,456789,ItemB,2,12.02
43,123456,ItemA,3,3.04
```

このマッピングには、降順のソートキーとして Order_ID が指定されたソータートランスフォーメーションが含まれています。データ統合サービスでは、複数のスレッドを使用してソータートランスフォーメーションを実行する場合、Order_ID の値が同一の行の相対順序は維持されません。例えば、マージされたターゲットファイルに次の順序で行を書き出すとします。

```
Order_ID,Item_ID,Item,Quantity,Price
45,123456,ItemA,5,3.04
45,000468,ItemD,5,0.56
43,123456,ItemA,3,3.04
41,456789,ItemB,2,12.02
```

安定ソートを維持するには、マッピングの **【最大並行処理】** ランタイムプロパティを 1 に設定して、マッピングのパーティション化を無効にします。

マッピングの最大並行処理のオーバーライド

デフォルトでは、各マッピングの **【最大並行処理】** プロパティは **【自動】** に設定されています。各マッピングは、データ統合サービスに定義されている最大並行処理値を使用します。最大並行処理値をオーバーライドし、特定のマッピングの最大値を定義できます。

データ統合サービスとマッピングで設定されている最大並行処理の整数値が異なる場合、データ統合サービスではこれらの最小値が使用されます。

以下の理由のため、マッピングの **【最大並行処理】** プロパティをオーバーライドした方がよい場合があります。

CPU の処理能力を超えるスレッドとなる複雑なマッピングを実行する。

完全なパイプラインのマッピングに実行できる並行処理スレッドの合計数は、並行処理値にパイプラインステージの数を掛けた値になります。各パーティションポイントには、追加のパイプラインステージが追加されます。複数のアグリゲータまたはジョイナートランスフォーメーションを使用する複雑なマッピングには、多くのパイプラインステージがある場合があります。多数のパイプラインステージにより、データ統合サービスで CPU の処理能力を超える数のスレッドが使用される可能性があります。

各パイプラインステージで並行処理スレッドが少ない方が、マッピングのパフォーマンスがよくなります。

単一のマッピングを少ない並行処理スレッドで実行すると、データ統合サービスで追加のジョブを実行するために使用できるスレッドが増えます。

トランスフォーメーションの推奨並行処理値を定義する必要がある。

マッピングの最大並行処理をオーバーライドする場合、マッピング内の特定のトランスフォーメーションの推奨並行処理値を定義できます。多数のポートが含まれるトランスフォーメーションや複雑な計算を実行するトランスフォーメーションでは、パフォーマンスを最適化するために推奨並行処理値を定義することをお勧めします。

アドレスバリデータまたは一致トランスフォーメーションに対して実行インスタンス数の値を定義する必要があります。

マッピングの最大並行処理をオーバーライドした場合、データ統合サービスは、マッピング内のアドレスバリデータまたは一致トランスフォーメーションに対する実行インスタンス数の値を考慮します。実行インスタンス数の値を定義して、トランスフォーメーションのパフォーマンスを最適化することをお勧めします。

トランスフォーメーションのための提案された並行処理

マッピングの**【最大並行処理】**ランタイムプロパティをオーバーライドする場合、マッピングランタイムプロパティの特定のトランスフォーメーションの**【提案された並行処理】**プロパティを定義できます。

データ統合サービスにより、トランスフォーメーションのパーティション化が可能な限り、トランスフォーメーションパイプラインステージのスレッド数について提案された並行処理値が考慮されます。例えば、行順序を保持するようにマッピングを設定した場合、データ統合サービスはトランスフォーメーション用に1つのスレッドを使用することが必要になる場合があります。

