



Informatica®
10.5.2

Developer tool ガイド

Informatica Developer tool ガイド

10.5.2

2022 年 4 月

© 著作権 Informatica LLC 2009, 2022

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

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

オプトアウトの権利の制限の下、本ソフトウェアによって、本ソフトウェアがデプロイされているコンピューティングおよびネットワーク環境に関する情報、デプロイメントのデータ使用状況およびシステム統計情報が米国の Informatica に自動的に送信されます。この送信は Informatica のプライバシーポリシーの下にサービスの一部と見なされ、Informatica は、<https://www.informatica.com/in/privacy-policy.html> のプライバシーポリシーに従って、この情報を使用または処理します。使用状況の収集は、Administrator ツールで無効にすることができます。

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

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

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

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

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

発行日: 2022-06-07

目次

序文	13
Informatica のリソース.....	13
Informatica Network.....	13
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.....	14
Informatica Velocity.....	14
Informatica Marketplace.....	14
Informatica グローバルカスタマサポート.....	14
 第 1 章 : Informatica Developer	15
Informatica Developer の概要.....	15
Informatica Data Quality およびプロファイリング.....	15
Informatica Data Services.....	17
Informatica Developer の起動.....	18
ローカルマシンでの Developer ツールの起動.....	18
リモートマシンでの Developer ツールの起動.....	18
Informatica Developer のインターフェース.....	19
Informatica Developer のビュー.....	20
Informatica Developer のようこそページ.....	21
チートシート.....	21
Informatica Developer のオンラインヘルプ.....	22
Informatica の設定.....	22
Informatica Marketplace.....	22
Informatica Developer の設定.....	23
手順 1. ドメインの追加.....	23
手順 2. モデルリポジトリの追加.....	23
手順 3. デフォルトのデータ統合サービスの選択.....	24
手順 4. デフォルトのメタデータアクセスサービスの選択.....	24
ドメイン.....	25
プロジェクト.....	25
プロジェクトの作成.....	25
プロジェクトのフィルタ.....	26
プロジェクトの権限.....	26
外部オブジェクトの権限.....	27
依存オブジェクトインスタンスに対する権限.....	27
親オブジェクトのアクセス.....	28
権限の割り当て.....	28
フォルダ.....	29
フォルダーの作成.....	29

オブジェクトのコピー操作.	30
オブジェクトのコピー.	30
オブジェクトのコピーの保存.	31
タグ.	31
タグの作成.	31
タグの割り当て.	32
タグの表示.	32
第2章: モデルリポジトリ.	33
モデルリポジトリの概要.	33
Informatica Developer のオブジェクト.	34
オブジェクトプロパティ.	37
リポジトリオブジェクトのロック.	37
ロック管理.	37
ロック管理のルールとガイドライン.	38
バージョン管理されたオブジェクトを使用したチームベース開発.	38
バージョンされたオブジェクトの管理.	39
オブジェクトの履歴バージョン.	42
[チェックアウトされたオブジェクト] ビュー.	42
[バージョン履歴] ビュー.	43
チームベース開発のトラブルシューティング.	43
モデルリポジトリへの接続.	43
モデルリポジトリサービスの更新.	44
第3章: Informatica Developer での検索.	45
Informatica Developer での検索の概要.	45
ドメイン検索.	45
オブジェクトとプロパティの検索.	46
ビジネス用語集の検索.	47
ビジネス用語集デスクトップのルックアップ.	47
ビジネス用語のルックアップ.	47
ビジネス用語のルックアップ用にホットキーをカスタマイズ.	48
エディタでの検索.	48
第4章: 接続.	50
接続の概要.	50
接続タイプ.	51
[接続エクスプローラ] ビュー.	52
接続管理.	53
接続の作成.	53
接続の編集.	54
接続のコピー.	54
接続の削除.	55

接続リストの更新.	55
接続の切り替え.	55
接続を切り替える前に.	56
接続の切り替え.	56
接続を切り替えた後に.	57
サードパーティ製 JDBC ドライバ.	59
環境 SQL.	59
接続環境 SQL.	59
トランザクション環境 SQL.	60
環境 SQL の設定に関するガイドライン.	60
第 5 章 : 物理データオブジェクト.	61
物理データオブジェクトの概要.	61
物理データオブジェクトのタイプ.	62
リレーショナルデータオブジェクト.	62
リレーショナルデータオブジェクトのインポート.	64
キーのリレーション.	64
カスタマイズデータオブジェクト.	66
キーのリレーション.	67
カスタマイズデータオブジェクトの書き込みプロパティ.	68
カスタマイズデータオブジェクトの作成.	69
カスタマイズデータオブジェクトへのリレーショナルリソースの追加.	70
カスタマイズデータオブジェクトへのリレーショナルデータオブジェクトの追加.	70
カスタマイズデータオブジェクトのキーの作成.	71
カスタマイズデータオブジェクト内のリレーションの作成.	71
ターゲットテーブルの作成または置換.	72
ターゲットテーブル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.	72
設計時の DDL の生成と実行.	73
実行時での DDL の作成と実行.	74
DDL の生成エラー.	74
カスタムクエリ.	75
カスタムクエリの最適化.	75
デフォルトクエリ.	75
ヒント.	76
個別に選択.	78
フィルタ.	78
ソート済みポート.	79
ユーザー定義ジョイン条件.	80
アウトージョインのサポート.	81
Informatica ジョイン構文.	82
PreSQL コマンドおよび PostSQL コマンド.	86
カスタムクエリの作成.	87
非リレーショナルデータオブジェクト.	87

非リレーショナルデータオブジェクトのインポート.	88
非リレーショナルデータ操作からの読み取り、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作成.	89
WSDL データオブジェクト.	89
WSDL データオブジェクトの概要ビュー.	90
WSDL データオブジェクトの詳細ビュー.	90
WSDL データオブジェクトのインポート.	90
WSDL の同期.	91
証明書管理.	92
同期化.	93
Informatica Developer でのフラットファイルデータオブジェクトの同期.	93
Informatica Developer でのリレーショナルデータオブジェクトの同期.	95
物理データオブジェクトのトラブルシューティング.	96
第 6 章: フラットファイルデータオブジェクト.	97
フラットファイルデータオブジェクトの概要.	97
ソースファイル名の生成.	98
フラットファイルデータオブジェクトの概要プロパティ.	98
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.	99
フォーマットのプロパティ.	100
カラム形式: 区切りプロパティ.	101
カラム形式: 固定幅プロパティ.	102
ランタイム読み取りプロパティ.	103
ランタイム書き込みプロパティ.	105
制御ファイル.	107
実行時のカラムの更新.	108
ランタイムカラム名の自動生成.	108
データファイルヘッダからのランタイムカラム名の生成.	109
制御ファイルからのカラムメタデータの生成.	109
制御ファイルの形式.	110
ランタイムプロパティのパラメータ化.	110
制御ファイルのランタイム処理.	110
制御ファイルに関するルールおよびガイドライン.	111
Excel からフラットファイルデータオブジェクトへのコピー.	111
Excel でのフラットファイルデータオブジェクトの編集.	112
フラットファイルデータオブジェクトへのメタデータのコピー.	112
Excel でのデータオブジェクトの編集例.	112
フラットファイルデータオブジェクトの作成.	113
空のフラットファイルデータオブジェクトの作成.	113
既存のフラットファイルからのフラットファイルデータオブジェクトの作成.	114
制御ファイルからのフラットファイルデータオブジェクトの作成.	116

第 7 章 : データの論理ビュー	117
データの論理ビューの概要	117
論理データオブジェクトモデルの例	118
データの論理ビューの開発	119
論理データオブジェクトモデル	119
論理データオブジェクトモデルの作成	119
モデリングツールからの論理データオブジェクトモデルのインポート	120
論理データオブジェクトモデルのプロパティ	121
CA ERwin Data Modeler のインポートプロパティ	121
IBM Cognos Business Intelligence Reporting - Framework Manager のインポートプロパティ	122
SAP BusinessObjects Designer のインポートプロパティ	123
SAP PowerDesigner CDM のインポートプロパティ	124
SAP PowerDesigner PDM のインポートプロパティ	125
XSD のインポートプロパティ	126
論理データオブジェクト	127
論理データオブジェクトのプロパティ	127
属性のリレーション	128
論理データオブジェクトの作成	128
論理データオブジェクトマッピング	131
論理データオブジェクト読み取りマッピング	132
論理データオブジェクト書き込みマッピング	132
論理データオブジェクトマッピングの作成	132
第 8 章 : データの表示	134
データの表示の概要	134
構成	135
構成プロパティ	135
データビューアの構成	139
マッピング構成	141
Web サービスの設定	142
デフォルトの構成プロパティの更新	142
構成のトラブルシューティング	143
データのエクスポート	143
オブジェクトの依存関係	143
オブジェクトの依存関係の表示	144
オブジェクトの依存関係の表示	144
オブジェクトの依存関係のフィルタリング	145
ログ	146
ログファイル形式	146
検証設定	146
エラーメッセージのグループ化	146

エラーメッセージの制限.	147
Developer tool からのジョブの監視.	147
第 9 章 : アプリケーションデプロイメント.	148
アプリケーションデプロイメントの概要.	148
n アプリケーションの作成.	149
完全なアプリケーション.	149
差分アプリケーション.	150
アプリケーションのプロパティ.	150
アプリケーションへのデプロイメント.	152
アーカイブファイルへのデプロイメント.	154
リソースパラメータを使用したデプロイメント.	154
ランタイムオブジェクトの表示.	156
アプリケーションの再デプロイメント.	156
アプリケーションの状態情報.	157
アプリケーションの作成、デプロイ、更新の方法.	158
アプリケーションの作成.	159
アプリケーションのデプロイ.	161
オブジェクトのデプロイ.	162
アプリケーションアーカイブのインポート.	163
ランタイムアプリケーションへの接続.	164
アプリケーションの編集.	164
データ統合サービスへのアプリケーションの再デプロイ.	165
第 10 章 : アプリケーションパッチのデプロイメント.	167
アプリケーションパッチのデプロイメントの概要.	167
アプリケーションオブジェクトインスタンス.	168
アプリケーションオブジェクトの依存関係.	168
差分デプロイメントウィザード.	172
アプリケーションの選択.	172
ソースオブジェクトの選択.	173
パッチのデプロイメント戦略.	175
アプリケーションパッチのタイプの選択.	175
直接依存関係のみの継承.	177
直接および間接の依存関係の継承.	178
直接、間接、およびリモートの依存関係の継承.	179
アプリケーションオブジェクトの影響の決定.	181
オブジェクトの影響のプレビュー.	182
影響を受けるオブジェクトの特定.	182
オブジェクトの影響の範囲の特定.	185
パッチデプロイメント戦略の指定.	185
パッチのデプロイメント戦略のプロパティ.	186
アプリケーションパッチのデプロイ.	187

アプリケーションパッチのデプロイメントに関するルールとガイドライン.....	189
第 11 章 : アプリケーションパッチのデプロイメント例.....	191
アプリケーションパッチのデプロイメント例の概要.....	191
パッチタイプ: 直接.....	192
パッチタイプ: 直接および間接.....	194
パッチタイプ: 直接、間接、リモート.....	196
まとめ.....	198
第 12 章 : 継続的統合と継続的配信 (CI/CD)	199
CI/CD の概要.....	199
REST API および infacmd クイックリファレンス.....	200
REST API または infacmd の使用.....	202
REST API を使用したアプリケーションのデプロイと再デプロイ.....	202
継続的統合.....	203
オブジェクトのデプロイ.....	203
オブジェクトの割り当て.....	204
オブジェクト競合の解決.....	204
継続的配信.....	205
自動化された統合プロセスのサンプル.....	205
手順 1. デプロイメント用にマッピングを準備.....	206
手順 2. データ統合サービスへのマッピングのデプロイ.....	206
手順 3. 正常性テストの実行.....	206
手順 4. 編集用にマッピングを準備.....	207
自動化された配信プロセスのサンプル.....	208
第 13 章 : オブジェクトのインポートとエクスポート.....	209
オブジェクトのインポートとエクスポートの概要.....	209
オブジェクトのインポートとエクスポート.....	210
オブジェクトエクスポート.....	211
オブジェクトのエクスポート.....	212
オブジェクトインポート.....	212
プロジェクトのインポート.....	213
オブジェクトのインポート.....	214
以前の Informatica リリースからのオブジェクトのインポート.....	216
付録 A : データ型リファレンス.....	217
データ型リファレンスの概要.....	217
トランスフォーメーションデータ型.....	218
Integer データ型.....	219
Binary データ型.....	221
Date/Time データ型.....	221
Decimal および Double データ型.....	223

String データ型.	225
複合データ型.	225
複合ファイルデータ型とトランスフォーメーションデータ型.	229
Avro のデータ型とトランスフォーメーションデータ型.	230
JSON のデータ型とトランスフォーメーションデータ型.	232
ORC のデータ型とトランスフォーメーションデータ型.	232
Parquet のデータ型とトランスフォーメーションデータ型.	233
データ型のルールとガイドライン.	235
フラットファイルとトランスフォーメーションデータ型.	237
DB2 for LUW とトランスフォーメーションデータ型.	238
DB2 for i5/OS および DB2 for z/OS のデータ型とトランスフォーメーションデータ型.	239
DB2 for i5/OS および DB2 for z/OS のサポートされていないデータ型.	240
JDBC のデータ型とトランスフォーメーションのデータ型.	241
Microsoft SQL Server のデータ型とトランスフォーメーションデータ型.	242
Uniqueidentifier データタイプ.	244
非リレーショナルのデータ型とトランスフォーメーションのデータ型.	245
ODBC データ型とトランスフォーメーションデータ型.	247
Oracle のデータ型およびトランスフォーメーションデータ型.	248
Number (P,S) データ型.	250
Char、Varchar、Clob データタイプ.	250
サポートされていない Oracle のデータ型.	251
SAP HANA およびトランスフォーメーションデータ型.	251
XML データ型とトランスフォーメーションデータ型.	253
データの変換.	255
ポート対ポートのデータ変換.	255
付録 B: キーボードショートカット.	257
詳細マッピングオプションのキーボードショートカット.	257
オブジェクトのキーボードショートカット.	258
ポートのキーボードショートカット.	259
[トランスフォーメーション] パレットのキーボードショートカット.	260
ワークベンチのキーボードショートカット.	260
付録 C: 接続プロパティ.	262
接続プロパティの概要.	263
Adabas 接続のプロパティ.	264
Amazon Redshift 接続のプロパティ.	266
Amazon S3 接続のプロパティ.	269
ブロックチェーン接続プロパティ.	272
Cassandra 接続のプロパティ.	273
Confluent Kafka 接続.	275
全般プロパティ.	276
Confluent Kafka Broker のプロパティ.	276

SSL プロパティ	277
infacmd を使用した Confluent Kafka 接続の作成	277
Databricks 接続プロパティ	277
Greenplum 接続プロパティ	279
Google Analytics 接続のプロパティ	281
Google BigQuery 接続のプロパティ	281
接続モード	283
Google Cloud Spanner 接続のプロパティ	283
Google Cloud Storage 接続のプロパティ	284
Google PubSub 接続のプロパティ	285
Hadoop 接続プロパティ	286
Hadoop クラスタプロパティ	286
共通プロパティ	288
拒否ディレクトリのプロパティ	290
Blaze 設定	290
Spark 設定	291
HBase 接続プロパティ	292
HDFS 接続プロパティ	293
MapR-DB の HBase 接続のプロパティ	295
Hive 接続のプロパティ	295
HTTP 接続のプロパティ	299
IBM DB2 接続のプロパティ	301
IBM DB2 for i5/OS 接続のプロパティ	304
IBM DB2 for z/OS 接続のプロパティ	308
IMS 接続のプロパティ	310
JDBC 接続のプロパティ	313
JDBC V2 接続のプロパティ	315
JD Edwards EnterpriseOne 接続プロパティ	317
Kafka 接続のプロパティ	318
全般プロパティ	319
Kafka Broker のプロパティ	319
SSL プロパティ	320
infacmd を使用した Kafka 接続の作成	321
Kudu 接続プロパティ	321
LDAP 接続プロパティ	322
Microsoft Azure Blob ストレージ接続のプロパティ	323
Microsoft Azure Cosmos DB SQL API 接続のプロパティ	324
Microsoft Azure Data Lake Storage Gen1 接続のプロパティ	325
Microsoft Azure Data Lake Storage Gen2 接続のプロパティ	326
Microsoft Azure SQL Data Warehouse 接続プロパティ	327
MS SQL Server 接続のプロパティ	329
Netezza 接続のプロパティ	333

OData 接続のプロパティ.....	334
ODBC 接続のプロパティ.....	335
Oracle 接続のプロパティ.....	336
Salesforce 接続のプロパティ.....	339
Salesforce Marketing Cloud 接続のプロパティ.....	340
SAP 接続のプロパティ.....	341
シーケンシャル接続のプロパティ.....	344
Snowflake 接続プロパティ.....	346
Teradata Parallel Transporter 接続のプロパティ.....	348
Tableau 接続のプロパティ.....	350
Tableau V3 接続のプロパティ.....	351
Twitter Streaming 接続プロパティ.....	352
VSAM 接続のプロパティ.....	353
Web サービス接続のプロパティ.....	356
データベース接続における識別子のプロパティ.....	357
通常の識別子.....	358
区切り記号付き識別子.....	358
識別子のプロパティ.....	358
索引.....	361

序文

『Informatica® Developer tool ガイド』を使用して、Developer tool 内のユーザーインターフェースについて学びます。モデルリポジトリに接続し、接続とデータオブジェクトを作成して、アプリケーションをデプロイする方法を学習します。

Informatica のリソース

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

Informatica Network

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

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

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

Informatica ナレッジベース

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

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

Informatica マニュアル

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

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

Informatica 製品可用性マトリックス

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

Informatica Velocity

Informatica Velocity は、Informatica プロフェッショナルサービスが開発したヒントとベストプラクティスのコレクションで、多数のデータ管理プロジェクトから得た実体験に基づいています。Informatica Velocity には、世界中の組織と連携してデータ管理ソリューションを計画、開発、デプロイ、管理する Informatica コンサルタントによる集合知を表しています。

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

Informatica Marketplace

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

Informatica グローバルカスタマサポート

電話または Informatica Network を介して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 (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>) を参照してください。

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

第 1 章

Informatica Developer

この章では、以下の項目について説明します。

- [Informatica Developer の概要, 15 ページ](#)
- [Informatica Developer の起動, 18 ページ](#)
- [Informatica Developer のインタフェース, 19 ページ](#)
- [Informatica Developer の設定, 23 ページ](#)
- [ドメイン, 25 ページ](#)
- [プロジェクト, 25 ページ](#)
- [プロジェクトの権限, 26 ページ](#)
- [フォルダ, 29 ページ](#)
- [オブジェクトのコピー操作, 30 ページ](#)
- [タグ, 31 ページ](#)

Informatica Developer の概要

Developer tool は、データ統合、データ品質、データプロファイリング、およびデータサービスに関連したソリューションを設計して実装するためのアプリケーションです。

Developer tool を使用して、メタデータのインポート、接続の作成、およびデータオブジェクトの作成を実行できます。また、Developer tool を使用して、プロファイル、マッピング、およびワークフローを作成して実行することもできます。

Informatica Data Quality およびプロファイリング

Developer ツールのデータ品質機能を使用して、データの内容と構造を分析し、ビジネスニーズに合った方法でデータを強化します。

Developer ツールを使用して、次のタスクを完了するプロセスを設計および実行します。

- データのプロファイリング。プロファイリングにより、データの内容と構造が明らかになります。プロファイリングは、どのようなデータプロジェクトにおいても重要です。プロファイリングを実施すると、データの強みと弱みが明らかになり、プロジェクト計画の定義に役立つからです。
- スコアカードを作成して、データ品質を確認します。スコアカードは、プロファイルでの品質解析結果を視覚的に表したものです。

- データ値の標準化。データを標準化し、プロファイルを実行したときに見つかるエラーや不整合を除去します。句読点、形式、およびスペルの差異を標準化できます。例えば、市区町村、州、および郵便番号の値に整合性を持たせることができます。
- データの解析。解析では、複数の値で構成されるフィールドを読み取り、各値のフィールドを、含まれている情報のタイプに従って作成します。レコードに情報を追加することもできます。例えば、製品データに測定単位を追加するように解析操作を定義できます。
- 郵便アドレスの検証。アドレス検証では、郵便アドレスデータの精度と配信性を評価および強化します。アドレス検証では、アドレスレコードと国内の郵便運送業者のアドレス参照データを照合することで、アドレスの誤りを修正し、部分的なアドレスを完全なアドレスにします。アドレス検証では、郵便情報を追加して、郵便の配達に要する時間とコストを削減することもできます。
- 重複レコードの検索。重複分析では、各レコードの 1 つ以上のフィールドのデータを比較して、レコード間の類似度を計算します。分析するフィールドを選択し、データに適用する比較ストラテジを選択します。Developer ツールでは 2 種類の重複分析を実行できます。類似するレコードまたは重複するレコードを特定するフィールド一致と、レコードデータの類似する ID または重複する ID を特定する ID 一致です。
- 例外の管理。例外とは、手動で修正するデータ品質問題が含まれるレコードです。マッピングを実行し、他のデータ品質処理の実行後にデータセットに残っている例外レコードをキャプチャできます。Analyst ツールで例外レコードの確認と編集を行います。
- 参照データテーブルの作成。Informatica には、標準化や解析などのいくつかのタイプのデータ品質プロセスを強化できる参照データが用意されています。プロファイル結果のデータを使用して参照テーブルを作成できます。
- データ品質ルールの作成と実行。Informatica には、ルールが用意されています。このルールは、そのまま実行することも、プロジェクト目標に合わせて編集することもできます。Developer ツールでマップレットを作成し、ルールとして検証することができます。
- Informatica ユーザーとの共同作業。参照データとルールはモデルリポジトリに格納され、このリポジトリは Developer ツールおよび Analyst ツールで使用できます。ユーザーはプロジェクトを共同で実施できます。また、プロジェクトのさまざまな段階でさまざまなユーザーがオブジェクトの所有権を取得できます。
- Developer tool または infacmd を使用して PowerCenter からマッピングをインポートするか、マッピングを PowerCenter にエクスポートしてメタデータを再利用します。