マッピングの**【最大並行処理】**ランタイムプロパティが**【自動】**に設定されていると、マッピングのトランスフォーメーションの提案された並行処理値を定義できません。トランスフォーメーションのための提案された並行処理の値を定義した後に、マッピングの最大並行処理の値を**【自動】**に設定した場合、データ統合サービスは提案された並行処理の値を無視します。

多数のポートが含まれるトランスフォーメーションや複雑な計算を実行するトランスフォーメーションでは、パフォーマンスを最適化するために提案された並行処理値を定義することをお勧めします。

例えば、パーティション化が有効になっているマッピングが小さなデータセットを処理する場合、データ統合サービスは式トランスフォーメーションパイプラインステージの処理にはスレッドが1つあれば十分だと判断するかもしれません。ただし、式トランスフォーメーションに多数の複雑な計算が含まれていると、トランスフォーメーションパイプラインステージの処理に時間がかかる可能性もあります。提案された並行処理値には、1より大きく、マッピングまたはデータ統合サービスに定義された最大並行処理値より小さい値を入力できます。データ統合サービスは式トランスフォーメーションのスレッド数の提案された並行処理値を使用します。

マッピングの最大並行処理をオーバーライドする場合、トランスフォーメーションの**【提案された並行処理】**プロパティの次の値を設定できます。

提案された並行処理値	説明
1	データ統合サービスは、1つのスレッドを使用してトランスフォーメーションを実行します。
自動	データ統合サービスは、マッピングおよびデータ統合サービスに対して定義された最大並行処理を考慮します。データ統合サービスは最低値を使用して、トランスフォーメーションを実行するスレッドの最適な数を決定します。 各トランスフォーメーションのデフォルトです。
1より大きい	データ統合サービスにより、トランスフォーメーションに定義された提案された並行処理、マッピングに定義された最大並行処理、およびデータ統合サービスに定義された最大並行処理が考慮されます。データ統合サービスは最低値を使用して、トランスフォーメーションを実行するスレッドの数を決定します。

次のトランスフォーメーションのマッピングランタイムプロパティの**【提案された並行処理】**プロパティを定義できます。

- アグリゲータ

- 式
- フィルタ
- Java
- ジョイナ
- ルックアップ
- ノーマライザ
- ランク
- ルータ
- シーケンスジェネレータ
- ソータ
- SQL
- 共有体
- アップデートストラテジ

アドレスバリデータおよび一致トランスフォーメーションの実行インスタンス数

マッピングの【最大並行処理】ランタイムプロパティをオーバーライドした場合、データ統合サービスは、アドレスバリデータまたは一致トランスフォーメーションに対して定義された【実行インスタンス数】詳細プロパティの値を考慮します。

データ統合サービスは、トランスフォーメーションがパーティション化可能な限り、実行インスタンス数の値を考慮してトランスフォーメーションパイプラインステージのスレッド数を決定します。例えば、行順序を保持するようにマッピングを設定した場合、データ統合サービスはトランスフォーメーション用に1つのスレッドを使用することが必要になる場合があります。

ID 照合分析用にトランスフォーメーションを設定した場合、一致トランスフォーメーション上の実行インスタンスの数を増やすことができます。フィールド一致分析用にトランスフォーメーションを設定した場合、一致トランスフォーメーション上の実行インスタンスの数を増やすことはできません。フィールド一致分析では、一致トランスフォーメーションは単一の実行インスタンスを使用します。

マッピングの【最大並行処理】ランタイムプロパティが【自動】に設定されている場合、データ統合サービスは、アドレスバリデータまたは一致トランスフォーメーションに対して定義された実行インスタンス数の値を無視します。

マッピングの最大並行処理をオーバーライドする場合、アドレスバリデータまたは一致トランスフォーメーションの【実行インスタンス数】詳細プロパティの以下の値を設定できます。

実行インスタンス数の値	説明
1	データ統合サービスは、1つのスレッドを使用してトランスフォーメーションを実行します。アドレスバリデータトランスフォーメーションのデフォルト。
自動	データ統合サービスは、マッピングおよびデータ統合サービスに対して定義された最大並行処理を考慮します。データ統合サービスは最低値を使用して、トランスフォーメーションを実行するスレッドの最適な数を決定します。 ID 照合分析での一致トランスフォーメーションのデフォルト。
1 より大きい	データ統合サービスは、トランスフォーメーションに対して定義された実行インスタンス数、マッピングに対して定義された最大並行処理、およびデータ統合サービスに対して定義された最大並行処理を考慮します。データ統合サービスは最低値を使用して、トランスフォーメーションを実行するスレッドの数を決定します。

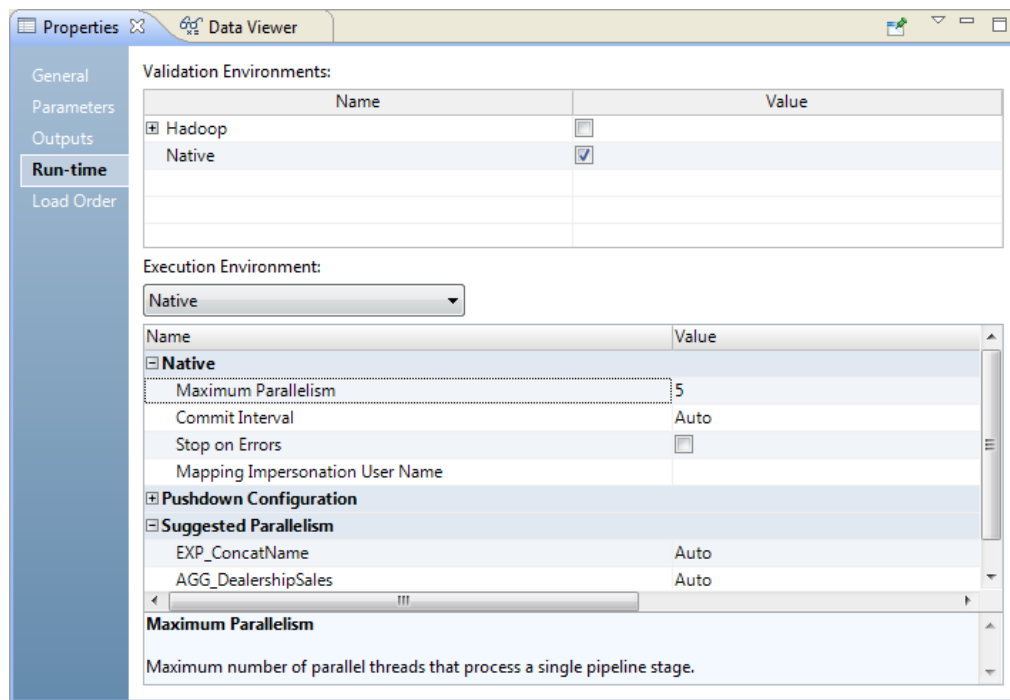
注: また、データ統合サービスは、アドレスバリデータトランスフォーメーションの最適なスレッド数を計算する際に、コンテンツ管理サービスの【最大アドレスオブジェクト数】プロパティも考慮します。【最大アドレスオブジェクト数】プロパティは、マッピング内で同時に実行できるアドレス検証インスタンスの最大数を決定します。【最大アドレスオブジェクト数】の値は、データ統合サービスの【最大並行処理】の値以上である必要があります。

並行処理の最大値のオーバーライド

最大並行処理値をオーバーライドするには、マッピングランタイムプロパティで最大並行処理を、2 以上、かつデータ統合サービス用に設定されている値よりも小さい整数値に設定します。

1. マッピングを開きます。
2. 【プロパティ】ビューで【ランタイム】タブをクリックします。
3. 【実行環境】に【ネイティブ】を選択します。
4. 【最大並行処理】プロパティには、2 以上、かつデータ統合サービス用に設定されている値よりも小さい整数値を入力します。
あるいは、ユーザー定義のパラメータをプロパティに割り当て、続いてパラメータセットまたはパラメータファイル内でパラメータ値を定義することもできます。
5. マッピングの特定のトランスフォーメーションに提案された並行処理値を定義するには、【提案された並行処理】セクションでトランスフォーメーションに対して 1 より大きい整数値を入力します。

次の画像は、トランスフォーメーションにオーバーライドされる [最大並行処理] 値およびデフォルトの [提案された並行処理] 値を持つマッピングを示しています。



6. マッピングを保存します。
7. ID 照合分析に設定されているアドレスバリデータまたは一致トランスフォーメーションに実行インスタンス値を定義するには、次の手順を完了します。
 - a. アドレスバリデータまたは一致トランスフォーメーションを開きます。
 - b. [詳細] ビューで、[実行インスタンス数] プロパティに 1 より大きい整数値を入力します。
 - c. トランスフォーメーションを保存します。

パーティション化されたマッピングのトラブルシューティング

次に示す状況に対する解決方法は、パーティション化されたマッピングのトラブルシューティングで役に立つことがあります。

データ統合サービスの最大並行処理プロパティが 2 以上の値に設定されていて、マッピングの最大並行処理プロパティが [自動] に設定されている。 しかし、マッピングが実行される時にはパーティション化が無効になっている。

複数のパーティションを持つマッピングを実行するためには、データ統合サービスに割り当てられたライセンスにパーティション化が含まれている必要があります。ライセンスにパーティション化が含まれない場合、データ統合サービスは以下のメッセージをマッピングログに書き込み、1 つのスレッドを使用して各マッピングパイプラインステージを処理します。

WARNING: The Integration Service cannot create partitions for the mapping because the license assigned to the Integration Service does not include partitioning.