PowerCenter からモデルリポジトリにデータをインポートするには、以下の作業を完了します。

1. PowerCenter Client または以下のコマンドを使用して、PowerCenter オブジェクトをファイルにエクスポートします。
`pmrep ExportObject`
2. 以下のコマンドを使用して、エクスポートファイルをモデルリポジトリファイルに変換します。
`infacmd ipc importFromPC`
3. Developer tool または以下のコマンドを使用して、オブジェクトをインポートします。
`infacmd tools importObjects`

モデルリポジトリから PowerCenter リポジトリにデータをエクスポートするには、以下の作業を完了します。

1. Developer tool または以下のコマンドを使用して、モデルリポジトリオブジェクトをファイルにエクスポートします。
`infacmd tools exportObjects`
または、`infacmd ipc ExportToPC` を直接実行してエクスポートすることもできます。
2. 以下のコマンドを使用して、エクスポートファイルを PowerCenter ファイルに変換します。
`infacmd ipc ExporttoPC`
3. PowerCenter または以下のコマンドを使用して、オブジェクトをインポートします。
`pmrep importObjects`

Informatica Data Services

データサービスは、データにアクセスしてそのデータを変換するために実行できる再利用可能な操作の集まりです。

Developer ツールのデータサービス機能を使用して、以下のタスクを実行できます。

- データの論理ビューの定義。データの論理ビューは、エンタープライズ内のデータの構造および用途を表します。エンタープライズで使用されているデータのタイプとそのデータの構造を示す論理データオブジェクトモデルを作成できます。
- データソースまたはターゲットへの論理モデルのマッピング。論理モデルのオブジェクトをデータソースまたはターゲットにリンクするマッピングを作成します。複数の異なるデータソースのデータをリンクすると、単一のデータビューを作成できます。モデルに準拠するデータを複数の異なるターゲットにロードすることもできます。
- データの仮想ビューの作成。仮想統合データベースを Data Integration Service にデプロイすることができます。エンドユーザーは、実際のソースデータに影響を与えることなく、仮想データに対して SQL クエリを実行することができます。
- Web サービスインタフェースを介したデータ統合機能へのアクセスの提供。Web サービスを Data Integration Service にデプロイできます。エンドユーザーは Web サービスに要求を送信し、SOAP メッセージを介して応答を受信します。
- Developer tool または infacmd を使用して PowerCenter からマッピングをインポートするか、マッピングを PowerCenter にエクスポートしてメタデータを再利用します。

PowerCenter からモデルリポジトリにデータをインポートするには、以下の作業を完了します。

1. PowerCenter Client または以下のコマンドを使用して、PowerCenter オブジェクトをファイルにエクスポートします。
`pmrep ExportObject`
2. 以下のコマンドを使用して、エクスポートファイルをモデルリポジトリファイルに変換します。
`infacmd ipc importFromPC`
3. Developer tool または以下のコマンドを使用して、オブジェクトをインポートします。
`infacmd tools importObjects`

モデルリポジトリから PowerCenter リポジトリにデータをエクスポートするには、以下の作業を完了します。

1. Developer tool または以下のコマンドを使用して、モデルリポジトリオブジェクトをファイルにエクスポートします。
`infacmd tools ExportObjects`
または、`infacmd ipc ExportToPC` を直接実行してエクスポートすることもできます。
 2. 以下のコマンドを使用して、エクスポートファイルを PowerCenter ファイルに変換します。
`infacmd ipc ExporttoPC`
 3. PowerCenter または以下のコマンドを使用して、オブジェクトをインポートします。
`pmrep importObjects`
- ドメインユーザーがコマンドラインから実行できるマッピングの作成とデプロイ。
 - データのプロファイリング。[プロファイリング] オプションを使用すると、データのプロファイリングによってデータの内容と構造が明らかになります。プロファイリングは、どのようなデータプロジェクトにおいても重要です。プロファイリングを実施すると、データの強みと弱みが明らかになり、プロジェクト計画の定義に役立つからです。

Informatica Developer の起動

Developer ツールがローカルマシンにインストールされている場合は、Windows の [スタート] メニューからツールを起動します。Developer ツールがリモートマシンにインストールされている場合は、コマンドラインからツールを起動します。

ローカルマシンでの Developer ツールの起動

ローカルマシンにインストールされた Developer ツールを起動するには、Windows の [スタート] メニューを使用します。

1. Windows の [スタート] メニューから、[すべてのプログラム] > [Informatica [バージョン]] > [クライアント] > [Developer クライアント] > [Informatica Developer の起動] をクリックします。

Developer ツールの初回実行時には、ようこそページにいくつかのアイコンが表示されます。それ以降、Developer ツールを実行しても、ようこそページは表示されません。

2. [ワークベンチ] をクリックします。

Developer tool を初めて起動するときは、ドメインとモデルリポジトリを追加する必要があります。ドメインに複数のデータ統合サービスが含まれている場合は、デフォルトのサービスも選択する必要があります。

リモートマシンでの Developer ツールの起動

リモートマシンにインストールされた Developer ツールを起動するには、コマンドラインを使用します。

Developer ツールがリモートマシンにインストールされている場合、インストールディレクトリへの書き込みアクセス権がないことがあります。そのため、Developer ツールが一時ファイルを書き込むワークスペースディレクトリをローカルマシン上に指定します。管理者は、デフォルトのローカルワークスペースディレクトリをすべてのユーザーに対して設定できます。Developer ツールの起動時に、デフォルトのディレクトリをオーバーライドできます。

設定済みのローカルのワークスペースディレクトリが存在しない場合、Developer ツールは一時ファイルを書き込むときにディレクトリを作成します。

1. コマンドプロンプトを開きます。
2. Developer ツールを起動するためのコマンドを入力します。デフォルトのローカルワークスペースディレクトリを使用することも、デフォルトのディレクトリをオーバーライドすることもできます。

- デフォルトのローカルワークスペースディレクトリを使用するには、次のコマンドを入力します：

```
\\<remote installation directory>\developer.exe
```

以下に例を示します。

```
\\MyRemoteMachine\Informatica\9.5.1\clients\DeveloperClient\developer.exe
```

- デフォルトのローカルワークスペースディレクトリをオーバーライドするには、次のコマンドを入力します：

```
\\<remote installation directory>\developer.exe -data <local workspace directory>
```

以下に例を示します。

```
\\MyRemoteMachine\Informatica\9.5.1\clients\DeveloperClient\developer.exe -data C:\temp\MyWorkspace
```

ローカルワークスペースディレクトリのフォルダ名に番号記号 (#) 文字は使用できません。ローカルワークスペースディレクトリのフォルダ名にスペースが含まれている場合は、ディレクトリ全体を二重引用符で囲みます。

Developer ツールの初回実行時には、ようこそページにいくつかのアイコンが表示されます。それ以降、Developer ツールを実行しても、ようこそページは表示されません。

3. **【ワークベンチ】** をクリックします。

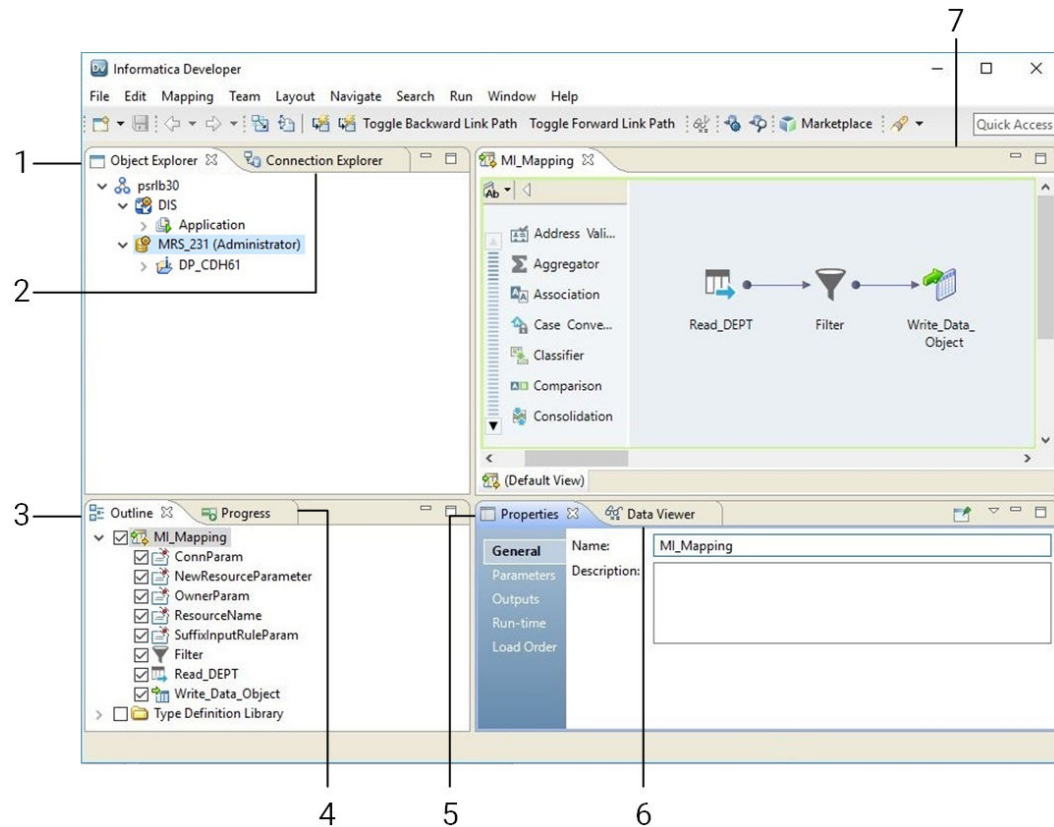
Developer tool を初めて起動するときは、ドメインとモデルリポジトリを追加する必要があります。ドメインに複数のデータ統合サービスが含まれている場合は、デフォルトのデータ統合サービスも選択する必要があります。

Informatica Developer のインタフェース

Developer tool を使用すると、データ品質およびデータサービスソリューションを設計、実装できます。

Developer tool で同時に複数のタスクを実行できます。また、同時に複数のフォルダーやプロジェクトで作業することもできます。Developer tool で作業するには、Developer tool のワークベンチにアクセスします。

次の図に Developer tool のワークベンチを示します。



1. [オブジェクトエクスプローラ] ビュー
2. [接続エクスプローラ] ビュー
3. [アウトライン] ビュー
4. [進捗状況] ビュー
5. [プロパティ] ビュー
6. [データビューア] ビュー
7. エディタ

推奨される Developer tool の画面解像度は、Windows では 1920 x 1080 です。

Informatica Developer のビュー

Developer tool のワークベンチには、エディタとビューがあります。エディタでマッピングなどのオブジェクトを編集します。Developer tool では、エディタ内で選択されているオブジェクトに応じたビューが表示されます。

追加のビューの選択、ビューの非表示、Developer tool のワークベンチ内の別の場所へのビューの移動などの操作が可能です。

表示するビューを選択するには、**【ウィンドウ】 > 【ビューの表示】** をクリックします。

デフォルトビュー

Developer tool では、デフォルトで以下のビューが表示されます。

【接続エクスプローラ】 ビュー

リレーショナルデータベースへの接続が表示されます。

【データビューア】 ビュー

ソースデータ、プロファイル結果を表示し、トランスフォーメーションの出力をプレビューします。

【オブジェクトエクスプローラ】 ビュー

ドメインと、ドメイン内の設計時オブジェクトおよびランタイムオブジェクトを表示します。設計時オブジェクトは、モデルリポジトリ内のプロジェクトおよびフォルダに保存されます。ランタイムオブジェクトは、データ統合サービス上にランタイムアプリケーションの一部として保存されます。

【アウトライン】 ビュー

【オブジェクトエクスプローラ】 ビューで選択したオブジェクトに応じたオブジェクトが表示されます。

【進捗状況】 ビュー

マッピング実行などの Developer tool での操作の進捗状況が表示されます。

【プロパティ】 ビュー

エディタで選択されたオブジェクトのプロパティが表示されます。

追加のビュー

Developer tool ワークベンチには、次のビューも表示されます。

【警告】 ビュー

接続ステータス警告が表示されます。

【監査ビューアー】 ビュー

マッピングが監査とともに実行された後に、ルールと条件の監査結果が表示されます。

【チートシート】 ビュー

開いているチートシートが表示されます。チートシートを開くには、**【ヘルプ】 > 【チートシート】** をクリックして、チートシートを選択します。

【チェックアウトされたオブジェクト】 ビュー

チェックアウトしたすべてのオブジェクトが表示されます。

【データプロセッサイベント】 ビュー

Developer tool でデータプロセッサトランスフォーメーションを実行するときに起こる初期化、実行、要約イベントに関する情報が表示されます。

【データプロセッサ 16 進数ソース】 ビュー

入力ドキュメントが 16 進形式で表示されます。

ヘルプビュー

状況依存のオンラインヘルプが表示されます。

【オブジェクトの依存関係】 ビュー

オブジェクトの表示、編集、または削除を行うときにオブジェクトの依存関係が表示されます。

【検索】 ビュー

検索結果が表示されます。また、検索オプションダイアログボックスを起動することもできます。

【タグ】 ビュー

ビジネス上の用途に基づいてモデルリポジトリ内のオブジェクトを定義するタグが表示されます。

【通知】 ビュー

ヒューマンタスクのすべての作業が完了したときにユーザーまたはグループに通知するオプションが表示されます。

【検証ログ】 ビュー

オブジェクトの検証エラーが表示されます。

【バージョン履歴】 ビュー

選択したオブジェクトのバージョン履歴が表示されます。チェックインコメントを確認し、オブジェクトチェックインに関するユーザー情報を表示できます。

Informatica Developer のようこそページ

Developer tool を初めて開くと、ようこそページが表示されます。ようこそページを使用して、Developer tool の設定方法を確認し、作業を始めてください。

ようこそページには次のオプションが表示されます。

- 概要: データ品質およびデータサービスソリューションの概要を表示するには、[概要] ボタンをクリックします。
- 最初の手順: Developer tool の設定および Informatica Data Quality Service と Informatica Data Services のレッスンへのアクセスに関する知識を深めるには、[最初の手順] ボタンをクリックします。
- チュートリアル: Developer tool のチートシートやデータ品質およびデータサービスソリューションのチートシートを表示するには、[チュートリアル] ボタンをクリックします。
- Web リソース: Informatica ナレッジベースへのリンクにアクセスするには、[Web リソース] ボタンをクリックします。Informatica How-To ライブラリにアクセスできます。Informatica How-To ライブラリには、Informatica Data Quality、Informatica Data Services、および他の Informatica 製品に関する記事が掲載されています。
- ワークベンチ: Developer tool での作業を開始するには、[ワークベンチ] ボタンをクリックします。

閉じた後にようこそページにアクセスするには、[ヘルプ] > [ようこそ] をクリックします。

チートシート

Developer ツールには、オンラインヘルプの一部としてチートシートが含まれています。チートシートは段階を追った手引きで、Developer ツールで 1 つまたは複数のタスクを完了するために役立ちます。

チートシートに従って、タスクを完了して結果を確認します。例えば、チートシートを完了して、物理データオブジェクトをインポートおよびプレビューすることができます。

チートシートにアクセスするには、**[ヘルプ]** > **[チートシート]** をクリックします。

Informatica Developer のオンラインヘルプ

Informatica Developer のオンラインヘルプシステムには、Developer tool を最大限活用するのに役立つ情報が含まれます。**[コンテンツ]**、**[検索]**、および **[インデックス]** オプションを使用して、Developer tool の特徴と機能について学習します。役に立つページがあれば、そのページへのブックマークを追加します。**[関連トピック]** オプションを使用して、同様の情報が含まれるページを見つけます。

Informatica の設定

[設定] ダイアログボックスには、Developer ツールおよび Eclipse プラットフォームの設定が含まれます。

Developer ツール内の設定を管理するには、Informatica 設定を使用します。例えば、Informatica 設定を使用して、設定、接続、トランスフォーメーション設定、タグ、使用可能な Data Integration Serviceなどを管理します。

Developer ツールは、Eclipse プラットフォーム上に構築されています。**[設定]** ダイアログには、Eclipse プラットフォームの設定を管理するための設定も含まれます。Informatica では、Informatica 設定のみがサポートされます。

Informatica 設定にアクセスするには、**[ウィンドウ]** > **[設定]** をクリックします。**[設定]** ダイアログボックスで、**[Informatica]** を選択します。

Informatica Marketplace

Informatica Marketplace は、データ統合の実装を強化、拡張、改良するための構築済みのソリューションを提供します。

Informatica Marketplace にアクセスするには、ツールバーで **[マーケットプレイス]** をクリックします。Developer tool に **[マーケットプレイス]** ビューが表示されます。

初めてマーケットプレイスにログインする前に、ユーザーとして登録する必要があります。

ログイン後、エディタで構築済みのソリューションへのリンクを表示できます。**[マーケットプレイス]** 検索ボックスでソリューションを検索し、検索結果を表示してソリューションを見つけることができます。ソリューションには、Developer tool で使用するモデルリポジトリにインポートできるマッピング、マッピングオブジェクト、プロファイルまたはワークフローが含まれている場合があります。

マーケットプレイスソリューションをインポートするには、マーケットプレイスソリューションの横にある **[インポート]** ボタンをクリックし、モデルリポジトリにソリューションをインポートする手順に従います。ソリューションをインポートするには、モデルリポジトリに接続する必要があります。ソリューションの関連ソースファイルおよびマニュアルをコピーするには、インポートプロセス中にフォルダを選択する必要があります。

モデルリポジトリにソリューションをインポートした後、マッピングを実行する、またはマッピングを実行する前に編集することができます。

また、マーケットプレイスコミュニティにソリューションを投稿し、その他のユーザーを支援することもできます。

Informatica Developer の設定

モデルリポジトリオブジェクトにアクセスするように Informatica Developer を設定します。データ統合サービスを選択し、データをプレビューして、プロファイル、マッピング、およびデータフローを実行します。

Developer ツールを設定するには、以下のタスクを実行します。

1. ドメインを追加します。
2. モデルリポジトリを追加します。
3. ドメインに複数のデータ統合サービスが含まれている場合は、デフォルトのサービスを選択します。
4. ドメインに複数のメタデータアクセスサービスが含まれている場合は、デフォルトのサービスを選択します。

Developer ツールを設定したら、作業を格納するプロジェクトとフォルダーを作成できます。

手順 1. ドメインの追加

ドメインで実行されるサービスにアクセスするには、Developer ツールでドメインを追加します。

ドメインを追加する前に、ドメインに接続するためのドメイン名、ホスト名、およびポート番号があることを確認します。この情報は管理者から提供されます。

1. **【ウィンドウ】 > 【設定】** をクリックします。
【設定】 ダイアログボックスが表示されます。
2. **【Informatica】 > 【ドメイン】** を選択します。
3. **【追加】** をクリックします。
【新しいドメイン】 ダイアログボックスが表示されます。
4. ドメイン名、ホスト名、およびポート番号を入力します。
5. **【完了】** をクリックします。
6. **【OK】** をクリックします。

手順 2. モデルリポジトリの追加

プロジェクトとフォルダーにアクセスするために、モデルリポジトリを追加します。

モデルリポジトリを追加する前に、以下の前提条件を確認します。

- 管理者が Administrator ツールで Model Repository サービスを設定済みである。
- Model Repository サービスにアクセスするためのユーザー名とパスワードがある。この情報は管理者から提供されます。

1. **【ファイル】 > 【リポジトリへの接続】** をクリックします。
【リポジトリへの接続】 ダイアログボックスが表示されます。
2. **【参照】** をクリックして、Model Repository サービスを選択します。
3. **【OK】** をクリックします。
4. **【次へ】** をクリックします。
5. ユーザー名とパスワードを入力します。
6. **【次へ】** をクリックします。
【プロジェクトを開く】 ダイアログボックスが表示されます。

7. **[Object Explorer]** ビューに表示されるプロジェクトの一覧をフィルタリングするには、開く予定のないプロジェクトを消去します。
8. **[完了]** をクリックします。
[Object Explorer] ビューのモデルリポジトリサービスに、選択したプロジェクトが表示されます。
[Object Explorer] ビューには、サービスを含むドメインも表示されます。