 INFO: Partitioning is disabled for the mapping.

索引

A

application 要素
パラメータファイル [66](#)

C

CUME 関数
パーティション化の制限事項 [264](#)

I

IBM DB2 for LUW ソース
パーティション化 [254](#)
IBM DB2 for LUW ターゲット
パーティション化 [260](#)
IBM DB2 ソース
プッシュダウンの最適化 [231](#)
infacmd
パラメータセットの使用 [64](#)

J

Java トランスフォーメーション
パーティション化 [262](#)
パーティション化の無効化 [264](#)

L

listMappingPersistedOutputs
説明 [88](#)

M

Microsoft SQL Server ソース
プッシュダウンの最適化 [231](#), [232](#)
プッシュダウンの最適化 [231](#), [232](#)
MOVINGAVG 関数
パーティション化の制限事項 [264](#)
MOVINGSUM 関数
パーティション化の制限事項 [264](#)

O

Oracle ソース
パーティション化 [254](#)
プッシュダウンの最適化 [231](#)

P

PowerCenter からのインポート
インポートの制限 [215](#)
インポートパフォーマンス [216](#)
オブジェクトのインポート [214](#)
オプション [212](#)
概要 [201](#)
競合の解決 [202](#)
システム定義パラメータ [211](#)
トランスフォーメーションのタイプの変換 [203](#)
パラメータの変換 [211](#)
PowerCenter へのエクスポート
オブジェクトのエクスポート [197](#)
オプション [195](#)
概要 [193](#)
互換性レベルの設定 [194](#)
制限 [198](#)
トラブルシューティング [200](#)
パラメータの変換 [195](#)
リリースの互換性 [194](#)
ルールおよびガイドライン [199](#)
project 要素
パラメータファイル [65](#)

S

SAP HANA ソース
プッシュダウンの最適化 [232](#)
SAP ソース
プッシュダウンの最適化 [232](#)
SQL トランスフォーメーション
パーティション化 [262](#)
パーティション化の無効化 [264](#)
SQL 文
パラメータ [54](#)
Sybase ASE ソース
プッシュダウンの最適化 [232](#)

U

updateMappingPersistedOutputs
説明 [88](#)

あ

アグリゲータトランスフォーメーション
キャッシュのパーティション化 [263](#)
パーティション化 [262](#)
複数のキャッシュディレクトリ [263](#)
アドレスバリデータトランスフォーメーション
実行インスタンス [268](#)

安定ソート順
パーティション化されたマッピングにおける維持 [266](#)

い

依存関係
暗黙 [27](#)
リンクパス [26](#)
一致トランスフォーメーション
実行インスタンス [268](#)
インスタンス値
パラメータの設定 [58](#), [78](#)

え

エクスポート
PowerCenter [193](#)
エラー時の停止
マッピングプロパティ [29](#)
演算子
ソースで利用可能 [246](#)
プッシュダウンの最適化 [246](#)

か

書き込みトランスフォーメーション
動的 [116](#)
仮想テーブルマッピング
パラメータの設定 [62](#)
関数
ソースで利用可能 [233](#)
プッシュダウンの最適化 [233](#)
完全最適化レベル
説明 [222](#)

き

キャッシュサイズ
パーティション [263](#)
キャッシュディレクトリ
最適化 [263](#)
複数のディレクトリ [263](#)
キャッシュのパーティション化
キャッシュサイズ [263](#)
説明 [263](#)
行インジケータ
拒否ファイル [191](#)
拒否ファイル
カラムインジケータ [192](#)
行インジケータ [191](#)
ターゲット [190](#)
場所 [190](#)
パーティション化 [190](#)
表示 [190](#)
読み取り [191](#)

こ

コストベース最適化
説明 [220](#)

さ

最小の最適化レベル
説明 [222](#)
最大並行処理
上書き [267](#)
説明 [251](#)
マッピングランタイム [29](#)
最適化
コストベースの最適化方式 [220](#)
準結合最適化方式 [221](#)
初期選択の最適化方法 [219](#)
初期プロジェクション最適化方法 [218](#)
データシップ結合最適化方式 [220](#)
ブランチ刈り込み最適化方式 [219](#)
マッピングのパフォーマンスの方式 [218](#)
最適化レベル
説明 [222](#)

し

式
パラメータの使用 [52](#)
プッシュダウンの最適化 [233](#)
式トランスフォーメーション
動的式 [122](#)
パーティション化 [262](#)
[マッピング出力] ビュー [85](#)
式パラメータ
説明 [54](#)
システム定義パラメータ
インポート [211](#)
実行インスタンス
アドレスバリデータトランスフォーメーション [268](#)
一致トランスフォーメーション [268](#)
実行時のターゲットの作成
ルールおよびガイドライン [118](#)
指定元
値またはパラメータ [54](#)
集計関数
説明 [87](#)
集計タイプ
マッピング出力 [83](#)
出力式
設定方法 [97](#)
マップレットでの設定 [104](#)
出力バインディング
ダイアログボックスの説明 [91](#)
[出力] ビュー
説明 [83](#)
出力ファイルディレクトリ
最適化 [257](#)
複数のディレクトリ [257](#)
準結合最適化
説明 [221](#)
順序
パーティション化されたマッピングにおける維持 [265](#)
パーティション化されたマッピングの安定ソート [266](#)
ジョイナトランスフォーメーション
キャッシュのパーティション化 [263](#)
パーティション化 [262](#)
複数のキャッシュディレクトリ [263](#)
初期選択の最適化
説明 [219](#)
初期プロジェクション最適化
説明 [218](#)

処理スレッド
マッピング [250](#)

す

スコープ
ポートセレクト [132](#)
スレッド
処理用マッピング [250](#)

せ

生成されたポート
概要 [120](#)
名前の変更 [146](#)
並べ替え [146](#)
生成されたポートの順序変更
動的マッピング [129](#)
例 [129](#)
生成されたポートの名前の変更
例 [127](#)
生成されたマップレット
概要 [42](#)
検証エラー [42](#)
作成 [42](#)
ルールおよびガイドライン [42](#)
制約
ターゲットのロード順 [32](#)
ターゲットロード順序の作成 [34](#)
セグメント
コピー [38](#)
選択条件
ポートセレクト [132](#)
選択ルール
動的マッピング [133](#)
ポートセレクト [132](#)
例 [134](#)

そ

ソース
パーティション化されたフラットファイル [253](#)
パーティション化されたリレーショナル [254](#)
ソータートランスフォーメーション
キャッシュのパーティション化 [263](#)
ソートリストパラメータ
説明 [57](#)
ソート順
パーティション化されたマッピングにおける維持 [265](#)

た

ターゲット
パーティション化されたフラットファイル [256](#)
パーティション化されたリレーショナル [260](#)
マージファイル [257](#)
ターゲットコミット間隔
マッピングプロパティ [29](#)
ターゲットロード順序の制約
作成 [34](#)
ターゲットロード順の制約
説明 [32](#)
ルールおよびガイドライン [34](#)
例 [35](#)

ターゲットロード順の制約 (続く)
[ロード順序] タブ [32](#)

て

提案された並行処理
トランスフォーメーション [267](#)
ディジヨントランスフォーメーション
パーティション化 [262](#)
パーティション化の無効化 [264](#)
データオブジェクト
実行時のカラムの取得 [112](#)
データシップ結合最適化
説明 [220](#)

と

同時読み込みのパーティション化
説明 [253](#)
動的式
概要 [122](#)
作成 [151](#)
例 [122](#)
動的ソース
実行時のカラムの取得 [112](#)
説明 [111](#)
動的ターゲット
実行時に作成または置換 [117](#)
実行時のカラムの取得 [116](#)
データオブジェクトに基づく定義 [117](#)
マッピングフローに基づく定義 [117](#)
動的ポート
概要 [120](#)
作成 [145](#)
設定 [146](#)
動的ポートおよび生成されたポート
設定 [121](#)
動的マッピング
トランスフォーメーション [121](#)
動的マッピング
開発と実行 [142](#)
概要 [107](#)
書き込みトランスフォーメーションの設定 [154](#)
検証 [160](#)
実行 [160](#)
生成されたポート [120](#)
生成されたポートの順序変更 [129](#)
生成されたポートの名前の変更 [126](#)