手順 3. デフォルトのデータ統合サービスの選択

Data Integration Service は、Developer ツールでデータ統合タスクを実行します。ドメインに複数のデータ統合サービスが含まれている場合は、デフォルトのサービスを選択します。マッピングまたはデータのプレビューを実行する際、デフォルトの Data Integration Service をオーバーライドできます。

注: ドメインに 1 つのデータ統合サービスしか含まれていない場合、この手順はオプションです。

Data Integration Service を選択する前に、ドメインを追加します。

1. **[ウィンドウ]** > **[設定]** をクリックします。
[設定] ダイアログボックスが表示されます。
2. **[Informatica]** > **[Data Integration Services]** を選択します。
3. ドメインを展開します。
4. Data Integration Service を選択します。
5. **[デフォルトとして設定]** をクリックします。
6. **[OK]** をクリックします。

手順 4. デフォルトのメタデータアクセスサービスの選択

メタデータアクセスサービスを使用すると、Developer tool は Hadoop 接続情報にアクセスして、メタデータをインポートおよびプレビューできます。ドメインに複数のメタデータアクセスサービスが含まれている場合は、デフォルトのサービスを選択します。データオブジェクトをインポートするときに、デフォルトのメタデータアクセスサービスをオーバーライドできます。

注: ドメインにメタデータアクセスサービスが 1 つしか含まれていない場合は、このステップは省略可能です。

メタデータアクセスサービスを選択する前に、ドメインを追加します。

1. **[ウィンドウ]** > **[設定]** をクリックします。
[設定] ダイアログボックスが表示されます。
2. **[Informatica]** > **[メタデータアクセスサービス]** を選択します。
3. ドメインを展開します。
4. メタデータアクセスサービスを選択します。
5. **[デフォルトとして設定]** をクリックします。
6. **[OK]** をクリックします。

ドメイン

Informatica ドメインは、Informatica 環境を定義するノードとサービスの集合です。

Developer ツールでドメインを追加します。ドメイン情報を編集したり、ドメインを削除したりすることもできます。Developer ツールの設定でドメイン情報を管理します。

プロジェクト

プロジェクトは、Developer ツールでフォルダおよびオブジェクトを格納するために使用する最上位のコンテナです。

プロジェクトを使用して、データサービスおよびデータ品質ソリューションに使用するオブジェクトを整理および管理します。

[Object Explorer] ビューで、プロジェクトを管理および表示します。プロジェクトを作成すると、Developer ツールにより、プロジェクトがモデルリポジトリ内に格納されます。

作成した各プロジェクトは、Analyst ツールにも表示されます。

以下の表に、プロジェクトに関して実行できるタスクを示します。

タスク	説明
プロジェクトの管理	プロジェクトコンテンツを管理します。プロジェクトの作成、複製、名前変更、および削除が可能です。プロジェクトコンテンツを表示することができます。
プロジェクトのフィルタ	[Object Explorer] ビューに表示されるプロジェクトの一覧をフィルタリングします。
フォルダの管理	プロジェクトコンテンツをフォルダで整理します。プロジェクト内のフォルダの作成、複製、名前変更、移動が可能です。
オブジェクトの管理	オブジェクトのコンテンツの表示のほか、プロジェクトまたはプロジェクト内のフォルダにあるオブジェクトの複製、名前変更、移動、および削除を行います。
プロジェクトの検索	プロジェクト内のフォルダやオブジェクトを検索します。検索結果を確認し、結果からオブジェクトを選択してコンテンツを表示することができます。
権限の割り当て	プロジェクトのオブジェクトを表示および編集できるユーザーとグループを選択します。他のおよびグループに権限を割り当てることができるユーザーとグループを指定します。

プロジェクトの作成

プロジェクトを作成して、オブジェクトとフォルダーを格納します。

1. **[Object Explorer]** ビューで Model Repository サービスを選択します。
2. **[ファイル] > [新規] > [プロジェクト]** をクリックします。
[新規プロジェクト] ダイアログボックスが表示されます。
3. プロジェクトの名前を入力します。
4. **[次へ]** をクリックします。

【新規プロジェクト】ダイアログボックスの【プロジェクトの権限】ページが表示されます。

5. 必要に応じて、ユーザーまたはグループを選択して権限を割り当てます。

6. 【完了】をクリックします。

【Object Explorer】ビューの【Model Repository サービス】の下にプロジェクトが表示されます。

プロジェクトのフィルタ

【Object Explorer】ビューに表示されるプロジェクトの一覧をフィルタリングできます。膨大な数のプロジェクトのうち、管理の必要なプロジェクトはほんの一部である場合などに、プロジェクトをフィルタリングします。

フィルタリングしたプロジェクトの一覧は、次にリポジトリに接続するときまで保持されます。

以下の時点でプロジェクトをフィルタリングできます。

リポジトリへの接続前

リポジトリへの接続前にプロジェクトをフィルタリングすると、リポジトリに接続するときの所要時間を短縮できます。

【ファイル】 > 【リポジトリへの接続】を選択します。リポジトリを選択した後、ユーザー名とパスワードを入力し、【次へ】をクリックします。【プロジェクトを開く】ダイアログボックスに、アクセスできるすべてのプロジェクトが表示されます。リポジトリで開くプロジェクトを選択したら【完了】をクリックします。

リポジトリへの接続後

リポジトリに接続している場合は、【ファイル】 > 【プロジェクトを閉じる】をクリックし、プロジェクトをフィルタリングして【Object Explorer】ビューから除去します。【プロジェクトを閉じる】ダイアログボックスには、現在【Object Explorer】ビューで開いているすべてのプロジェクトが表示されます。フィルタリングで除去するプロジェクトを選択し、【完了】をクリックします。

フィルタリングしたプロジェクトを開く場合は、【ファイル】 > 【プロジェクトを開く】をクリックします。

プロジェクトの権限

プロジェクト権限をユーザーまたはグループに割り当てます。プロジェクト権限は、ユーザーまたはグループがオブジェクトの表示や編集を行えるかどうか、また、他のユーザーに権限を割り当てられるかどうかを決定します。

以下の権限を割り当てることができます。

読み取り

ユーザーまたはグループは、プロジェクトのすべてのオブジェクトを開き、プレビュー、エクスポート、検証、およびデブroyを行うことができます。ユーザーまたはグループはプロジェクトの詳細を表示することもできます。

書き込み

ユーザーまたはグループは、プロジェクトのすべてのオブジェクトに対して読み取り権限があります。さらに、プロジェクトのすべてのオブジェクトの編集、プロジェクトの詳細の編集、プロジェクト内のすべてのオブジェクトの削除、プロジェクトの削除を行うことができます。

付与

ユーザーまたはグループは、プロジェクトのすべてのオブジェクトに対して読み取り権限があります。さらに、ユーザーまたはグループは権限を他のユーザーまたはグループに割り当てることができます。

Model Repository サービスの管理者ロールに割り当てられたユーザーは、Model Repository サービスのすべてのプロジェクトに対するすべての権限を継承します。グループに割り当てられたユーザーは、そのグループの権限を継承します。

外部オブジェクトの権限

権限はプロジェクト内のオブジェクトに適用されます。他のオブジェクト内に存在する依存オブジェクトまでは適用されません。

依存オブジェクトとは、他のオブジェクトで使用されるオブジェクトです。例えば、再利用不可能な式トランスフォーメーションを含むマプレットを作成したとします。この場合、マプレットは親オブジェクトです。式トランスフォーメーションは、このマプレットの依存オブジェクトです。

親オブジェクト内で再利用可能なオブジェクトを使用すると、Developer tool によりオブジェクトのインスタンスが作成されます。例えば、再利用可能なルックアップトランスフォーメーションを使用してマッピングを作成したとします。このマッピングは親オブジェクトです。このマッピングには、ルックアップトランスフォーメーションのインスタンスが含まれます。

オブジェクトは、他のプロジェクトに存在する依存オブジェクトのインスタンスを含むことができます。他のオブジェクトにある依存オブジェクトを表示するには、そのプロジェクトに対する読み取り権限が必要です。他のプロジェクトにある依存オブジェクトを編集するには、親オブジェクトプロジェクトに対する書き込み権限と他のプロジェクトに対する読み取り権限が必要です。

依存オブジェクトインスタンスに対する権限

別のプロジェクトからの依存オブジェクトインスタンスを含むオブジェクトへのアクセスが必要な場合があります。その別プロジェクトに対する読み取り権限がない場合、親オブジェクトへのアクセス権限に基づいて、異なるオプションが表示されます。

表示権限のない依存オブジェクトインスタンスを含む親オブジェクトにアクセスしようとすると、Developer ツールに警告メッセージが表示されます。操作を続行すると、操作タイプごとに異なる結果が生成されます。

以下の表に、親オブジェクトに対して実行可能な操作の結果を示します。

操作	結果
親オブジェクトを開く。	親オブジェクトを開く方法を指定するよう求められます。 <ul style="list-style-type: none">- コピーを開く。親オブジェクトのコピーが作成されます。コピーには、表示権限のない依存オブジェクトインスタンスは含まれません。- 開く。Developer ツールでオブジェクトが開かれますが、表示権限のない依存オブジェクトインスタンスは削除されます。親オブジェクトを保存すると、親オブジェクトから依存オブジェクトインスタンスが削除されます。リポジトリからは依存オブジェクトは削除されません。- キャンセル。親オブジェクトは開かれません。
親オブジェクトを XML ファイルにエクスポートすると、Developer tool で使用できるようになります。	依存オブジェクトインスタンスを含まないエクスポートファイルが作成されます。
親オブジェクトを PowerCenter にエクスポートする。	親オブジェクトをエクスポートできません。

操作	結果
親オブジェクトを検証する。	依存オブジェクトが親オブジェクトの一部でないかのように、親オブジェクトが検証されます。
親オブジェクトをデプロイする。	親オブジェクトをデプロイできません。
親オブジェクトをコピーして貼り付ける。	依存オブジェクトインスタンスを含まない新しいオブジェクトが作成されます。

セキュリティの詳細

表示権限のない依存オブジェクトインスタンスを含むオブジェクトにアクセスすると、Developer ツールに警告メッセージが表示されます。警告メッセージからは、その依存オブジェクトに関する詳細を表示できます。

依存オブジェクトに関する詳細を表示するには、警告メッセージの【詳細】ボタンをクリックします。モデルリポジトリサービスのセキュリティの詳細表示権限がある場合、表示できないオブジェクトを含むプロジェクトが表示されます。ユーザーにセキュリティの詳細表示権限がない場合、プロジェクト名を表示する権限がないことが表示されます。

親オブジェクトのアクセス

別のプロジェクトからの依存オブジェクトインスタンスを使用する親オブジェクトを作成した場合、その親オブジェクトを編集することはできません。ユーザーが親オブジェクトを編集し、その親オブジェクトの機能を保持できるようにするには、マプレットに依存オブジェクトのインスタンスを作成します。

例えば、別のプロジェクトからの再利用可能なルックアップトランスフォーメーションを含むマッピングを作成するとします。プロジェクトのユーザーにマッピングの編集を許可し、ルックアップトランスフォーメーションの編集は禁止するとします。

マッピングにルックアップトランスフォーメーションを配置すると、他のプロジェクトに対する読み取り権限がないユーザーがマッピングを開くと警告メッセージが表示されます。ユーザーはマッピングのコピーやマッピング自体を開くことはできますが、Developer ツールによってルックアップトランスフォーメーションインスタンスは削除されます。

ユーザーがマッピングを編集できるようにするには以下のタスクを実行します。

1. プロジェクトにマプレットを作成します。マプレットに入力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、および出力トランスフォーメーションを追加します。
2. マッピングを編集し、ルックアップトランスフォーメーションをマプレットで置き換えます。
3. マッピングを保存します。

プロジェクトのユーザーがマッピングを開くと、ルックアップトランスフォーメーションの代わりにマプレットが表示されます。ユーザーは、マプレット以外のマッピングを自由に編集できます。

ユーザーがマッピングをエクスポートした場合、エクスポートファイルにルックアップトランスフォーメーションは含まれません。

権限の割り当て

ユーザーとグループをプロジェクトに追加して、それらのユーザーとグループに権限を割り当てることができます。権限を割り当てて、ユーザーがプロジェクト内のオブジェクトに対して実行できるタスクを決定します。

1. **[Object Explorer]** ビューでプロジェクトを選択します。

2. **【ファイル】** > **【プロパティ】** をクリックします。
【プロパティ】 ウィンドウが表示されます。
3. **【権限】** を選択します。
4. **【追加】** をクリックしてユーザーを追加し、そのユーザーに権限を割り当てます。
【ドメインユーザーとグループ】 ダイアログボックスが表示されます。
5. ユーザーおよびグループのリストをフィルタリングするには、名前または文字列を入力します。
必要に応じて、フィルタでワイルドカード文字を使用します。
6. セキュリティドメインでフィルタリングするには、**【セキュリティドメインでフィルタ】** ボタンをクリックします。
7. **【ネイティブ】** を選択して、ネイティブセキュリティドメインのユーザーとグループを表示します。または、**【すべて】** を選択してすべてのユーザーとグループを表示します。
8. ユーザーまたはグループを選択し、**【OK】** をクリックします。
ユーザーとグループが**【新規プロジェクト】** ダイアログボックスの**【プロジェクトの権限】** ページに表示されます。
9. ユーザーとグループに対する読み取り、書き込み、または権限の付与を選択します。
10. **【OK】** をクリックします。

フォルダ

フォルダを使用して、プロジェクト内のオブジェクトを整理します。フォルダを作成して、ビジネスニーズに基づいてオブジェクトをグループ化します。例えば、フォルダを作成すると、プロジェクト内の特定のタスクに使用するオブジェクトをグループ化できます。フォルダは、プロジェクトまたは別のフォルダ内に作成することができます。

フォルダは、**【Object Explorer】** ビューのプロジェクト内に表示されます。フォルダには他のフォルダ、データオブジェクト、およびオブジェクトタイプを含めることができます。

フォルダに対しては、以下のタスクを実行できます。

- フォルダを作成する。
- フォルダを表示する。
- フォルダの名前を変更する。
- フォルダを複製する。
- フォルダを移動する。
- フォルダを削除する。

フォルダーの作成

プロジェクトに、関連するオブジェクトを格納するフォルダーを作成します。フォルダーは、プロジェクトまたは別のフォルダー内に作成する必要があります。

1. **【Object Explorer】** ビューで、フォルダを作成する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】** > **【フォルダ】** をクリックします。
【新規フォルダー】 ダイアログボックスが表示されます。

3. フォルダの名前を入力します。
4. **【完了】** をクリックします。
フォルダがプロジェクトまたは親フォルダの下に表示されます。

オブジェクトのコピー操作

オブジェクトをプロジェクト内または別のプロジェクトにコピーすることができます。また、オブジェクトを同じプロジェクト内のフォルダまたは別のプロジェクト内のフォルダにコピーすることもできます。

オブジェクトのコピーに別の名前を付けて保存できます。オブジェクトをリンクとしてコピーして、Analyst ツールにオブジェクトを表示したり、メールメッセージなどの別のメディアにオブジェクトへのリンクを設定したりすることもできます。

以下のオブジェクトを、他のプロジェクトまたはフォルダにコピーするか、オブジェクトのコピーに別の名前を付けて保存するか、またはオブジェクトをリンクとしてコピーできます。

- アプリケーション
- データサービス
- 論理データオブジェクトモデル
- Mapping
- マップレット
- 物理データオブジェクト
- プロファイル
- 参照テーブル
- 再利用可能なトランスフォーメーション
- ルール
- スコアカード
- 仮想ストアドプロシージャ
- ワークフロー

オブジェクトをコピーする場合は以下のガイドラインを使用してください。

- マッピング、マップレット、ルール、および仮想ストアドプロシージャのセグメントをコピーできます。
- フォルダを別のプロジェクトにコピーすることができます。
- 論理データオブジェクトをリンクとしてコピーできます。
- コピーしたオブジェクトは複数回貼り付けることができます。
- プロジェクトまたはフォルダに同じ名前のオブジェクトが含まれている場合は、そのオブジェクトの名前を変更するか上書きすることができます。

オブジェクトのコピー

オブジェクトをコピーして、別のプロジェクトまたはフォルダでできるようにします。

1. プロジェクトまたはフォルダ内のオブジェクトを選択します。

2. **【編集】** > **【コピー】** をクリックします。
3. オブジェクトをコピーするプロジェクトまたはフォルダを選択します。
4. **【編集】** > **【貼り付け】** をクリックします。

オブジェクトのコピーの保存

オブジェクトに別の名前を付けて保存するために、オブジェクトのコピーを保存します。

モデルリポジトリの外部の場所にある参照データ値が参照データオブジェクトで指定されている場合に、その参照データオブジェクトのコピーを保存すると、モデルリポジトリはそのオブジェクトを無効としてマークします。コピーを作成するには、**【Object Explorer】** ビューでオブジェクトを選択し、**【編集】** > **【コピー】** をクリックした後、必要な場所にオブジェクトを貼り付けます。

1. エディタでオブジェクトを開きます。
2. **【ファイル】** > **【名前を付けてコピーを保存】** をクリックします。
3. オブジェクトのコピーの名前を入力します。
4. **【参照】** をクリックして、オブジェクトをコピーするプロジェクトまたはフォルダを選択します。
5. **【完了】** をクリックします。

タグ

タグは、ビジネス上の用途に基づいてドメイン内のオブジェクトを定義するメタデータです。ビジネス上の用途に従ってオブジェクトをグループ化するには、タグを作成します。

タグを作成したら、そのタグを 1 つ以上のオブジェクトに関連付けることができます。タグとオブジェクトの関連付けを削除することもできます。タグを使用すると、タグに関連付けられているオブジェクトをドメイン内で検索できます。Developer ツールにはすべてのタグの用語集が表示されます。

例えば、XYZCorp_CustomerOrders という名前のタグを作成し、XYZ 社の顧客の注文に関する情報が格納されたテーブルに割り当てているとします。この場合、XYZCorp_CustomerOrders タグで検索を実行して、そのタグに関連付けられているテーブルを識別することができます。

タグに関連付けられているマッピングをデプロイすると、そのタグは、データ統合サービス上のランタイムバージョンのマッピングにプロパゲートされます。アプリケーションパッチを使用して、デプロイ済みのマッピングを更新すると、パッチの名前がタグとしてランタイムバージョンのマッピングと関連付けられます。

注: Developer tool でオブジェクトに関連付けられたタグは、Analyst ツールでも同じオブジェクトのタグとして表示されます。

タグの作成

タグを作成し、ビジネス上の用途に基づいてオブジェクトを定義するメタデータを追加します。

1. 以下のいずれかの方法を使用して、タグを作成します。
 - **【ウィンドウ】** - **【設定】** をクリックします。**【設定】** ダイアログボックスで、**【Informatica】** > **【タグ】** を選択します。モデルリポジトリサービスを選択し、**【追加】** をクリックします。
 - エディタでオブジェクトを開きます。**【タグ】** ビューで、**【編集】** をクリックします。**【オブジェクトのタグの割り当て】** ダイアログボックスで、**【新規】** をクリックします。
2. タグの名前を入力します。

3. 必要に応じて、説明を入力します。
4. **[OK]** をクリックします。

タグの割り当て

タグをオブジェクトに割り当て、メタデータの定義をオブジェクトに関連付けます。

1. エディタでオブジェクトを開きます。
2. **[タグ]** ビューで、**[編集]** をクリックします。
[オブジェクトのタグの割り当て] ダイアログボックスが表示されます。**[使用可能なタグ]** 領域に、リポジトリで定義されたすべてのタグが表示されます。名前または説明でタグを検索できます。**[タグの割り当て]** 領域に、開かれているオブジェクトとそのオブジェクトに割り当てられているすべてのタグが表示されます。
3. **[使用可能なタグ]** 領域で、タグを選択します。
4. **[タグの割り当て]** 領域で、オブジェクトを選択します。
5. **[割り当て]** をクリックします。
6. オブジェクトからタグを削除するには、**[使用可能なタグ]** 領域でタグを選択し、**[タグの割り当て]** 領域でオブジェクトを選択して、**[削除]** をクリックします。

タグの表示

オブジェクトに割り当てられているすべてのタグを表示したり、モデルリポジトリで定義されたすべてのタグを表示できます。

1. オブジェクトに割り当てられているタグを表示するには、エディタでそのオブジェクトを開きます。
2. **[タグ]** ビューを選択します。
[タグ] ビューに、オブジェクトに割り当てられているすべてのタグが表示されます。
3. モデルリポジトリに定義されているすべてのタグを表示するには、**[ウィンドウ]** > **[設定]** をクリックします。
[設定] ダイアログボックスが表示されます。
4. **[Informatica]** > **[タグ]** を選択します。
[タグ] 領域に、モデルリポジトリで定義されたすべてのタグが表示されます。名前または説明でタグを検索できます。

第 2 章

モデルリポジトリ

この章では、以下の項目について説明します。

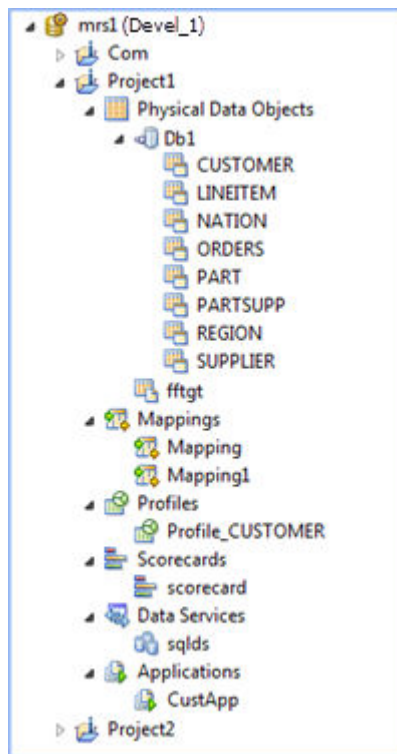
- [モデルリポジトリの概要, 33 ページ](#)
- [Informatica Developer のオブジェクト, 34 ページ](#)
- [リポジトリオブジェクトのロック, 37 ページ](#)
- [バージョン管理されたオブジェクトを使用したチームベース開発, 38 ページ](#)
- [モデルリポジトリへの接続, 43 ページ](#)
- [モデルリポジトリサービスの更新, 44 ページ](#)

モデルリポジトリの概要

モデルリポジトリは、プロジェクトおよびフォルダのメタデータを格納するリレーショナルデータベースです。

モデルリポジトリに接続し、物理データオブジェクト、マッピング、プロファイル、および他のオブジェクトを作成および編集します。アプリケーションにオブジェクトを追加し、エンドユーザーおよびサードパーティのツールでオブジェクトにアクセスできるようにアプリケーションをデプロイします。

以下の図に、**【オブジェクトエクスプローラ】** ビューで mrs1 という名前のモデルリポジトリを開いた状態を示します。



モデルリポジトリサービスでは、モデルリポジトリが管理されます。モデルリポジトリにアクセスするすべてのクライアントアプリケーションおよびアプリケーションサービスは、モデルリポジトリサービスを介して接続します。クライアントアプリケーションには Developer tool および Analyst ツールが含まれます。モデルリポジトリにアクセスする Informatica サービスには、モデルリポジトリサービス、アナリストサービス、およびデータ統合サービスが含まれます。

Developer tool を設定する場合は、モデルリポジトリを追加する必要があります。Developer ツールを開くたびに、モデルリポジトリに接続してプロジェクトおよびフォルダにアクセスします。

オブジェクトを編集するときには、排他的に編集できるように、モデルリポジトリによってそのオブジェクトがロックされます。モデルリポジトリをサードパーティ製のバージョン管理システムと統合することもできます。バージョン管理システムを統合すると、オブジェクトのチェックアウトとチェックイン、オブジェクトのチェックアウトの取り消し、オブジェクトの履歴バージョンの表示と取得を実行できます。

Informatica Developer のオブジェクト

Developer tool の【オブジェクトエクスプローラ】ビューで特定のオブジェクトを作成、管理、または表示できます。

Developer tool では、以下のモデルリポジトリオブジェクトを作成できます。

アプリケーション

データオブジェクト、マッピング、SQL データサービス、Web サービス、ワークフローを含む、デプロイ可能なオブジェクト。アプリケーションの作成、編集、削除を行うことができます。

データサービス

データにアクセスしてそのデータを変換するために実行できる、再利用可能な操作のコレクション。データサービスは、Web サービスを通じてアクセスできる、または SQL クエリを実行する対象となるデータの統合されたモデルを提供します。データサービスの作成、編集、削除を行うことができます。

データオブジェクト操作

ソースまたはターゲットで特定のランタイム操作を実行するために必要なプロパティを含むリポジトリオブジェクト。一部の PowerExchange アダプタデータソースに必要です。

フォルダ

モデルリポジトリ内のオブジェクトのコンテナ。フォルダを使用してプロジェクト内のオブジェクト体系的に管理したり、フォルダを作成してビジネスニーズに基づいてオブジェクトをグループ化したりできます。フォルダの作成、編集、削除を行うことができます。

論理データオブジェクト

論理データオブジェクトモデル内のオブジェクトであり、企業の論理エンティティを表します。論理データオブジェクトには属性とキーがあり、属性間のリレーションを表します。論理データオブジェクトモデル内の論理データオブジェクトの作成、編集、削除を行うことができます。

論理データオブジェクトマッピング

論理データオブジェクトを 1 つ以上の物理データオブジェクトにリンクするマッピング。論理データオブジェクトマッピングには、トランスフォーメーションロジックを含めることができます。論理データオブジェクトの論理データオブジェクトマッピングの作成、編集、削除を行うことができます。

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を含み、オブジェクト間のリレーションを定義するデータモデル。論理データオブジェクトモデルの作成、編集、削除を行うことができます。

マッピング

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されている入力と出力。マッピングの作成、編集、削除を行うことができます。

マップレット

複数のマッピングで使用したり、ルールとして検証可能なトランスフォーメーション一式を含む再利用可能なオブジェクト。マップレットの作成、編集、削除を行うことができます。

操作マッピング

Web サービスクライアントに対して Web サービス操作を実行するマッピング。操作マッピングには、入力トランスフォーメーション、出力トランスフォーメーション、および複数のフォルトトランスフォーメーションを含めることができます。Web サービスの操作マッピングの作成、編集、削除を行うことができます。

物理データオブジェクト

リソースからの読み取り、リソースの検索、またはリソースへの書き込みに使用されるデータの物理的な表現。物理データオブジェクトの作成、編集、削除を行うことができます。

プロファイル

ソースデータ内のパターンを検出するルールが含まれるオブジェクト。データ構造を評価し、必要なタイプの情報がデータカラムに含まれていることを確認するには、プロファイルを実行します。プロファイルの作成、編集、削除を行うことができます。

参照テーブル

参照テーブルには、一連のデータ値の標準バージョンおよび検索する可能性のある値の代替バージョンが含まれます。参照テーブルの表示と削除を行うことができます。

ルール

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。これは、プロファイルで使用する中間ストリームマップレットです。ルールの作成、編集、削除を行うことができます。

ルール仕様

1 つ以上のビジネスルールのロジックを含む再利用可能なオブジェクト。Analyst ツールのユーザーは、ルール仕様を作成し、ルール仕様オブジェクトをモデルリポジトリに保存します。[Object Explorer] ビューでルール仕様を選択し、ルール仕様をマッピングにドラッグすることができます。

ルール仕様は、Developer tool では読み取り専用オブジェクトです。ルール仕様のロジックを表示または編集するには、ルール仕様を右クリックし、Analyst ツールを開くオプションを選択します。

注: Analyst ツールのユーザーは、ルール仕様からマップレットを生成することもできます。ルール仕様と対応するマップレットは、モデルリポジトリ内の独立したオブジェクトです。マップレットは、Developer tool で編集できます。

スコアカード

ソースカラムの有効値またはプロファイル結果内のルールの出力を視覚的に表したものの。スコアカードの作成、編集、削除を行うことができます。

トランスフォーメーション

データの生成や変更を行ったり、データを渡したりする、マッピング内のリポジトリオブジェクト。各トランスフォーメーションは別の関数を実行します。トランスフォーメーションは再利用可能または再利用不可能にすることができます。トランスフォーメーションの作成、編集、削除を行うことができます。

型定義ライブラリ

Spark エンジンで実行されるマッピングの複合データ型定義を格納するポジトリオブジェクト。複合データ型定義は、構造データのスキーマを表します。[アウトライン] ビューおよびマッピングエディタで、マッピングまたはマップレットの型定義ライブラリと複合データ型定義を表示できます。

仮想スキーマ

データベース構造を定義する仮想データベースのスキーマ。SQL データサービスの仮想スキーマの作成、編集、削除を行うことができます。

仮想ストアドプロシージャ

SQL データサービスにおける一連の手順またはデータフローの命令。仮想スキーマの仮想ストアドプロシージャの作成、編集、削除を行うことができます。

仮想テーブル

仮想データベース内のテーブル。仮想スキーマ内の仮想テーブルの作成、編集、削除を行うことができます。

仮想テーブルマッピング

仮想テーブルをターゲットとして含むマッピング。仮想テーブルの仮想テーブルマッピングの作成、編集、削除を行うことができます。

ワークフロー

ビジネスプロセスを定義する一連のイベント、タスク、およびディシジョンを視覚的に表したものの。ワークフローの作成、編集、削除を行うことができます。

オブジェクトプロパティ

プロジェクト、フォルダー、またはモデルリポジトリ内のその他のオブジェクトのプロパティを表示できます。

[全般設定] タブ（**[プロパティ]** ビュー内）には、オブジェクトのプロパティが表示されます。オブジェクトのプロパティには、リポジトリ内のオブジェクトの名前、説明、場所が含まれます。オブジェクトのプロパティには、オブジェクトの作成ユーザー、最終更新ユーザー、およびイベントの発生時刻も含まれます。

オブジェクトのプロパティにアクセスするには、**[Object Explorer]** ビューでオブジェクトを選択して、**[ファイル]** > **[プロパティ]** をクリックします。

リポジトリオブジェクトのロック

オブジェクトの編集を開始すると、他のユーザーがそのオブジェクトに対する変更を保存できないように、モデルリポジトリによってそのオブジェクトがロックされます。オブジェクトを保存すると、ロック状態が保持されます。オブジェクトを閉じると、リポジトリではロックが解除されます。

オブジェクトのロックは、CI/CD パイプラインにおける継続的な統合の一貫です。CI/CD の詳細については、[「CI/CD の概要」](#)（ページ 199）を参照してください。

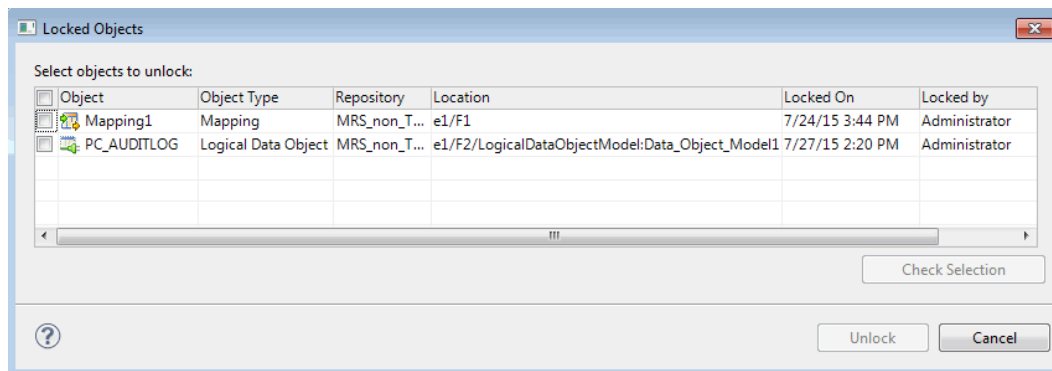
ロック管理

Developer tool が予期せず停止した場合、モデルリポジトリはオブジェクトをロックした状態で保持します。モデルリポジトリに再接続すると、ロックしたオブジェクトを表示できます。オブジェクトの編集を継続するか、オブジェクトのロックを解除することができます。

オブジェクトをロックした開発者が不在で、他の開発者がオブジェクトの開発を任された場合、オブジェクトのロックを解除することが必要になる場合があります。**[ロックされたオブジェクト]** ダイアログボックスでロックされたオブジェクトを表示してロックを解除します。

[ロックされたオブジェクト] ダイアログボックスを表示するには、**[ファイル]** > **[ロックされたオブジェクト]** をクリックします。

以下の図に、**[ロックされたオブジェクト]** ダイアログボックスを示します。



[ロックされたオブジェクト] ダイアログボックスでは1つ以上のオブジェクトのロックを解除できます。以下のいずれかのアクションを実行して、ロックを解除するオブジェクトを選択します。

- チェックボックスを使用して1つ以上のオブジェクトを選択する。
- 1つ以上の行を選択して**[選択のチェック]**をクリックし、選択した各オブジェクトのチェックボックスを有効にする。

次に、**[ロック解除]** をクリックして、選択したすべてのオブジェクトのロックを解除します。

ヒント: 作業が中断した場合は、ロックしたオブジェクトを表示し、作業中のオブジェクトを特定することができます。

ロック管理のルールとガイドライン

オブジェクトロックを管理する場合は、以下のルールとガイドラインを考慮してください。

- モデルリポジトリでは、オブジェクトを開くときにそのオブジェクトがロックされることはありません。モデルリポジトリでオブジェクトがロックされるのは、オブジェクトの編集を開始した後だけです。例えば、編集可能なフィールドにカーソルを挿入するときやマッピングオブジェクトに接続するとき、マッピング仕様がロックされます。
- オブジェクトの開発には複数のクライアントツールを使用できます。例えば、1 台のマシンでオブジェクトを編集し、その後で別のマシンでそのオブジェクトを開いて編集を続けることができます。最初のマシンに戻る場合は、オブジェクトエディタを閉じてもう一度開くことによってロックを再取得する必要があります。管理特権を持つユーザーが、別のユーザーが開いたオブジェクトのロックを解除するときにも、この同じ原則が適用されます。
- 編集のためにオブジェクトを開いた状態で、モデルリポジトリサービスを再起動した場合は、オブジェクトのロックが解除されます。オブジェクトのロックを再取得するまで、別のユーザーがオブジェクトを開いて編集することができます。オブジェクトのロックを再取得するには、オブジェクトへの変更を保存してから閉じ、次に、編集のためにオブジェクトを再度開きます。
- オブジェクトを含むフォルダを削除する場合、どのオブジェクトの削除も許可されていなければ、フォルダを削除することはできません。例えば、あるオブジェクトに対するロックを所有していないためにそのオブジェクトを削除できない場合、そのオブジェクトとフォルダは残ります。
- 複数の開発者が同時に SQL データサービスオブジェクトのコンテンツを開き、編集することができます。例えば、SQL データサービスを UserA が開いて編集を開始し、次に UserB が同じオブジェクトを開いて編集を開始することができます。UserB が UserA より先にオブジェクトを保存して閉じると、モデルリポジトリは UserA がオブジェクトを保存するまで競合の可能性を UserA に伝えません。この場合、UserA は別の名前で SQL データサービスを保存することで変更を保存できます。
- 管理者は、ロックされたオブジェクトに対する書き込み権限を取り消したり、ロックを別のユーザーに再割り当てしたりできます。この場合、オブジェクトの編集や保存はできません。オブジェクトは別のユーザー名で保存できます。

バージョン管理されたオブジェクトを使用したチームベース開発

チームベース開発は、モデルリポジトリをサードパーティ製のバージョン管理システムと統合するものです。バージョン管理システムでは、複数のオブジェクトバージョンが保存され、各バージョンにバージョン番号が割り当てられます。

オブジェクトバージョンの管理は、Developer tool を使用して行います。オブジェクトのチェックアウトとチェックイン、オブジェクトのバージョン履歴の表示と取得、チェックアウトの取り消しなどのアクションを実行できます。

モデルリポジトリでは、開発チームの他のメンバによって上書きされないようにオブジェクトが保護されます。別のユーザーによってすでにチェックアウトされているオブジェクトを開くと、そのオブジェクトをチェックアウトしたユーザーを特定する通知が送られてきます。チェックアウト済みのオブジェクトは、読み取り専用モードで開くことができるほか、別の名前で保存することもできます。

バージョン管理システムへの接続がアクティブなとき、モデルリポジトリには各オブジェクトの最新バージョンが含まれています。

バージョン管理システムへの接続を失うと、モデルリポジトリはチェックアウトされたオブジェクトの状態を維持します。バージョン管理システムとの接続が遮断している間、ユーザーはオブジェクトのオープン、編集、保存、およびクローズを継続できます。モデルリポジトリによってオブジェクト状態が追跡および維持されません。

接続が復旧した時点で、バージョン管理システムに関連するアクション（オブジェクトのチェックインやオブジェクトのチェックアウトの取り消しなど）を再開できます。接続が遮断していた間にオブジェクトを開いて編集した場合は、モデルリポジトリによってそのオブジェクトがユーザーに対してチェックアウトされます。

注: SQL データサービスオブジェクトはバージョン管理に含まれません。

オブジェクトのバージョン管理は、CI/CD パイプラインの継続的な統合の一貫です。CI/CD の詳細については、[「CI/CD の概要」 \(ページ 199\)](#)を参照してください。

バージョンされたオブジェクトの管理

モデルリポジトリがバージョン管理システムと統合されている場合は、オブジェクトのバージョンを管理できます。例えば、オブジェクトのチェックアウトとチェックイン、チェックアウトの取り消し、チェックアウトしたオブジェクトの表示などが可能です。

以下のアクションを実行できます。

オブジェクトをチェックアウトする。

オブジェクトをチェックアウトすると、そのオブジェクトは、チェックインするかチェックアウトを取り消すまでチェックアウト状態を維持します。チェックアウトされたオブジェクトは、**「チェックアウトされたオブジェクト」** ビューで表示できます。オブジェクトをチェックアウトするには、最初にオブジェクトを編集するか、**「オブジェクトエクスプローラ」** でオブジェクトを右クリックして **「チェックアウト」** を選択します。

オブジェクトのチェックアウトを取り消す。

チェックアウトを取り消すと、変更は適用されず、バージョン番号やバージョン履歴が増分されずに、オブジェクトがチェックインされます。チェックアウト後にオブジェクトに対して行った変更はすべて失われます。チェックアウトを取り消すには、**「チェックアウトされたオブジェクト」** ビューまたはオブジェクトの右クリックメニューを使用できます。例えば、チェックアウトを取り消すことによって、オブジェクトに加えた変更を削除できます。

注: オブジェクトがチェックアウトされているときにユーザーがオブジェクトを移動し、チェックアウトを取り消すと、オブジェクトは現在の場所に残り、そのバージョン履歴が再開します。チェックアウトを取り消しても、チェックアウト前の場所にはリストアされません。

オブジェクトのチェックイン。

オブジェクトをチェックインすると、バージョン管理システムによってバージョン履歴が更新され、バージョン番号が増分されます。チェックインコメントを最大 4KB の制限内で追加できます。オブジェクトをチェックインするには、**「チェックアウトされたオブジェクト」** ビューまたはオブジェクトの右クリックメニューを使用できます。

注: Perforce および SVN バージョン管理システムでは、パスの長さを 260 バイトに制限する Windows の制限のために、ファイルパスの長さが制限されます。この制限を超えるパスを持つファイルをチェックインしようとする、Perforce はエラーを返します。パスの長さが 260 バイトの場合、使用しているシステムの文字セットに応じて、パスの長さが変わります。

オブジェクトを削除する。

バージョンされたオブジェクトを削除するには、まずそのオブジェクトをチェックアウトする必要があります。削除アクションを行う時点でオブジェクトをチェックアウトしていないと、モデルリポジトリによ

ってそのオブジェクトがチェックアウトされ、削除対象としてマークされます。削除アクションを完了するには、そのオブジェクトをチェックインする必要があります。

バージョンされたオブジェクトを削除すると、バージョン管理システムによってすべてのバージョンが削除されます。

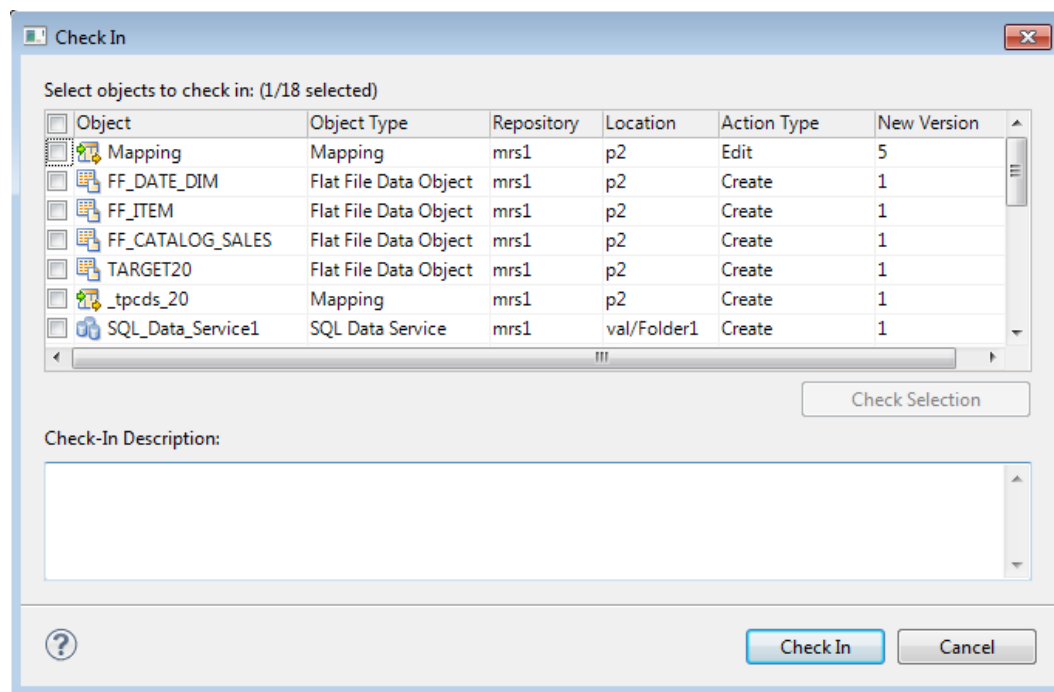
オブジェクトを削除するには、**[チェックアウトされたオブジェクト]** ビューまたはオブジェクトの右クリックメニューを使用できます。