設計時リンク [134](#)
設定の概要 [108](#)
選択ルール [133](#)
ソース [111](#)
ソースデータオブジェクトのパラメータ化 [114](#)
ソース名のパラメータ化 [113](#)
ターゲットオブジェクト [115](#)
ターゲットデータオブジェクトのパラメータ化 [119](#)
ターゲット名のパラメータ化 [119](#)
動的ポート [120](#)
動的ポートの作成 [145](#)
トラブルシューティング [139](#)
入力ルール [123-125](#)
入力ルール [123-125](#)
入力ルールの定義 [146](#)
残りのすべてのポートを含める [125](#)
パラメータ [110](#)
ポートおよびリンク [109](#)

動的マッピング (続く)

- ポートセレクタ [133](#)
- ポートを含めるまたは除外する [125](#)
- ランタイムリンク [136](#), [137](#)
- ランタイムリンクの作成 [158](#)
- リンクの解決 [136](#)

動的マッピングのコンポーネント

- データソース [108](#)
- ルール [110](#)

動的マッピングのルール

- 概要 [110](#)
- トラブルシューティング
- PowerCenter へのオブジェクトのエクスポート [200](#)
- 動的マッピング [139](#)
- トランスフォーメーション
- 提案された並行処理 [267](#)
- パーティション化 [262](#)

に

入力リンクセットパラメータ

- 説明 [57](#)

入力ルール

- 概要 [123](#)
- 生成されたポートの名前の変更 [126](#)
- 設定 [124](#)
- 定義 [146](#)
- 残りのすべてのポートを含める [125](#)
- ポートを含めるまたは除外する [125](#)

の

ノーマル最適化レベル

- 説明 [222](#)

は

パーティション化の制限事項

- Java トランスフォーメーション [262](#)
- SQL トランスフォーメーション [262](#)
- アグリゲータトランスフォーメーション [262](#)
- 式トランスフォーメーション [262](#)
- ジョイナトランスフォーメーション [262](#)
- 数値関数 [264](#)
- ディシジョントランスフォーメーション [262](#)
- ランクトランスフォーメーション [262](#)
- リレーショナルソース [255](#), [256](#)
- リレーショナルターゲット [261](#)

パーティションポイント

- 説明 [250](#)

パイプラインステージ

- 説明 [250](#)

パイプラインのマッピング

- 説明 [250](#)

バインディング

- 出力のワークフロー変数へのマッピング [101](#)
- マッピングへのマップレット出力 [91](#), [102](#)

パーティション化

- IBM DB2 for LUW ソース [254](#)
- IBM DB2 for LUW ターゲット [260](#)
- Java トランスフォーメーション [262](#)
- Oracle ソース [254](#)
- SQL トランスフォーメーション [262](#)
- アグリゲータトランスフォーメーション [262](#)
- アドレスバリデータトランスフォーメーション [268](#)

パーティション化 (続く)

- 安定ソート順の維持 [266](#)
- 一致トランスフォーメーション [268](#)
- キャッシュ [263](#)
- キャッシュサイズ [263](#)
- 行順序を保持 [265](#)
- 拒否ファイル [190](#)
- 最大並行処理 [251](#)
- 式トランスフォーメーション [262](#)
- ジョイナトランスフォーメーション [262](#)
- ディシジョントランスフォーメーション [262](#)
- 同時読み取り [253](#)
- トラブルシューティング [270](#)
- トランスフォーメーション [262](#)
- フラットファイルソース [253](#)
- フラットファイルターゲット [256](#)
- マージされたファイルターゲット [257](#)
- マッピング [251](#)
- マッピング用に減らす [266](#)
- ランクトランスフォーメーション [262](#)
- リレーショナル接続タイプ [255](#), [261](#)
- パフォーマンスのチューニング
- コストベースの最適化方式 [220](#)
- 最適化方式 [218](#)
- 最適化レベル [222](#)
- 述部最適化方式 [219](#)
- 準結合最適化方式 [221](#)
- 初期選択の最適化方式 [219](#)
- 初期プロジェクション最適化方法 [218](#)
- データシップ結合最適化方式 [220](#)
- プッシュダウンの最適化 [225](#)
- ブランチ刈り込み最適化方式 [219](#)

パラメータ

- PowerCenter からのインポート [211](#)
- PowerCenter へのエクスポート [195](#)
- SQL 文 [54](#)
- インスタンス値 [78](#)
- 仮想テーブルマッピング [62](#)
- 式 [52](#)
- 設定方法 [71](#)
- ソースデータオブジェクト [114](#)
- ターゲットデータオブジェクト [119](#)
- テーブル名とリソース [55](#)
- 動的マッピング [110](#)
- フラットファイルソース [113](#)
- フラットファイルの区切り文字 [56](#)
- マップレット [59](#)
- リレーショナルテーブルのプロパティ [113](#)
- 論理データオブジェクト [61](#)

パラメータ

- 日付フォーマット [48](#)
- パラメータインスタンス値
- 設定 [58](#)
- パラメータのバインド
- インスタンス値 [78](#)
- パラメータセット
- 概要 [63](#)
- 作成 [79](#)

パラメータファイル

- application 要素 [66](#)
- project 要素 [65](#)
- XML スキーマ定義 [64](#)
- 構造 [64](#)
- 作成 [70](#)
- サンプル [67](#)
- マッピングの実行 [64](#)
- 目的 [64](#)
- ワークフローの実行時に使用 [64](#)

ひ

日付パラメータ
有効な形式 [48](#)
非リレーショナルソース
ブッシュダウンの最適化 [232](#)

ふ

フィードバックのバインディング
説明 [87](#)
ブッシュダウンの最適化
Greenplum ソース [232](#)
IBM DB2 ソース [231](#)
Microsoft SQL Server ソース [231](#), [232](#)
ODBC ソース [232](#)
Oracle ソース [231](#)
SAP HANA ソース [232](#)
SAP ソース [232](#)
Sybase ASE ソース [232](#)
z/OS の非リレーショナルソース [232](#)
演算子 [246](#)
概要 [225](#)
関数 [233](#)
式 [233](#)
ソース [229](#)
ブッシュダウンタイプ [226](#)
リレーショナルソース [229](#), [231](#)
リレーショナルソース [229](#), [231](#)
ブッシュダウンの最適化方式
完全なブッシュダウン [226](#)
ソースブッシュダウン [227](#)
ブッシュダウンの設定 [227](#)
フラットファイルソース
パーティション化 [253](#)
フラットファイルターゲット
拒否ファイル [190](#)
パーティション化 [256](#)
パーティションの統合 [257](#)
複数の出力ディレクトリ [257](#)
フラットファイルの区切り文字
パラメータの使用 [56](#)
フラットファイルデータオブジェクト
拒否ファイル [190](#)
ブランチ刈り込み最適化
説明 [219](#)

へ

並行処理
マッピング用に減らす [266](#)

ほ

方法
パラメータの設定 [71](#)
保持された出力を入力タスクにバインドする [100](#)
マッピング出力の設定 [93](#)
マッピング出力の保持 [99](#)
マッピング出力のワークフロー変数へのバインド [101](#)
マッピングでの出力式の定義 [97](#)
マップレット出力の定義 [103](#)
マップレット出力をマッピングにバインドする [102](#)
マップレットでの出力式の定義 [104](#)

ポート
位置によるリンク [25](#)
自動的にリンク [24](#)
手動でリンク [24](#)
接続の検証 [36](#)
トランスフォーメーションでプロパゲートされる属性 [27](#)
名前によるリンク [24](#)
リンク [22](#)
リンクのルールおよびガイドライン [25](#)

ポートセレクト
作成 [150](#)
説明 [132](#)
選択ルール [132](#), [133](#)
選択ルール [132](#), [133](#)
動的式で [122](#)
例 [134](#)

ポート属性
プロパゲート [26](#)
ポートのプレビュー
ポートセレクト [132](#)
ポートリストパラメータ
説明 [57](#)

ま

マッピング
SQL クエリから [18](#)
エラー時の停止 [29](#)
オブジェクトの依存性 [17](#)
オブジェクト [16](#)
オブジェクトの検証 [37](#)