[チェックイン] ダイアログボックス

[チェックイン] ダイアログボックスには、チェックアウトしたすべてのオブジェクトが表示されます。

[チェックイン] ダイアログボックスを使用してオブジェクトをチェックインします。またこれを使用して、削除対象としてマークしたオブジェクトを削除することもできます。

以下の図に、**[チェックイン]** ダイアログボックスを示します。



【アクションタイプ】列には、各アイテムをチェックインするときにそのアイテムに対してモデルリポジトリサービスが実行するアクションが表示されます。モデルリポジトリサービスは以下のいずれかのアクションタイプを実行します。

アクションタイプ	説明
作成	このオブジェクトは新しいオブジェクトであり、チェックインされたことがありません。これを選択し、【チェックイン】をクリックすると、モデルリポジトリサービスは、このオブジェクトをバージョン管理システムにチェックインし、このオブジェクトにバージョン番号 1 を割り当てます。
削除	オブジェクトが削除対象としてマークされます。これを選択し、【チェックイン】をクリックすると、モデルリポジトリサービスはこのオブジェクトを削除します。
編集	オブジェクトが編集対象としてマークされます。これを選択し、【チェックイン】をクリックすると、モデルリポジトリサービスは、バージョン管理システム内のバージョンを新しいバージョンに置き換えます。
移動	オブジェクトをチェックアウトすると、オブジェクトが 1 つのモデルリポジトリプロジェクトまたはフォルダから別のモデルリポジトリプロジェクトまたはフォルダに移動します。これを選択し、【チェックイン】をクリックすると、モデルリポジトリサービスは、このオブジェクトを新しい場所のバージョン管理システムにチェックインします。

【場所】列には、オブジェクトの現在の場所が表示されます。オブジェクトを移動した場合、【アクションタイプ】は【移動】です。

ヒント: 【バージョン履歴】ビューでは、移動したオブジェクトの以前の場所を確認できます。

【新しいバージョン】列には、チェックイン完了後のオブジェクトのバージョン番号が表示されます。

【チェックイン】ダイアログボックスでは、1 つ以上のオブジェクトをチェックインできます。以下のいずれかのアクションを実行して、チェックインするオブジェクトを選択します。

- チェックボックスを使用して 1 つ以上のオブジェクトを選択する。
- 1 つ以上の行を選択して【選択のチェック】をクリックし、選択した各オブジェクトのチェックボックスを有効にする。

オプションで、【チェックインの説明】テキストペインで説明を追加できます。次に【チェックイン】をクリックすると、選択したすべてのオブジェクトをチェックインできます。

バージョンされたオブジェクトの削除

バージョンされたオブジェクトを削除する場合は、削除するオブジェクトに印を付けてチェックインします。

1. 【オブジェクトエクスプローラ】ビューまたは【チェックアウトされたオブジェクト】ビューでオブジェクトを右クリックし、【削除】を選択します。
アクション属性が【編集】から【削除】に変わります。
2. オブジェクトを選択して【チェックイン】を選択します。
モデルリポジトリからオブジェクトが削除されます。

オブジェクトの履歴バージョン

バージョン管理システムによって、オブジェクトのチェックインのたびにオブジェクトのコピーが保存されます。

バージョン管理システムは、オブジェクトのバージョンを保存し、そのバージョンにバージョン番号を割り当てます。最も大きな番号が最新のバージョンになります。

【バージョン履歴】 ビューを使用してオブジェクトのバージョン履歴を表示し、読み取り専用ビューで履歴バージョンを開いて、オブジェクトの履歴バージョンをリストアします。オブジェクトの履歴バージョンをリストアし、以前の機能をリストアすることができます。

オブジェクトの履歴バージョンをリストアすると、そのバージョンが最新バージョンになります。モデルリポジトリは、オブジェクトの現行バージョンを履歴バージョンに置き換え、オブジェクトをチェックアウトします。オブジェクトがエディタで開かれると、Developer tool によってオブジェクトがリストアしたバージョンに更新されます。

オブジェクトの履歴バージョンのリストア

バージョン管理システムからオブジェクトの履歴バージョンを表示および取得できます。

1. **【オブジェクトエクスプローラ】** のオブジェクトを右クリックし、**【バージョン履歴の表示】** を選択します。

【改訂履歴】 ビューが表示されます。

2. 履歴バージョンを表示するには、**【改訂履歴】** に一覧表示されるバージョンの 1 つを右クリックし、**【バージョンの表示】** を選択します。

Developer tool では、オブジェクトの読み取り専用バージョンが開きます。バージョン番号はエディタウィンドウのタイトルバーに表示されます。

3. 履歴バージョンをリストアするには、**【改訂履歴】** に一覧表示されるバージョンの 1 つを右クリックし、**【バージョンをリストア】** を選択します。

オブジェクトがエディタで表示されます。オブジェクトを編集したり、編集せずにチェックインしたりすることができます。

4. オブジェクトをチェックインします。

リストアされたオブジェクトのバージョンは、モデルリポジトリおよびバージョン管理システムで最新バージョンになります。

【チェックアウトされたオブジェクト】 ビュー

【チェックアウトされたオブジェクト】 ビューには、チェックアウトしたオブジェクトがすべて一覧表示されます。

【チェックアウトされたオブジェクト】 ビューでは、以下のアクションを実行できます。

- オブジェクトのチェックアウトの取り消し
- オブジェクトのチェックイン

これらのどちらかを実行するには、オブジェクトを右クリックし、アクションを選択します。

【アクションタイプ】 列で削除対象としてマークされたオブジェクトを削除するには、オブジェクトを右クリックし、**【チェックイン】** を選択します。

[バージョン履歴] ビュー

[バージョン履歴] ビューには、選択したオブジェクトのバージョン履歴が表示されます。チェックインコメントを確認し、オブジェクトチェックインに関するユーザー情報を表示できます。

このビューには以下の情報が表示されます。

- バージョン。バージョンが保存された順序を表示する整数。バージョン管理システムでは、最新のチェックインバージョンに最も大きな番号を割りてます。
- 名前。オブジェクトの名前。
- 場所。モデルリポジトリ内のオブジェクトへのパス。
- アクションタイプ。オブジェクトのチェックインの際にユーザーが実行したアクション。オブジェクトのチェックインでは、オブジェクトを追加、編集、または削除することができます。
- ユーザー。オブジェクトをチェックインしたチームメンバーのユーザー ID。
- バージョン日付。ユーザーがオブジェクトをチェックインした日時。
- バージョンの説明。選択したバージョンのチェックインコメント。

バージョン履歴を表示するには、[オブジェクトエクスプローラ] のオブジェクトを右クリックし、[バージョン履歴の表示] を選択します。

チームベース開発のトラブルシューティング

チームベースの開発に関連する機能を使用する場合は、以下のトラブルシューティングのヒントを考慮してください。

Perforce バージョン管理システムが一部のオブジェクトのチェックインに失敗し、オブジェクトパス名が長すぎるというエラーが表示されます。

ファイルパスの文字数に関する Windows OS の制限のため、長いパスおよびファイル名を持つモデルリポジトリオブジェクトをチェックインしようとする失敗します。「送信が強制終了されました」という Perforce エラーメッセージが表示され、ファイルパスが内部の長さの制限を超えたことが示されます。

この問題を回避するには、モデルリポジトリ内のプロジェクト名、フォルダ名、オブジェクト名の長さを制限します。各インスタンスに短い名前を使用することで、オブジェクトパス名の合計文字数を制限することができます。

モデルリポジトリへの接続

Developer ツールを開くたびに、モデルリポジトリに接続してプロジェクトおよびフォルダーにアクセスします。モデルリポジトリに接続するときに、接続情報を入力して、モデルリポジトリを管理するモデルリポジトリサービスを含むドメインにアクセスします。

1. [Object Explorer] ビューで、モデルリポジトリを右クリックして [接続] をクリックします。

[リポジトリへの接続] ダイアログボックスが表示されます。

2. ドメインユーザー名とパスワードを入力します。
3. 名前空間を選択します。
4. [OK] をクリックします。

Developer ツールがモデルリポジトリに接続します。Developer ツールにリポジトリ内のプロジェクトが表示されます。

モデルリポジトリサービスの更新

モデルリポジトリで新しいオブジェクトと更新されたオブジェクトを確認するために、モデルリポジトリサービスを更新できます。

モデルリポジトリサービスを更新するには、**[Object Explorer]** ビューの「モデルリポジトリサービス」を右クリックし、**[更新]** を選択します。

第 3 章

Informatica Developer での検索

この章では、以下の項目について説明します。

- [Informatica Developer での検索の概要, 45 ページ](#)
- [ドメイン検索, 45 ページ](#)
- [ビジネス用語集の検索, 47 ページ](#)
- [エディタでの検索, 48 ページ](#)

Informatica Developer での検索の概要

Informatica Developer で検索を行い、オブジェクトの検索とビジネス用語のルックアップを行えます。

[Object Explorer] ビューに表示されるドメインを検索して、最新バージョンのオブジェクトおよびオブジェクトのプロパティを見つけることができます。

Developer tool から Business Glossary Desktop にアクセスして Developer tool のオブジェクト名をビジネス用語として検索できます。エディタ内のオブジェクト、ポート、グループ、式、属性などを検索することもできます。

ドメイン検索

ドメインでオブジェクトやオブジェクトプロパティを検索できます。

検索クエリを作成して検索結果をフィルタリングすることができます。検索結果を確認し、結果からオブジェクトを選択してコンテンツを表示することができます。検索結果は**【検索】**ビューに表示されます。検索で 2048 個を超えるオブジェクトが見つかった場合には、結果は表示できません。2048 を超える数のオブジェクトが結果に含まれるために検索が失敗したときは、検索条件に一致するオブジェクトの数が少なくなるように検索オプションを変更します。

以下の表に、オブジェクトの検索に使用可能な検索オプションを一覧表示します。

検索オプション	説明
含まれるテキスト	検索対象のオブジェクトまたはプロパティ。正確な文字列を入力するか、ワイルドカードを使用します。大文字と小文字は区別されません。
名前	名前を含む 1 つ以上のオブジェクト。正確な文字列を入力するか、ワイルドカードを使用します。大文字と小文字は区別されません。
タグ	タグを使用する 1 つ以上のオブジェクト。正確な文字列を入力するか、ワイルドカードを使用します。大文字と小文字は区別されません。
検索	検索対象の 1 つ以上のオブジェクトタイプ。
スコープ	ワークスペースまたは選択したオブジェクトを検索します。

検索エンジンが、ドメイン内のメタデータのインデックスを作成します。メタデータのインデックスを適切に作成するために、検索エンジンでは、インデックスを作成するメタデータの言語に適した検索アナライザが使用されます。Developer tool は、検索エンジンを使用して設計時オブジェクトおよびランタイムオブジェクトの検索を行います。

検索はさまざまな言語で実行できます。別の言語で検索する場合は、管理者が検索アナライザを変更して、その検索アナライザを使用するように以下のモデルリポジトリを設定する必要があります。

- 設計時オブジェクトを保存するモデルリポジトリ。
- ランタイムオブジェクトのデプロイ先のデータ統合サービスと関連付けられているモデルリポジトリ。

注: オブジェクトまたはオブジェクトプロパティを検索するには、**[Object Explorer]** ビューでオブジェクトに接続されている必要があります。検索を追加のオブジェクトに拡張するには、モデルリポジトリ、またはオブジェクトを含むランタイムアプリケーションに接続します。

検索のガイドライン

ドメインを検索する際は、以下のガイドラインを参照してください。

- 設計時オブジェクトは検索する前に保存する必要があります。
- 検索では、ポートと動的ポートは返されますが、生成されたポートは返されません。
- マッピングを検索すると、検索パターンに似た名前を持つマップレットが検索結果に表示されます。
- モデルリポジトリサービスまたはランタイムアプリケーションを更新すると、**[検索]** ビューの検索結果は表示されなくなります。更新されたモデルリポジトリサービスまたは更新されたランタイムアプリケーションに基づいて、ドメインを再度検索して結果を得る必要があります。

オブジェクトとプロパティの検索

ドメインでオブジェクトやプロパティを検索します。

1. **[検索]** > **[オブジェクト検索]** の順にクリックします。
[検索] ダイアログボックスが表示されます。
2. 検索対象のオブジェクトまたはプロパティを入力します。必要に応じて、ワイルドカード文字を含めます。
3. オブジェクト内のプロパティを検索する場合、必要に応じて 1 つまたは複数の名前またはタグをカンマで区切って入力することができます。
4. 必要に応じて、検索するオブジェクトタイプを選択します。

5. ワークスペースの中から検索するか、選択したオブジェクトの中から検索するかを選択します。
6. **【検索】** をクリックします。
検索結果が **【検索】** ビューに表示されます。
7. **【検索】** ビューで、オブジェクトをダブルクリックして **【エディタ】** で開きます。
オブジェクトがランタイムアプリケーションの一部である場合、そのオブジェクトは読み取り専用です。

ビジネス用語集の検索

Developer ツールのオブジェクト名の意味を、ビジネス用語として Business Glossary Desktop でルックアップし、そのビジネス要件と現在の実装を理解します。

ビジネス用語集とは、ビジネスユーザー向けに、ビジネス言語を使用して概念を定義した用語の集まりです。ビジネス用語は、概念のビジネス上の定義および使用方法を説明したものです。Business Glossary Desktop は、ビジネス用語集をホストする Metadata Manager サービスに接続するクライアントです。Business Glossary Desktop を使用して、ビジネス用語集内のビジネス用語をルックアップします。

Business Glossary Desktop がご使用のマシンにインストールされている場合は、Developer ツールでオブジェクトを選択し、ホットキーか **【検索】** メニューを使用して、ビジネス用語集でオブジェクトの名前をルックアップします。オブジェクトの名前をルックアップするには、**【Object Explorer】** ビューなど、Developer tool のビューを使用します。カラム、プロファイル、トランスフォーメーションポートの名前をルックアップするにはエディタを使用します。

例えば、開発者が Developer ツール内の Sales_Audit データオブジェクトに対応するビジネス用語をビジネス用語集で検索するとします。開発者は、そのビジネス用語の詳細を確認して、ビジネス要件と、Developer ツールにおける Sales_Audit オブジェクトの現在の実装を理解したいと考えています。これは、そのデータオブジェクトの意味や、オブジェクトへの実装が必要な可能性がある変更を把握ことができます。

ビジネス用語集デスクトップのルックアップ

ビジネス用語集デスクトップは、ビジネス用語集でオブジェクト名をルックアップし、オブジェクト名に一致するビジネス用語を返します。

オブジェクト名にハイフン、アンダースコア、大文字が含まれていると、ビジネス用語集デスクトップでは、名前が2つに分割されます。

例えば、開発者が Sales_Audit という名前のデータオブジェクトを検索する場合、ビジネス用語集デスクトップでは検索ボックスに Sales_Audit と表示されますが、Sales と Audit に名前を分割して両方のビジネス用語を検索します。

ビジネス用語のルックアップ

ビジネス要件と現在の実装を理解するため、Developer ツールのオブジェクト名をビジネス用語としてビジネス用語集デスクトップでルックアップします。

ビジネス用語集デスクトップは、使用するマシンにあらかじめインストールしておく必要があります。

1. オブジェクトを選択します。
2. ホットキーを使用するか、**【検索】** メニューを使用するかを選択して、ビジネス用語集デスクトップを開きます。

- ホットキーを使用する場合は、以下のホットキーの組み合わせを使用します。
CTRL+Shift+F
 - [検索] メニューを使用する場合は、[検索] > [Business Glossary] をクリックします。
- ビジネス用語集デスクトップが表示され、オブジェクト名に一致するビジネス用語が表示されます。

ビジネス用語のルックアップ用にホットキーをカスタマイズ

ホットキーをカスタマイズして、ビジネス用語集デスクトップを開くキーの組み合わせを変更します。

1. Developer ツールのメニューから、[ウィンドウ] > [設定] > [全般] > [キー] をクリックします。
2. [Business Glossary の検索] をコマンドのリストで見ついたり、検索したりするには、以下のいずれかを選択します。
 - 検索してキーを探すには、検索ボックスに「ビジネス用語集検索」と入力します。
 - スクロールしてキーを探すには、[コマンド] カラムでスクロールして [Business Glossary の検索] コマンドを見つけます。
3. [Business Glossary の検索コマンド] をクリックします。
4. [コマンドのバインド解除] をクリックします。
5. [バインディング] フィールドで、キーの組み合わせを入力します。
6. [適用] をクリックしてから、[OK] をクリックします。

エディタでの検索

エディタ内で開いているオブジェクト、ポート、グループ、式、属性を検索できます。Developer ツールで開いているエディタ内のオブジェクトがハイライトされます。オブジェクトをモデルリポジトリに置く必要はありません。

エディタの下に検索フィールドを表示するには、[編集] > [検索/置換] を選択します。オブジェクトを検索するには、検索対象オブジェクトの文字列および型を入力します。検索可能なオブジェクトタイプは、エディタによって異なります。オブジェクトのタイプを指定しない場合は、トランスフォーメーション内の検索文字列が検索されます。

ポート、カラム、または属性を検索する場合、データ型も選択できます。例えば、文字列「_ID」を含む名前を持つ integer 型または bigint 型のポートを検索できます。

以下の表に、各エディタで検索可能なオブジェクトのタイプを示します。

エディタ	オブジェクトタイプ
マッピング	マッピングオブジェクト、式、グループ、ポート、複合データ型の定義
マップレット	マップレットオブジェクト、式、グループ、ポート、複合データ型の定義
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および属性
物理データオブジェクトの読み取りまたは書き込みマッピング	オブジェクトおよびカラムのマッピング

エディタ	オブジェクトタイプ
SQL データサービス	仮想テーブルおよび属性
仮想ストアドプロシージャ	トランスフォーメーション、式、グループ、ポート
仮想テーブルマッピング	仮想テーブルのマッピングオブジェクト、式、グループ、ポート
Web サービス操作のマッピング	Web サービス操作のマッピングオブジェクト、式、グループ、ポート
ワークフロー	ワークフローオブジェクト
注: モデルリポジトリには生成されたポートの名前は格納されないため、エディタでポートを検索することはできません。	

検索文字列が見つかったら、そのオブジェクトの場所が表示されます。また、検索文字列が出現するオブジェクトもハイライトされます。検索文字列がマッピングエディタ内のアイコン化したトランスフォーメーションに含まれている場合は、アイコン化したトランスフォーメーションがハイライトされます。

以下のオプションを選択して、検索の結果をナビゲートすることができます。

- 次の一致。検索文字列の次のオカレンスを検索します。
- 直前の一致。検索文字列の直前のオカレンスを検索します。
- すべてハイライト。検索文字列のすべてのオカレンスをハイライトします。
- アイコン化したトランスフォーメーションの展開。検索文字列を含む、アイコン化したすべてのトランスフォーメーションを展開します。

第 4 章

接続

この章では、以下の項目について説明します。

- [接続の概要, 50 ページ](#)
- [\[接続エクスプローラ\] ビュー, 52 ページ](#)
- [接続管理, 53 ページ](#)
- [接続の切り替え, 55 ページ](#)
- [サードパーティ製 JDBC ドライバ, 59 ページ](#)
- [環境 SQL, 59 ページ](#)

接続の概要

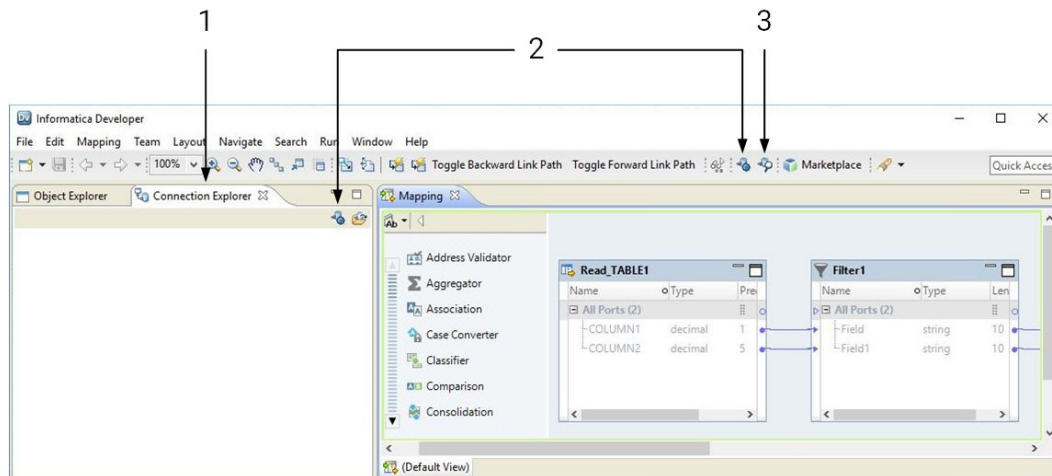
接続は、ドメイン環境設定リポジトリ内の接続を定義するリポジトリオブジェクトです。

接続を作成することで、データオブジェクトのインポート、データのプレビュー、データのプロファイリング、およびマッピングの実行が行えます。Developer ツールは、データオブジェクトをインポートする際に接続を使用します。データ統合サービスは、データのプレビュー、マッピングの実行、または Web サービスの使用時に接続を使用します。

注: Developer tool は、フラットファイルデータオブジェクトのインポートや、フラットファイルデータのプレビュー、読み取り、または書き込みに接続を使用しません。

Developer ツールは接続をドメイン環境設定リポジトリに格納します。Developer ツールで作成した接続は、Analyst ツールや Administrator ツールで使用できます。

【設定】ダイアログボックスまたは【接続エクスプローラ】ビューで接続を作成および管理します。



1. 【接続エクスプローラ】ビュー
2. 接続の作成
3. 接続の表示

接続を作成したら、その接続に対して以下のアクションを実行できます。

接続を編集する。

接続名および説明を変更できます。接続の詳細（ユーザー名、パスワード、および接続文字列など）の編集もできます。

データ統合サービスは、接続 ID で各接続を識別します。したがって、接続名を変更することができます。接続の名前を変更すると、Developer ツールは、その接続を使用しているオブジェクトを更新します。

デプロイされたアプリケーションおよびパラメータファイルは、接続 ID ではなく名前で接続を識別します。したがって、接続の名前を変更する場合、その接続を使用するすべてのアプリケーションを再デプロイする必要があります。さらに、接続パラメータを使用するパラメータファイルもすべて更新する必要があります。

接続をコピーする。

接続をコピーして、別の接続に類似の接続を作成できます。例えば、ユーザー名とパスワードのみが異なる 2 つの Oracle 接続を作成できます。

接続を削除する。

接続を削除すると、その接続を使用するオブジェクトは無効になります。接続を誤って削除した場合、削除した接続と同じ接続 ID で接続を再作成できます。

接続リストを更新する。

接続リストを更新して、ドメインの最新の接続リストを確認できます。ユーザーが Administrator ツールまたは Analyst ツールで接続を追加、削除、または名前変更した場合は、接続リストを更新します。

接続タイプ

接続を使用して、データソースに対してデータの読み取りと書き込みを実行できます。

非ネイティブ環境、Hadoop、Databricks にアクセスするための接続を作成します。HBase、HDFS、あるいは Hadoop 環境の Hive ソースまたはターゲットにアクセスする場合は、これらの接続も作成する必要があります。Developer tool、Administrator ツール、および infacmd を使用すると、接続を作成できます。

以下の種類の接続を作成できます。

Hadoop 接続

Hadoop 環境でマッピングを実行できるように Hadoop 接続を作成します。

HBase 接続

HBase 接続を作成して HBase にアクセスします。HBase 接続は NoSQL 接続です。

HDFS 接続

Hadoop クラスタの HDFS ファイルシステムとの間でデータを読み込み、書き込むための HDFS 接続を作成します。

Hive 接続

ソースまたはターゲットとして Hive に接続できるように Hive 接続を作成します。ネイティブ環境または Hadoop 環境でマッピングを有効にすると、ソースとして Hive にアクセスできます。Blaze エンジンでマッピングを実行すると、ターゲットとして Hive にアクセスできます。

JDBC 接続

JDBC 接続を作成し、Sqoop を介してリレーショナルデータのインポートとエクスポートができるようにこの接続で Sqoop プロパティを設定します。

Databricks 接続

Databricks 環境でマッピングを実行できるように Databricks 接続を作成します。

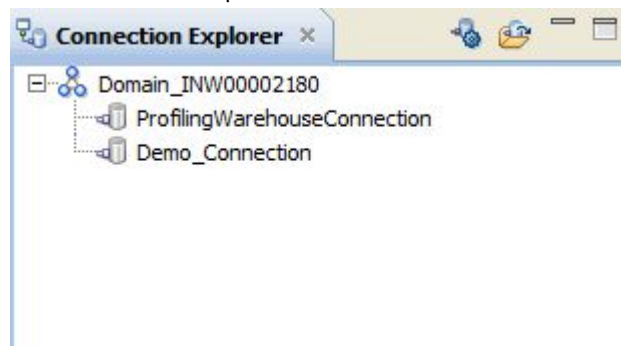
注: ソーシャルメディア web サイトや Teradata など、他のソースまたはターゲットへの接続の作成に関する詳細については、対応する PowerExchange アダプタユーザーガイドを参照してください。

各データソースのサポート対象バージョンについては、[Informatica Product Availability Matrix](#) を参照してください。

[接続エクスプローラ] ビュー

リレーショナルデータベース接続の表示およびリレーショナルデータオブジェクトの作成には、**[接続エクスプローラ]** ビューを使用します。

以下の図に、Developer ツールでの **[接続エクスプローラ]** ビューを示します。



[接続エクスプローラ] ビューでは、以下のタスクを実行できます。

- ビューに接続を追加する。**[接続の選択]** ボタンをクリックして 1 つ以上の接続を選択し、**[接続エクスプローラ]** ビューに追加します。
- リレーショナルデータベースに接続する。データベースを右クリックして、**[接続]** をクリックします。
- リレーショナルデータベースから切断する。データベースを右クリックして、**[切断]** をクリックします。

- リレーショナルデータオブジェクトの作成。リレーショナルデータベースに接続し、テーブルが表示されるようにデータベースを拡大します。テーブルを右クリックし、**【プロジェクトに追加】** をクリックして **【新しいリレーショナルデータオブジェクト】** ダイアログボックスを開きます。
- 接続の更新。接続を右クリックして、**【更新】** をクリックします。
- デフォルトスキーマのみ表示。接続を右クリックして、**【デフォルトのスキーマのみを表示】** をクリックします。デフォルトでは有効になっています。
- **【接続エクスプローラ】** ビューからの接続の削除。接続はモデルリポジトリに残ります。接続を右クリックして、**【削除】** をクリックします。

接続管理

【設定】 ダイアログボックスまたは **【接続エクスプローラ】** ビューで接続を作成および管理します。

接続の作成

計算クラスタ、データベース、エンタープライズアプリケーション、ファイルシステム、非リレーショナルデータベース、NoSQL データベース、ソーシャルメディアアプリケーション、または Web サービスにアクセスするための接続を作成します。物理データオブジェクトのインポート、データのプレビュー、データのプロファイリング、またはマッピングを行う前に、接続を作成します。

1. **【ウィンドウ】** > **【設定】** をクリックします。
2. 作成する接続のタイプを選択します。
 - 非 Web サービス接続を選択するには、**【Informatica】** > **【接続】** を選択します。
 - Web サービス接続を選択するには、**【Informatica】** > **【Web サービス】** > **【接続】** を選択します。
3. **【使用可能な接続】** リストでドメインを展開します。
4. **【使用可能な接続】** リストで接続タイプを選択して、**【追加】** をクリックします。
【新しい<接続タイプ>接続】 ダイアログボックスが表示されます。
5. 次の情報を入力します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明（オプション）。
場所	接続が存在するドメイン。
タイプ	Oracle、Twitter、または Web サービスなど、特定の接続タイプ。

6. **【次へ】** をクリックします。

7. 接続のプロパティを設定します。
 8. **【テスト接続】** をクリックして、入力した接続のプロパティが正しく、データベース、アプリケーション、ファイルシステム、または URI に接続できることを確認します。
注: **【テスト接続】** は Databricks ではサポートされていません。
 9. **【完了】** をクリックします。
- 接続を作成したら、**【接続エクスプローラ】** ビューに追加できます。

接続の編集

接続名、説明、および接続プロパティを編集できます。

1. **【ウィンドウ】** > **【設定】** をクリックします。
2. 編集する接続のタイプを選択します。
 - 非 Web サービス接続を選択するには、**【Informatica】** > **【接続】** を選択します。
 - Web サービス接続を選択するには、**【Informatica】** > **【Web サービス】** > **【接続】** を選択します。
3. **【使用可能な接続】** リストでドメインを展開します。
4. **【使用可能な接続】** で接続を選択して、**【編集】** をクリックします。
【接続の編集】 ダイアログボックスが表示されます。
5. 必要に応じて、接続名と説明を編集します。
注: 接続名を変更する場合、その接続を使用するすべてのアプリケーションを再デプロイする必要があります。接続パラメータを使用するパラメータファイルもすべて更新する必要があります。
6. **【次へ】** をクリックします。
7. 必要に応じて、接続のプロパティを編集します。
8. **【テスト接続】** をクリックして、入力した接続のプロパティが正しく、データベースに接続できることを確認します。
9. **【OK】** をクリックして、**【接続の編集】** ダイアログボックスを閉じます。
10. **【OK】** をクリックして、**【設定】** ダイアログボックスを閉じます。

接続のコピー

ドメイン内、またはドメイン間で接続をコピーできます。

1. **【ウィンドウ】** > **【設定】** をクリックします。
2. コピーする接続のタイプを選択します。
 - 非 Web サービス接続を選択するには、**【Informatica】** > **【接続】** を選択します。
 - Web サービス接続を選択するには、**【Informatica】** > **【Web サービス】** > **【接続】** を選択します。
3. **【使用可能な接続】** リストでドメインを展開します。
4. **【使用可能な接続】** で接続を選択して、**【コピー】** をクリックします。
【接続のコピー】 ダイアログボックスが表示されます。
5. 接続名および ID を入力し、ドメインを選択します。
名前と ID は、ドメイン内で一意である必要があります。
注: Hadoop、HDFS、HBase、Hive、または Databricks 接続を別のドメインにコピーする前に、その接続に関連付けられているクラスタ構成に一致する名前を使用してドメイン内にクラスタ構成を作成します。

例えば、接続が XYZ という名前のクラスタ構成に関連付けられている場合は、ターゲットドメインに XYZ という名前のクラスタ構成を作成します。

6. **[OK]** をクリックして、**[接続のコピー]** ダイアログボックスを閉じます。
7. **[OK]** をクリックして、**[設定]** ダイアログボックスを閉じます。

接続の削除

[設定] ダイアログボックスで接続を削除すると、モデルリポジトリから接続が削除されます。

1. **[ウィンドウ]** > **[設定]** をクリックします。
2. 削除する接続のタイプを選択します。
 - 非 Web サービス接続を選択するには、**[Informatica]** > **[接続]** を選択します。
 - Web サービス接続を選択するには、**[Informatica]** > **[Web サービス]** > **[接続]** を選択します。
3. **[使用可能な接続]** リストでドメインを展開します。
4. **[使用可能な接続]** で接続を選択して、**[削除]** をクリックします。
5. **[OK]** をクリックして、**[設定]** ダイアログボックスを閉じます。

接続リストの更新

ドメインの接続の最新のリストを確認するには、接続リストを更新します。

1. **[ウィンドウ]** > **[設定]** をクリックします。
2. 更新する接続のタイプを選択します。
 - 非 Web サービス接続を選択するには、**[Informatica]** > **[接続]** を選択します。
 - Web サービス接続を選択するには、**[Informatica]** > **[Web サービス]** > **[接続]** を選択します。
3. **[使用可能な接続]** リストでドメインを選択します。
4. **[更新]** をクリックします。
5. **[使用可能な接続]** リストでドメインを展開して、接続の最新のリストを表示します。
6. **[OK]** をクリックして、**[設定]** ダイアログボックスを閉じます。

接続の切り替え

リレーショナルデータオブジェクトまたはカスタマイズデータオブジェクトの接続を切り替えて、異なるリレーショナルデータベース接続を使用できます。複数のデータオブジェクトの接続を同時に切り替えることもできます。接続を切り替えると、新しい接続を使用するために各マッピングを更新する必要がないため、時間と労力が節約されます。

接続を切り替えた後に、Developer tool はデータオブジェクトに基づいてすべての読み取り、書き込み、および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の接続の詳細を更新します。Developer tool は、新しい接続が参照するデータベースに基づいてデータオブジェクトのデータベースタイプも更新します。

次のいずれかの接続タイプに接続を切り替えることができます。

- IBM DB2
- Microsoft SQL Server

- ODBC
- Oracle

接続を切り替えるとき、Developer tool はメタデータの互換性を検証しません。このため、接続を切り替える前に、新しい接続が参照するデータベースに、接続を切り替えるデータオブジェクトと同じカラムとメタデータを持つテーブルが含まれることを確認する必要があります。新しく参照されるデータベースに同じカラムとメタデータが含まれていない場合、データの消失またはデータの矛盾が発生する可能性があります。

例

Developer tool で Oracle リレーショナルデータオブジェクトを作成して、書き込みトランスフォーメーションとして複数のマッピングに追加しました。

Oracle データベースを IBM DB2 データベースに移行します。Developer tool で既存のマッピングを更新して、データを IBM DB2 データベースに書き込みます。

各マッピングで書き込みトランスフォーメーションが基にしているデータオブジェクトを置き換える代わりに、Oracle データオブジェクトの接続を IBM DB2 の接続に切り替えることができます。新しい接続はデータを書き込む IBM DB2 データベースを指しています。Oracle リレーショナルデータオブジェクトを含むマッピングを実行すると、データ統合サービスは新しい IBM DB2 接続を使用してマッピングを実行します。

接続を切り替える前に

接続を切り替える前に、以下の要件が満たされていることを確認します。

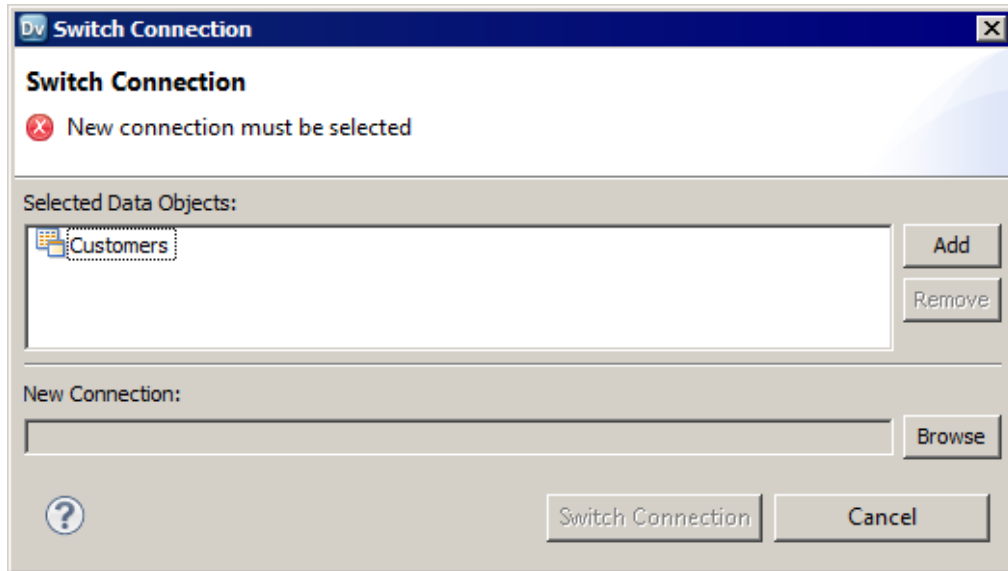
- 接続を切り替えるデータオブジェクトに対する書き込み権限がある。
- 新しい接続が参照するデータベースに、接続を切り替えるデータオブジェクトと同じカラムとメタデータを持つテーブルが含まれる。テーブル内のカラムの精度とスケールがデータオブジェクトのそれらよりも低い場合、データの損失や不整合が発生する可能性があります。
- 接続を切り替えるデータオブジェクトに、保存されていない変更が含まれない。保存されていない変更が含まれる場合、Developer tool は接続を切り替えません。
- 接続を切り替えるカスタマイズデータオブジェクトに、異なるデータベースタイプのリレーショナルデータオブジェクトが含まれない。含まれる場合、接続を切り替えるときに Developer tool はエラーを表示しません。ただし、そのカスタマイズデータオブジェクトを使用してマッピングを実行した場合、マッピングは失敗します。
- 元の接続と新しい接続の参照先のテーブル名が大文字小文字を含め正確に一致する。引用識別子および大文字小文字の混在したテーブル名が接続でサポートされる場合、Developer tool は大文字小文字を区別してテーブル名を処理します。
- 接続を切り替えるデータオブジェクトがバージョン管理されたりボジトリに保存されている場合、そのオブジェクトがチェックアウトされている。

接続の切り替え

リレーショナルデータオブジェクトまたはカスタマイズデータオブジェクトの接続を切り替えて別のリレーショナルデータベース接続を使用し、同時に既存のマッピングで新しい接続が使用されるように更新できます。

1. **【オブジェクトエクスプローラ】** ビューで、接続を切り替えるデータオブジェクトを右クリックします。
1 つのプロジェクトまたは複数のプロジェクトの別々のフォルダにある複数のデータオブジェクトを選択できます。
接続を選択して、その接続を使用するすべてのリレーショナルデータオブジェクトおよびカスタマイズデータオブジェクトの接続を同時に切り替えることもできます。
2. **【接続の切り替え】** をクリックします。

【接続の切り替え】 ダイアログボックスが表示され、選択したデータオブジェクトが表示されます。



3. 【追加】 をクリックして、接続を切り替えるデータオブジェクトのリストを更新します。
オブジェクトを削除するには、オブジェクトを選択して【削除】 をクリックします。
4. 【新しい接続】 フィールドの横にある【参照】 をクリックします。
Developer tool で、データオブジェクトに使用できる接続が表示されます。
5. データオブジェクトに使用する新しい接続を選択して、【OK】 をクリックします。
6. 【接続の切り替え】 をクリックします。
新しい接続が参照するデータベースに、接続を切り替えるデータオブジェクトと同じカラムとメタデータを持つテーブルが含まれることを確認するように求めるメッセージが表示されます。
7. 【OK】 をクリックして、データオブジェクトの接続を切り替えます。
接続が正常に切り替えられたことを示すメッセージが表示されます。Developer tool は、そのデータオブジェクトに関連付けられているすべての Developer tool オブジェクトのデータオブジェクトの接続の詳細およびデータベースタイプを更新します。そのデータオブジェクトを含むマッピングを実行すると、データ統合サービスが新しい接続を使用します。

接続を切り替えた後に

接続を切り替えた後、データオブジェクトプロパティを確認し、必要に応じて手動でプロパティを編集します。

接続の切り替え後、次のタスクを実行します。

- データ型マッピングの検証。
- テーブル所有者名の検証。
- ルックアップトランスフォーメーションの検証。
- ヒントの再設定。
- データオブジェクトの同期。

データ型マッピングの検証

接続を切り替える際に、Developer tool は元の接続および新しい接続が参照するデータベース間のデータ型の最適な一致を特定します。マッピング内のデータオブジェクトおよびトランスフォーメーションのデータ型

が、データオブジェクトに基づいてそれぞれ設定されます。マッピングを実行する前に、データ型を確認して必要な場合はそれらを手動で更新します。

例えば、Oracle から Microsoft SQL Server に接続を切り替えるとします。固定精度のデータ型の場合、デフォルトでは、Developer tool は Microsoft SQL Server データベースの精度に基づいて精度を設定します。一方、可変精度のデータ型の場合、Developer tool は Oracle データベースに基づいて精度およびスケールを設定します。Timestamp with Time Zone のようなデータ型の場合、Developer tool は、データ型を Varchar (0,0)、または Microsoft SQL Server データベースが Varchar としてサポートしている同等のデータ型に設定します。

テーブルオーナー名の確認

接続を切り替える場合、Developer tool によりデータオブジェクトのテーブル所有者名が保持されます。必要に応じてデータオブジェクトプロパティのテーブル所有者名を手動で編集できます。

テーブル所有者名を空白に設定し接続を切り替えると、新規の接続が指すデータベースのデフォルトスキーマまたはパブリックスキーマにそのテーブルが存在する場合のみ、データを正常にプレビューできます。そうでない場合、データプレビューは失敗します。データを正常にプレビューするには、データオブジェクトのランタイムプロパティのテーブル所有者名を手動で更新する必要があります。

ルックアップトランスフォーメーションの検証

接続を切り替えた後、必要に応じて、データオブジェクトに基づきルックアップトランスフォーメーションのルックアップキャッシュを再構築し、ルックアップ条件の検証を行います。

次のタスクを実行します。

ルックアップキャッシュの再構築

ルックアップキャッシュを保持し、関連付けられたデータオブジェクトの接続の切り替えを行うようにデータ統合サービスを設定する場合、ルックアップトランスフォーメーションを更新して新規の接続に基づくルックアップキャッシュを再構築する必要があります。再構築しないと、マッピング実行時にキャッシュファイルが異なるデータベース接続で作成されたというエラーが発生します。

ルックアップ条件の検証

接続を切り替えると、Developer tool は元の接続と新規の接続が指すデータベース間で最適なデータ型の一致を識別し、データ型を適切に設定します。接続を切り替えると、ルックアップ条件は、データ型が変更になるため、有効ではないことがあります。ルックアップ条件を検証し手動で更新する必要があります。

例えば、IBM DB2 ソーステーブル、ルックアップテーブル、およびターゲットテーブルを含むマッピングを作成するとします。ソーステーブルおよびルックアップテーブルの Integer カラムでルックアップ条件を設定します。IBM DB2 から Oracle ヘルックアップテーブルの接続を切り替えると、ルックアップテーブルの Integer カラムのトランスフォーメーションデータ型が Decimal に変更されます。Integer カラムを Decimal カラムと比較できないため、ルックアップ条件は無効になります。

ヒントの再設定

カスタマイズデータオブジェクトの接続を切り替えた場合、Developer tool では、そのカスタマイズデータオブジェクトに設定されたヒントを保持しません。このため、ヒントを手動で再設定する必要があります。