オブジェクトの追加 [17](#)
開発 [18](#)
概要 [15](#)
環境 [29](#)
拒否ファイル [190](#)
検査 [37](#)
検証 [36](#)
検証環境 [29](#)
最大並行処理 [29](#), [251](#)
最適化されたマッピング [222](#)
最適化方式 [218](#)
作成 [18](#)
式の検証 [37](#)
実行環境 [29](#)
実行中 [37](#)
述部最適化方式 [219](#)
処理スレッド [250](#)
制約の作成 [34](#)
接続の検証 [36](#)
ターゲットコミット間隔 [29](#)
ターゲットロード順の制約 [32](#)
パーティション化 [251](#)
パーティションポイント [250](#)
パイプライン [250](#)
並行処理の減少 [266](#)
ランタイムプロパティ [29](#)
マッピング、動的
トラブルシューティング [139](#)
マッピング管理
概要 [187](#)
マッピング出力
infacmd の使用 [88](#)
概要 [82](#)
システム定義 [82](#)
集計タイプ [83](#)
出力式 [85](#)

マッピング出力 (続く)

出力の定義方法 [96](#)

[出力] ビュー [83](#)

設定手順 [93](#)

定義 [83](#)

デプロイメントの変更 [88](#)

保持 [87](#)

保持のガイドライン [88](#)

マプレットへのバインディング [90](#)

ユーザー定義 [83](#)

論理データオブジェクト [93](#)

マッピング出力の保持

infacmd の使用 [88](#)

説明 [87](#)

タスク入力へのバインド [100](#)

方法 [99](#)

ルールおよびガイドライン [88](#)

マッピングタスクの入力

保持された出力をバインド [100](#)

マッピングパラメータ

infacmd [64](#)

概要 [45](#)

仮想テーブルマッピング [62](#)

作成する場所 [48](#)

システム [46](#)

タイプ [47](#)

ユーザー定義 [47](#)

割り当て先 [49](#)

マッピングパラメータとして公開

説明 [58](#), [59](#)

タスクの説明 [77](#)

マプレット

PowerCenter へのエクスポート [194](#)

作成 [44](#)

出力 [42](#)

入力 [41](#)

パラメータの使用 [59](#)

マプレット

概要 [39](#)

検証 [44](#)

生成 [42](#)

タイプ [41](#)

ルール [44](#)

マプレット出力

定義 [103](#)

マッピング出力へのバインディング [102](#)

マッピングへのバインディング [90](#)

マッピングへのバインド [91](#)

マッピングへのバインド方法 [105](#)

マプレットパラメータ

例 [60](#)

も

文字列パラメータ

精度の制限 [47](#)

ら

リンクトランスフォーメーション

キャッシュのパーティション化 [263](#)

リンクトランスフォーメーション (続く)

パーティション化 [262](#)

複数のキャッシュディレクトリ [263](#)

ランタイムリンク

概要 [136](#)

作成 [158](#)

動的マッピング [137](#)

リンクポリシー [137](#)

例 [139](#)

り

リレーショナルソース

パーティション化 [254](#)

パーティション化の制限事項 [255](#), [256](#)

プッシュダウンの最適化 [231](#)

リレーショナルデータオブジェクト

拒否ファイル [190](#)

パーティション化 [254](#), [260](#)

リレーショナルターゲット

拒否ファイル [190](#)

パーティション化 [260](#)

パーティション化の制限事項 [261](#)

リンクの解決

動的マッピング [136](#)

る

ルールおよびガイドライン

実行時のターゲットの作成 [118](#)

ルックアップトランスフォーメーション

キャッシュのパーティション化 [263](#)

れ

例

生成されたポートの順序変更 [129](#)

生成されたポートの名前の変更 [127](#)

動的式 [122](#)

動的マッピングの選択ルール [134](#)

ランタイムリンク [139](#)

ろ

ロード順

制約 [32](#)

[ロード順序] タブ

説明 [32](#)

論理データオブジェクト

パラメータの使用 [61](#)

マッピング出力 [93](#)

わ

ワークフロー変数

へのマッピング出力のバインディング [101](#)