データオブジェクトの同期

接続を切り替えると、Developer tool はアクティブな参照キー制約のみを保持します。

複数のデータオブジェクトの接続を同時に切り替える場合、データオブジェクトを同期して、正確なキーリレーションを維持する必要があります。

データオブジェクトに複数のテーブルが含まれ、それらのテーブル間に循環参照キー制約がある場合に、テーブルのサブセットの接続を別のデータベースタイプに切り替えると、Developer テーブルはエラーを表示することなく接続を切り替えます。ただし、データオブジェクトのキーリレーションを表示すると、Developer tool には、元のデータベースを参照するキーリレーションが表示されます。キーリレーションを更新し、新しいデータベースを参照させるには、データオブジェクトを同期する必要があります。

サードパーティ製 JDBC ドライバ

JDBC を介してソース、ターゲット、およびルックアップに接続する場合は、サードパーティ製の JDBC タイプ 4 のドライバをインストールして設定します。

Developer ツールでメタデータをインポートする際は、サードパーティ製の JDBC ドライバ jar ファイルを次の場所にコピーします。

```
<Informatica インストールディレクトリ>\clients\externaljdbcjars
```

データのプレビュー、プロファイル、またはマッピングを実行するには、JDBC ドライバ jar ファイルを次の場所にコピーします。

```
<Informatica インストールディレクトリ>\externaljdbcjars
```

環境 SQL

Integration Service は自動コミットモードで環境 SQL を実行し、SQL の発行後にトランザクションを閉じます。読み込みまたは書き込み処理中に開いているトランザクションに依存しない SQL コマンドを使用します。たとえば、ソースデータベースが読み取り専用モードに設定されており、ソース接続内にトランザクションを読み取り専用で設定する環境 SQL 文を作成した場合、Integration Service はこの SQL の実行後にコミットを発行するため、読み取り専用モードでソースを読み取ることができません。

接続環境 SQL またはトランザクション環境 SQL を設定できます。

ソース、ターゲット、ルックアップ、およびストアドプロシージャの接続に対して、環境 SQL を使用します。SQL 構文が有効でない場合、Integration Service はデータベースに接続せず、セッションは失敗します。

注: 接続オブジェクトに「環境 SQL」が含まれていると、接続は「接続環境 SQL」を使用します。

接続環境 SQL

このカスタム SQL 文字列は、以降のトランザクションの環境を設定します。Integration Service は、データベースに接続するたびに、接続環境 SQL を実行します。接続環境 SQL をターゲット接続内に設定し、そのパイプラインに 3 つのパーティションを設定した場合、Integration Service は、ターゲットデータベースへのそれぞれの接続に 1 回ずつ、SQL を合計で 3 回実行します。読み込みまたは書き込み処理中に開いているトランザクションに依存しない SQL コマンドを使用します。

たとえば、接続期間に対して引用識別子パラメータを設定するには、以下の SQL 文を使用します。

```
SET QUOTED_IDENTIFIER ON
```

SQL 文は以下の場合に使用します。

- 二重引用符がオブジェクト識別子となるよう接続環境を設定する場合。

- ターゲットロードタイプを [ノーマル] に設定し、Microsoft SQL Server ターゲット名にスペースが含まれる場合。

トランザクション環境 SQL

このカスタム SQL 文字列も環境を設定しますが、Integration Service はトランザクション環境 SQL を各トランザクションの開始時に実行します。

読み込みまたは書き込み処理中に開いているトランザクションに依存する SQL コマンドを使用します。たとえば、トランザクション環境 SQL として次の文を使用して、セッションが文字を処理する方法を変更できます。

```
ALTER SESSION SET NLS_LENGTH_SEMANTICS=CHAR
```

このコマンドは、各トランザクションの前に実行する必要があります。パラメータを接続ごとに 1 回設定するだけでは不十分であるため、このコマンドは接続環境 SQL には適していません。

環境 SQL の設定に関するガイドライン

SQL 文を作成するときは、以下のガイドラインに従ってください。

- 接続オブジェクトに関連付けられているデータベースで有効な任意の SQL コマンドを入力することができます。Integration Service では、データベース側で許されているとしても、ネストされたコメントは許されません。
- SQL エディタで SQL を入力するときは、SQL 文を入力します。
- 複数の文を区切るにはセミコロン (;) を使用します。
- /*...*/内のセミコロンは無視されます。
- コメントの外部でセミコロンを使用する必要がある場合は、バックスラッシュ (\) でセミコロンをエスケープします。
- パラメータおよび変数は、環境 SQL で使用できます。パラメータファイルで定義可能なパラメータまたは変数タイプを使用します。SQL 文内にパラメータまたは変数を入力したり、パラメータまたは変数を環境 SQL として使用することができます。たとえば、接続またはトランザクション環境 SQL としてセッションパラメータ \$ParamMyEnvSQL を使用したり、パラメータファイル内の SQL 文に \$ParamMyEnvSQL を設定できます。
- DB2 接続の接続環境 SQL 内で、sqlid を使用してテーブルオーナー名を設定できます。ただし、ターゲットインスタンス内のテーブルオーナー名により、環境 SQL 内の SET sqlid 文が無効になります。SET sqlid 文に指定されているテーブルオーナー名を使用するには、[ターゲット名のプレフィックス] に名前を入力しないでください。

第 5 章

物理データオブジェクト

この章では、以下の項目について説明します。

- [物理データオブジェクトの概要, 61 ページ](#)
- [物理データオブジェクトのタイプ, 62 ページ](#)
- [リレーショナルデータオブジェクト, 62 ページ](#)
- [カスタマイズデータオブジェクト, 66 ページ](#)
- [ターゲットテーブルの作成または置換, 72 ページ](#)
- [カスタムクエリ, 75 ページ](#)
- [非リレーショナルデータオブジェクト, 87 ページ](#)
- [WSDL データオブジェクト, 89 ページ](#)
- [同期化, 93 ページ](#)
- [物理データオブジェクトのトラブルシューティング, 96 ページ](#)

物理データオブジェクトの概要

物理データオブジェクトは、データソースからの読み取り、データソースのルックアップ、またはデータソースへの書き込みに使用されるデータの物理的な表現です。データオブジェクトソースが変更された場合、物理データオブジェクトを同期させることができます。物理データオブジェクトを同期化すると、Developer ツールによってオブジェクトメタデータが再インポートされます。

物理データオブジェクトは、プロジェクトやフォルダー内に作成できます。プロジェクトやフォルダー内の物理データオブジェクトは再利用可能なオブジェクトです。再利用可能なオブジェクトはどのタイプのマッピング、マップレット、またはプロファイルでも使用できますが、マッピング、マップレット、またはプロファイル内のデータオブジェクトを変更することはできません。物理データオブジェクトを更新するには、プロジェクトまたはフォルダー内のオブジェクトを編集します。

物理データオブジェクトは、マッピング、マップレット、またはプロファイルに含めることができます。物理データオブジェクトは、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として、マッピングまたはマップレットに追加できます。物理データオブジェクトを論理データオブジェクトマッピングに追加して、論理データオブジェクトにマッピングできます。

SQL データサービスを定義する際に、物理データオブジェクトを仮想テーブルマッピングに含めることもできます。また、Web サービスを定義する際に、操作マッピングに物理データオブジェクトを含めることもできます。

物理データオブジェクトのタイプ

データを読み取る/書き込むデータソースの種類に基づいて、さまざまな種類の物理データオブジェクトを作成できます。

物理データオブジェクトには、次のようなタイプがあります。

リレーショナルデータオブジェクト

ソースとしてリレーショナルテーブル、ビュー、またはシノニムを使用する物理データオブジェクト。例えば、Oracle ビューからリレーショナルデータオブジェクトを作成できます。

オブジェクトタイプに応じて、マッピングまたはマップレットにソース、ターゲット、またはルックアップトランスフォーメーションとしてリレーショナルデータオブジェクトを追加できます。

カスタマイズデータオブジェクト

ソースとして1つ以上の関連するリレーショナルリソースまたはリレーショナルデータオブジェクトを使用する物理データオブジェクト。リレーショナルリソースには、テーブル、ビュー、およびシノニムが含まれます。例えば、プライマリキー-外部キーリレーションを持つ2つの Microsoft SQL Server テーブルから、カスタマイズデータオブジェクトを作成できます。

データの結合、行のフィルタリング、ポートのソート、または再利用可能なデータオブジェクトでカスタムクエリの実行を行う場合は、カスタマイズデータオブジェクトを作成します。

非リレーショナルデータオブジェクト

ソースとして非リレーショナルデータベースリソースを使用する物理データオブジェクト。例えば、VSA ソースから非リレーショナルデータオブジェクトを作成できます。

フラットファイルデータオブジェクト

ソースとしてフラットファイルを使用する物理データオブジェクト。区切りフラットファイルまたは固定長フラットファイルから、フラットファイルデータオブジェクトを作成できます。

WSDL データオブジェクト

ソースとして WSDL ファイルを使用する物理データオブジェクト。

Informatica PowerExchange® アダプタは、SAP、Salesforce、Netezza などの物理データオブジェクトを作成するために使用できる多くのデータソースへのアクセスも可能にします。

リレーショナル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は、リレーショナルテーブル、ビュー、またはシノニムをソースとして持つ物理データオブジェクトです。リレーショナルデータオブジェクトは、マッピング、マップレット、またはプロファイルに含めることができます。

リレーショナルデータオブジェクトは、データベース内のリソースを説明するものです。リポジトリにリレーショナルデータオブジェクトを追加する場合、ソースデータベースからリレーショナルデータオブジェクトをインポートします。リポジトリにリレーショナルデータオブジェクトをインポートした後、その定義を変更できます。ポートの追加と削除や、プライマリキーの定義ができるほか、リポジトリ内の複数のリレーショナルデータオブジェクト間のリレーションを設定できます。デフォルトの接続、データベースの所有者、およびリソース名の変更またはパラメータ化を行うことができます。また、[すべての NULL 可能を有効化] または [すべての NULL 可能を無効化] アイコンを使用して、カラムのすべての NULL 可能値を選択またはクリアすることもできます。

リレーショナルデータオブジェクトをマッピング、マップレット、またはプロファイルに追加する場合、読み取りオブジェクトまたは書き込みオブジェクトのどちらを作成するかを示します。オブジェクトがソース、ターゲット、またはルックアップテーブルのいずれであるかに基づいて、異なるランタイムプロパティを設定できます。

以下の図に、エディタで開いているサンプルのリレーショナルデータオブジェクトを示します。

Overview

General

Name: ReportDefinition_RelationalDataObject

Description:

ReportDefinition_Re...

Name Native T ^

DEALERSHIP_ID number

REGION varchar2

CUSTOMER_NO number

FIRSTNAME varchar2

Columns

Search

	Name	Native Type	Precision	Scale	Primary Key	Nullable	Description
1	DEALERSHIP_ID	number	15	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2	REGION	varchar2	10	0	<input type="checkbox"/>	<input type="checkbox"/>	
3	CUSTOMER_NO	number(p,s)	5	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	FIRSTNAME	varchar2	20	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	LASTNAME	varchar2	20	0	<input type="checkbox"/>	<input type="checkbox"/>	

Overview

Keys

Relationships

Advanced

ソースデータベース内でのリレーシンの有無にかかわらず、リレーショナルデータオブジェクト間の、プライマリキー-外部キーのリレーシオンを作成できます。

複数のリレーショナルデータオブジェクトを、ソースとしてマッピングまたはマップレットに含めることができます。同時に複数のリレーショナルデータオブジェクトを追加する場合、以下のどちらかの方法でオブジェクトを追加するように Developer ツールにメッセージが表示されます。

- 関連するデータオブジェクトとして。複数のリレーショナルリソースを持つ読み取りトランスフォーメーションが 1 つ作成されます。読み取りトランスフォーメーションは、カスタマイズデータオブジェクトと同じ機能を持ちます。
- 独立したデータオブジェクトとして。リレーショナルデータオブジェクトごとに読み取りトランスフォーメーションが 1 つ作成されます。読み取りトランスフォーメーションは、リレーショナルデータオブジェクトと同じ機能を持ちます。

次のタイプのリレーショナルデータオブジェクトをインポートすることができます。

- DB2 for i5/OS
- DB2 for z/OS
- HAWQ
- IBM DB2
- JDBC
- Microsoft SQL Server
- Netezza

- ODBC
- Oracle
- SAP HANA

リレーショナルデータオブジェクトのインポート

リレーショナルデータオブジェクトをインポートし、マッピング、マップレット、またはプロファイルに追加します。

リレーショナルデータオブジェクトをインポートする前に、データベースへの接続を設定する必要があります。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】 > 【新規】 > 【データオブジェクト】** をクリックします。
【新規】 ダイアログボックスが表示されます。
3. **【リレーショナルデータオブジェクト】** を選択して、**【次へ】** をクリックします。
【新しいリレーショナルデータオブジェクト】 ダイアログボックスが表示されます。
4. **【接続】** オプションの横にある **【参照】** をクリックして、データベースへの接続を選択します。
5. **【既存のリソースからデータオブジェクトを作成】** をクリックします。
6. **【リソース】** オプションの隣にある **【参照】** をクリックします。
【リソースの選択】 ダイアログボックスが表示されます。
7. インポートする必要があるテーブル、ビュー、同義語を選択します。
8. データオブジェクトをフィルタリングするときは、**【フィルタ】** セクションで名前を入力し、**【検索】** をクリックします。
 検索クエリでは、疑問符 (?) を 1 文字のワイルドカード、アスタリスク (*) を複数文字のワイルドカードとして使用できます。
注: 検索結果は最大 20,000 件表示されます。
9. インポートするリソースと同じ名前の物理データオブジェクトがすでに存在する場合、**【インポートの競合の解決】** ダイアログボックスが表示されます。インポートの競合の解決方法を選択します。別の名前で物理データオブジェクトを作成するか、既存のオブジェクトを再利用するか、または既存のオブジェクトを置き換えることができます。
【すべての競合に適用する】 を選択すると、同じ解決方法がすべてのインポート競合に適用されます。
10. 物理データオブジェクトの名前を入力します。
11. **【場所】** オプションの横にある **【参照】** をクリックして、リレーショナルデータオブジェクトをインポートするプロジェクトを選択します。
12. **【完了】** をクリックします。
 データオブジェクトが、**【Object Explorer】** ビューのプロジェクトまたはフォルダにある **【物理データオブジェクト】** の下に表示されます。

キーのリレーション

リレーショナルデータオブジェクト間にはキーリレーションを作成できます。キーリレーションを使用すると、リレーショナルデータオブジェクトをカスタマイズデータオブジェクトでデータソースとして使用する場合、またはマッピングやマップレットで読み取りトランスフォーメーションとして使用する場合に、リレーショナルデータオブジェクトを結合できます。

リレーショナルデータオブジェクトをインポートする場合、データベースに定義されているプライマリキー情報は保持されます。関連するリレーショナルデータオブジェクトを同時にインポートする場合、外部キーと

キーリレーションも保持されます。ただし、関連するリレーショナルデータオブジェクトを個別にインポートする場合は、オブジェクトをインポートした後にキーリレーションを再作成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間にキーリレーションを作成するには、最初に、参照先オブジェクトにプライマリキーを作成します。次に、外部キーを含むリレーショナルデータオブジェクトにリレーションを作成します。

作成したキーリレーションは、リレーショナルデータオブジェクトのメタデータに存在します。ソースのリレーショナルリソースを変更する必要はありません。

リレーショナルデータオブジェクトのキーの作成

キーカラムは、リレーショナルデータオブジェクトの各行を識別するために作成します。リレーショナルデータオブジェクトごとに1つのプライマリキーを作成できます。

1. リレーショナルデータオブジェクトを開きます。
2. **【キー】** ビューを選択します。

以下の図に、エディタ内で開いているリレーショナルデータオブジェクト例の**【キー】** ビューを示します。

3. **【追加】** をクリックします。
【新しいキー】 ダイアログボックスが表示されます。
4. キー名を入力します。
5. キーがプライマリキーの場合、**【プライマリキー】** を選択します。
6. キーカラムを選択します。
7. **【OK】** をクリックします。
8. リレーショナルデータオブジェクトを保存します。

リレーショナルデータオブジェクト間のリレーションの作成

リレーショナルデータオブジェクト間にはキーリレーションを作成できます。リレーショナルデータオブジェクトとカスタマイズデータオブジェクト間にキーリレーションを作成することはできません。

参照するリレーショナルデータオブジェクトにはプライマリキーが必要です。

1. 外部キーを作成するリレーショナルデータオブジェクトを開きます。
2. **【リレーション】** ビューを選択します。
3. **【追加】** をクリックします。
【新しいリレーション】 ダイアログボックスが表示されます。
4. 外部キーの名前を入力します。

5. 参照先リレーショナルデータオブジェクトからプライマリキーを選択します。
6. **[OK]** をクリックします。
7. **[リレーション]** プロパティで、外部キーカラムを選択します。
8. リレーショナルデータオブジェクトを保存します。

カスタマイズデータオブジェクト

カスタマイズデータオブジェクトは、1つ以上のリレーショナルソースを持つ物理データオブジェクトです。データ統合サービスでのソースデータの読み込み時に、データのジョイン、行のフィルタリング、ポートのソート、カスタムクエリの実行を行う場合は、カスタマイズデータオブジェクトを作成します。カスタマイズデータオブジェクトは、マッピング、マップレット、またはプロファイルで再利用することができます。

カスタマイズデータオブジェクトはプロジェクトまたはフォルダーに作成できます。カスタマイズデータオブジェクトは、マッピング、マップレット、またはプロファイルから変更することはできません。プロジェクトまたはフォルダ内のカスタマイズデータオブジェクトを変更すると、そのオブジェクトを使用するすべてのマッピング、マップレット、およびプロファイル内でオブジェクトが更新されます。

以下の図に、エディタ内で開いているカスタムデータオブジェクト例を示します。

Overview

General

Name: DistinctReportDefinitions
Description:

DistinctReportDefinitions

Name	Native Type
ReportDefini...	
NAME	varchar2
DESCRIPTION	varchar2
POSITION	number(p,s)

Columns

Candidate columns:

	Name	Native Type	Precision	Scale	Visibility	Description
	ReportDefinition					
1	NAME	varchar2	80	0	Read and...	
2	DESCRIPTION	varchar2	80	0	Read and...	
3	POSITION	number(p,s)	10	0	Read and...	

When column metadata changes:
☒ Synchronize input and output
☐ Do not synchronize

Overview
Read
Write
Parameters
Advanced

カスタマイズデータオブジェクトは、以下のタスクを実行するために作成します。

- カスタムクエリを作成し、データ統合サービスでソースデータを読み込むためのデフォルトクエリと置き換える。デフォルトクエリは、データ統合サービスがソースから読み込む各カラムを参照する SELECT 文です。
- データオブジェクトのパラメータを定義する。接続を表すカスタマイズデータオブジェクトに、パラメータを定義して割り当てることができます。接続名、テーブル所有者、およびテーブル名のパラメータを定義できます。カスタマイズデータオブジェクトを使用するマッピングを実行するとき、ランタイムの接続パラメータに対して個別に値を定義することができます。

- 同じソースデータベースから生成されたソースデータを結合する。データベース内にリレーションが存在するかどうかにかかわらず、プライマリキー-外部キーのリレーションで複数のテーブルを結合できます。
- オブジェクトをソースと同期するときにキーリレーションを保持する。複数のテーブルを含むカスタマイズデータオブジェクトを作成し、データベースに存在しないキーリレーションを定義する場合、データオブジェクトを同期するときにリレーションを保持することができます。
- ソースから個別の値を選択する。[個別に選択]を選択すると、データ統合サービスは SELECT DISTINCT 文をデフォルトの SQL クエリに追加します。
- データ統合サービスがソースデータを読み込むときに行をフィルタする。フィルタ条件を含めると、データ統合サービスはデフォルトクエリに WHERE 句を追加します。
- ソート済みポートを指定する。ソート済みポートに対して数値を指定すると、データ統合サービスは ORDER BY 句をデフォルトの SQL クエリに追加します。
- デフォルトのインナージョインの代わりにアウタージョインを指定する。ユーザ定義ジョインを含めると、データ統合サービスは SQL クエリのメタデータで指定された結合情報を置き換えます。
- マッピング実行前およびマッピング実行後 SQL コマンドを追加する。データ統合サービスは、ソースデータベースに対してマッピング実行前の SQL コマンドを実行した後、ソースを読み込みます。また、ターゲットへ書き込みをした後にソースデータベースに対してマッピング実行後 SQL コマンドを実行します。

以下のタイプの接続およびオブジェクトからカスタマイズデータオブジェクトを作成できます。

- DB2 i5/OS 接続
- DB2 z/OS 接続
- IBM DB2 接続
- JDBC 接続
- Microsoft SQL Server 接続
- ODBC 接続
- Oracle 接続
- リレーショナルデータオブジェクト

カスタム SQL クエリを介して、ソースをカスタマイズデータオブジェクトに追加できます。

キーのリレーション

ソースがリレーショナルリソースの場合、カスタマイズデータオブジェクトのソース間にキーリレーションを作成できます。キーリレーションを使用して、カスタマイズデータオブジェクト内のソースを結合することができます。

注: カスタマイズデータオブジェクトがソースとしてリレーショナルデータオブジェクトを使用する場合、カスタマイズデータオブジェクトにキーリレーションを作成することはできません。代わりに、リレーショナルデータオブジェクト間にキーリレーションを作成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をカスタマイズデータオブジェクトにインポートする場合、データベースに定義されているプライマリキー情報は保持されます。関連するリレーショナルリソース同士をカスタマイズデータオブジェクトに同時にインポートする場合、キーリレーション情報も保持されます。ただし、関連するリレーショナルデータリソースを個別にインポートする場合は、オブジェクトをカスタマイズデータオブジェクトにインポートした後にキーリレーションを再作成する必要があります。

カスタマイズデータオブジェクトのソース間にキーリレーションが存在する場合、Data Integration Service は、各ソースの関連キーに基づいてテーブルを結合します。このデフォルトジョインは、WHERE 句に次の構文を使用するインナー等価ジョインです。

Source1.column_name = Source2.column_name

ユーザー定義ジョインを入力するか、またはカスタムクエリを作成することで、デフォルトジョインをオーバーライドできます

カスタマイズデータオブジェクトにキーリレーションを作成するには、最初に、参照先ソーストランスフォーメーションにプライマリキーを作成します。次に、外部キーを含むソーストランスフォーメーションにリレーションを作成します。

作成したキーリレーションは、カスタマイズデータオブジェクトのメタデータに存在します。ソースのリレシヨナルリソースを変更する必要はありません。

カスタマイズデータオブジェクトの書き込みプロパティ

データ統合サービスは、データをリレシヨナルリソースに書き込むときに書き込みプロパティを使用します。書き込みプロパティを編集するには、**【書き込み】** ビューで入力トランスフォーメーションを選択し、**【詳細】** プロパティを選択します。

以下の表に、カスタマイズデータオブジェクトに対して設定する書き込みプロパティを示します。

プロパティ	説明
Hive ターゲットパーティションを切り詰めます	データが挿入されている Hive ターゲットのパーティションを上書きします。このオプションを有効にするには、ターゲットテーブルを切り詰めるオプションも選択する必要があります。 外部のパーティション化されたテーブルおよび非パーティション化されたテーブルを切り詰めることができます。 デフォルトでは無効になっています。
ロードタイプ	ターゲットロードのタイプ。[ノーマル] または [一括] を選択します。 [ノーマル] を選択した場合、データ統合サービスはターゲットを通常どおりにロードします。DB2、Sybase、Oracle、Microsoft SQL Server にロードする場合には、[一括] を選択できます。他のデータベースタイプに [一括] を指定すると、データ統合サービスがノーマルロードに戻します。バルクロードによってマッピングのパフォーマンスは向上しますが、データベースロギングが発生しないので、リカバリ機能が制限されます。バルクロードを使用して Oracle ターゲットに書き込むときに、Oracle データベースの制約を無効化することでパフォーマンスを最適化できます。 マッピングにアップデイトストラテジトランスフォーメーションが含まれる場合には、[ノーマル] モードを選択してください。[ノーマル] を選択したときに Microsoft SQL Server ターゲット名にスペースが含まれる場合は、接続オブジェクトで以下の環境 SQL を設定します。 <code>SET QUOTED_IDENTIFIER ON</code>
更新オーバーライド	ターゲットのデフォルトの UPDATE 文をオーバーライドします。
削除	削除のフラグが設定された行をすべて削除します。 デフォルトは [有効] です。
挿入	挿入のフラグが設定された行をすべて挿入します。 デフォルトは [有効] です。

プロパティ	説明
ターゲットスキーマストラテジ	リレーショナルまたは Hive ターゲットテーブルのターゲットスキーマストラテジのタイプ。 次のいずれかのターゲットスキーマストラテジを選択します。 <ul style="list-style-type: none"> - 維持-既存のターゲットスキーマを維持。データ統合サービスは、既存のターゲットスキーマを維持します。 - 作成-実行時にテーブルを作成または置換。データ統合サービスは、実行時にターゲットテーブルを削除して、指定したターゲットテーブルに基づくテーブルで置換します。 - パラメータの割り当て。ターゲットスキーマストラテジの値を表すパラメータを割り当てておき、実行時にそのパラメータを変更できます。
作成または置換の DDL クエリ	データ統合サービスがターゲットテーブルを作成または置換する際にに基づく DDL クエリ。 このオプションは【作成-実行時にテーブルを作成または置換】ターゲットスキーマストラテジを選択したときに適用されます。
ターゲットテーブルの切り詰め	データをロードする前にターゲットを切り詰めます。 デフォルトでは無効になっています。
アップデートストラテジ	既存の行のアップデートストラテジ。次のいずれかのストラテジを選択します。 <ul style="list-style-type: none"> - 更新時に更新。データ統合サービスは更新のフラグが設定された行をすべて更新します。 - 挿入時に更新。データ統合サービスは更新のフラグが設定された行をすべて挿入します。ターゲットの【挿入】オプションも選択する必要があります。 - 更新しない場合は挿入。更新のフラグが設定された行がターゲット内に存在している場合、データ統合サービスは、これらの行を更新した後、挿入のマークが付いている残りの行を挿入します。ターゲットの【挿入】オプションも選択する必要があります。
PreSQL	ソースを読み込む前にターゲットデータベースに対してデータ統合サービスが実行する SQL コマンド。Developer tool では、SQL は検証されません。
PostSQL	ソースを読み込んだ後にターゲットデータベースに対してデータ統合サービスが実行する SQL コマンド。Developer tool では、SQL は検証されません。

カスタマイズデータオブジェクトの作成

マッピング、マップレット、またはプロファイルに追加するカスタマイズデータオブジェクトを作成します。カスタマイズデータオブジェクトを作成したら、ソースに追加します。

1. **【Object Explorer】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】 > 【新規】 > 【データオブジェクト】** をクリックします。
【新規】 ダイアログボックスが表示されます。
3. **【リレーショナルデータオブジェクト】** を選択して、**【次へ】** をクリックします。
【新しいリレーショナルデータオブジェクト】 ダイアログボックスが表示されます。
4. **【接続】** オプションの横にある **【参照】** をクリックして、データベースへの接続を選択します。
5. **【カスタマイズデータオブジェクトの作成】** をクリックします。
6. カスタマイズデータオブジェクトの名前を入力します。
7. **【場所】** オプションの横にある **【参照】** をクリックして、カスタマイズデータオブジェクトを作成するプロジェクトを選択します。

8. **【完了】** をクリックします。

カスタマイズデータオブジェクトが、**【Object Explorer】** ビューのプロジェクトまたはフォルダーに表示されます。

カスタマイズデータオブジェクトにソースを追加します。リレーショナルリソースまたはリレーショナルデータオブジェクトを、ソースとして追加できます。カスタム SQL クエリを使用してソースを追加することもできます。

カスタマイズデータオブジェクトへのリレーショナルリソースの追加

カスタマイズデータオブジェクトを作成したら、ソースに追加します。リレーショナルリソースをソースとして使用できます。

リレーショナルリソースをカスタマイズデータオブジェクトに追加する前に、データベースへの接続を設定します。

1. **【Object Explorer】** ビューで、同一のリレーショナル接続でリレーショナルリソースを 1 つ以上選択します。
2. **【接続エクスプローラ】** ビューを右クリックして、**【プロジェクトに追加】** を選択します。
【プロジェクトに追加】 ダイアログボックスが表示されます。
3. **【既存のカスタマイズされたデータオブジェクトに関連リソースとして追加】** を選択して、**【OK】** をクリックします。
【データオブジェクトに追加】 ダイアログボックスが表示されます。
4. カスタマイズデータオブジェクトを選択し、**【OK】** をクリックします。
5. 複数のリソースをカスタマイズデータオブジェクトに追加する場合、書き込み先のリソースを選択するように求められます。リソースを選択して **【OK】** をクリックします。

マッピングのカスタマイズ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使用すると、Developer ツールはデータをこのリソースに書き込みます。

リソースがカスタマイズデータオブジェクトに追加されます。

カスタマイズデータオブジェクトへのリレーショナルデータオブジェクトの追加

カスタマイズデータオブジェクトを作成したら、ソースに追加します。リレーショナルデータオブジェクトをソースとして使用できます。

1. カスタマイズデータオブジェクトを開きます。
2. **【読み取り】** ビューを選択します。
3. **【Object Explorer】** ビューで、1 つのリレーショナル接続からリレーショナルデータオブジェクトを 1 つ以上選択します。
4. オブジェクトを **【Object Explorer】** ビューからカスタマイズデータオブジェクトの **【読み取り】** ビューにドラッグします。
5. 複数のリレーショナルデータオブジェクトをカスタマイズデータオブジェクトに追加する場合、書き込み先のオブジェクトを選択するように求められます。オブジェクトを選択して **【OK】** をクリックします。
マッピングのカスタマイズ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使用すると、Developer ツールはデータをこのリレーショナルデータオブジェクトに書き込みます。

リレーショナルデータオブジェクトがカスタマイズデータオブジェクトに追加されます。

カスタマイズデータオブジェクトのキーの作成

ソーストランスフォーメーションの各行を識別するためのキーカラムを作成します。各ソーストランスフォーメーションごとに1つのプライマリキーを作成できます。

1. カスタマイズデータオブジェクトを開きます。
2. **【読み取り】** ビューを選択します。
3. キーを作成するソーストランスフォーメーションを選択します。
ソースは、リレーショナルデータオブジェクトではなくリレーショナルリソースにする必要があります。ソースがリレーショナルデータオブジェクトの場合、リレーショナルデータオブジェクトにキーを作成する必要があります。
4. **【キー】** プロパティを選択します。
5. **【追加】** をクリックします。
【新しいキー】 ダイアログボックスが表示されます。
6. キー名を入力します。
7. キーがプライマリキーの場合、**【プライマリキー】** を選択します。
8. キーカラムを選択します。
9. **【OK】** をクリックします。
10. カスタマイズデータオブジェクトを選択します。

カスタマイズデータオブジェクト内のリレーシヨンの作成

カスタマイズデータオブジェクトにソース間のキーリレーシヨンを作成できます。

参照するソース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す。

1. カスタマイズデータオブジェクトを開きます。
2. **【読み取り】** ビューを選択します。
3. 外部キーを作成するソーストランスフォーメーションを選択します。
ソースは、リレーショナルデータオブジェクトではなくリレーショナルリソースにする必要があります。ソースがリレーショナルデータオブジェクトの場合、リレーショナルデータオブジェクトにリレーシヨンを作成する必要があります。
4. **【リレーシヨン】** プロパティを選択します。
5. **【追加】** をクリックします。
【新しいリレーシヨン】 ダイアログボックスが表示されます。
6. 外部キーの名前を入力します。
7. 参照先のソーストランスフォーメーションからプライマリキーを選択します。
8. **【OK】** をクリックします。
9. **【リレーシヨン】** プロパティで、外部キーカラムを選択します。
10. カスタマイズデータオブジェクトを選択します。

ターゲットテーブルの作成または置換

Developer tool では、モデルリポジトリで 1 つ以上のリレーショナルデータオブジェクトの DDL スクリプトを生成し、DDL スクリプトを実行してターゲットデータベースでテーブルの作成または置換ができます。データベースにすでにターゲットが存在する場合、そのターゲットを削除して再作成できます。

設計時または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。設計時にターゲットを作成または置換するには、マッピング実行前に DDL スクリプトを作成して実行する必要があります。実行時にリレーショナルターゲットを作成または置換するように、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スキーマ戦略を設定できます。

Developer tool ですべてのサポートされた接続タイプに対応したデータベース固有バージョンの DDL スクリプトが生成されます。JDBC または ODBC ターゲットを選択すると、Developer tool は ANSI SQL-92 generic データ型形式を生成します。次のデータベースタイプの DDL スクリプトを生成できます。

- IBM DB2
- Greenplum
- Hive
- JDBC
- Microsoft SQL Server
- Netezza
- ODBC
- Oracle
- Teradata

注: OLE DB SQL Server 接続を使用する場合は、DDL スクリプトを生成できません。

ターゲットテーブル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

DDL スクリプトを生成および実行する場合は、以下のルールとガイドラインを考慮します。

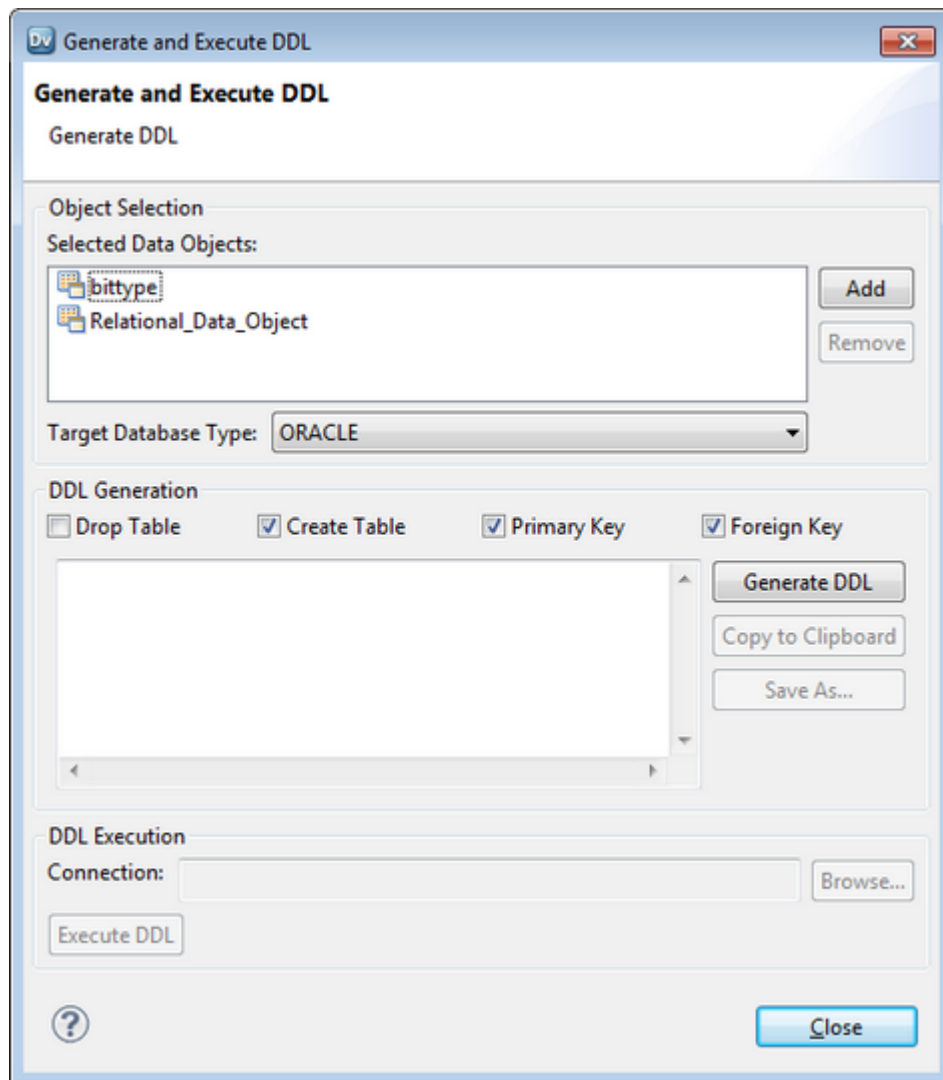
- 同じソーステーブル名を持つ複数のオブジェクトを選択しないようにします。同じソーステーブル名を持つ複数のオブジェクトを選択すると、DDL コードが失敗します。同じソーステーブル名を持つ 3 つのオブジェクトに [テーブルの削除] オプションおよび [テーブルの作成] オプションを選択した場合、DDL コードは最初の [テーブルの削除] コマンドおよび [テーブルの作成] コマンドでは成功しますが、後続のコマンドでは失敗します。
- マッピングを実行する前に、データ型を確認して必要な場合はそれらを手動で更新します。DDL スクリプトを生成するときに、Oracle の char および byte のセマンティクスは無視されます。Oracle データベースの char カラムおよび varchar2 カラムを含むテーブルを作成する場合は、char および byte セマンティクスのためのストレージを定義できます。Oracle テーブルのメタデータを Developer tool にインポートする場合、char および byte セマンティクスは無視されます。Oracle テーブルの DDL スクリプトを生成する場合、Developer tool はデータ型を Char として定義します。
- ODBC 接続の DDL を生成する場合、Developer tool は DDL スクリプトを ANSI SQL-92 の汎用データ型形式で作成します。ANSI SQL-92 形式は、ターゲットデータベースでデータ型またはデータ長がサポートされないことがあるため、一部のデータベースでは実行されない場合があります。
- Greenplum から Netezza にデータベーステーブルを移行するために DDL スクリプトを作成する場合、Netezza データベースの NVARCHAR データ型では最大 16000 文字のみがサポートされるため、NVARCHAR カラムに含めることができるのは 16000 文字のみです。

- DDL スクリプトを生成する際に、Developer tool は元の接続および新しい接続が参照するデータベース間のデータ型の最適な一致を特定します。データ型の精度およびスケールはデータベースによって異なります。Oracle データベースでは、Timestamp データ型のデフォルトの精度およびスケールは(29, 9)です。Oracle から Microsoft SQL Server への DDL スクリプトを生成する場合、Timestamp データ型の精度およびスケールは(26, 6)に減少します。Oracle から DB2 への DDL スクリプトを生成する場合、Timestamp データ型の精度およびスケールは(27, 7)に減少します。

設計時の DDL の生成と実行

DDL スクリプトを生成および実行する前に、ユーザーにターゲットデータベースにアクセスするための適切な読み取り権限および書き込み権限があることを確認します。

- Developer tool の【オブジェクトエクスプローラ】ビューで、ターゲットデータベースにテーブルを作成するリレーショナルデータオブジェクトを選択します。複数のテーブルを作成する場合は、**Shift** キーと **Ctrl** キーを押しながら、複数のデータオブジェクトを選択します。
- 選択したデータオブジェクトを右クリックして、**[DDL の生成と実行]** を選択します。
[DDL の生成と実行] ダイアログボックスが表示されます。



3. 【オブジェクトの選択】領域で、DDL を生成する【ターゲットデータベースタイプ】を選択します。

4. [DDL の生成] 領域では、次のオプションを選択できます。
 - **テーブルの削除。** データベースにテーブルを作成する前にテーブルを削除します。このオプションは、テーブルを置き換える場合に選択します。
 - **テーブルの作成。** ターゲットデータベースにテーブルを作成します。
 - **プライマリキー。** 選択したデータオブジェクトに基づいてプライマリキーを作成します。
 - **外部キー。** 選択したデータオブジェクトに基づいて外部キーを作成します。
 - **クリップボードにコピー。** DDL スクリプトをクリップボードにコピーします。
 - **名前を付けて保存。** DDL スクリプトをファイルに保存します。
5. [DDL の生成] 領域で、**[DDL の生成]** をクリックします。

選択したデータオブジェクトのために生成された DDL が [DDL の生成] 領域に表示されます。

警告: [DDL の生成] 領域に生成された DDL スクリプトを編集した場合、その DDL スクリプトを実行したときにエラーが発生することがあります。
6. [DDL の実行] 領域で、**[参照]** をクリックしてターゲットデータベースの接続を選択します。

[接続の選択] ダイアログボックスには、[オブジェクトの選択] 領域で選択したターゲットデータベースタイプに基づいてターゲットデータベースの接続が一覧表示されます。例えば、ターゲットデータベースタイプとして Oracle を選択した場合、Developer tool は Oracle の接続を表示します。
7. **[DDL の実行]** をクリックします。
8. **[閉じる]** をクリックします。

実行時での DDL の作成と実行

実行時にリレーショナルターゲットまたは Hive ターゲットを作成または置換するように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スキーマ戦略を選択するときに、DDL クエリを定義できます。

データ統合サービスは、ユーザーが定義した 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と Hive ターゲットテーブルを作成または置換します。

テーブルをカスタマイズする DDL クエリを定義するか、データ統合サービスがターゲットテーブルを作成または置換するときにベースとする必要があるパーティションなど追加パラメータを指定できます。テーブルには、DDL クエリで定義するカラムが含まれます。DDL クエリにはプレースホルダも入力できます。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。

詳細については、『*Informatica Developer マッピングガイド*』を参照してください。

DDL の生成エラー

DDL スクリプトを生成して実行するときに、エラーが発生する場合があります。エラーの原因として以下のいずれかが考えられます。

- 選択したデータ型をターゲットデータベースがサポートしていない。
- 選択した物理データオブジェクトに循環依存関係が含まれる。
- 選択した物理データオブジェクトにカラムが含まれない。

カスタムクエリ

カスタム SQL クエリは、カスタマイズデータオブジェクト内のデフォルトの SQL クエリを上書きする SELECT 文です。

カスタムクエリは、データ統合サービスがリレーショナルソースからデータを読み取るために使用するデフォルトの SQL クエリをオーバーライドします。カスタムクエリは、ソースフィルタの入力、ソート済みポートの使用、ユーザー定義ジョインの入力、または異なるポートの選択時に定義する単純クエリの設定もオーバーライドします。

データベース言語では有効であるがトランスフォーメーション言語では使用できない SQL 操作を実行するためのカスタムクエリを作成できます。カスタマイズデータオブジェクトにカスタムクエリを定義する場合、複数のマッピングまたはプロファイルでオブジェクトを再利用することができます。

カスタマイズデータオブジェクトにカスタムクエリを作成する場合のガイドラインは以下のとおりです。

- SELECT 文では、ソーストランスフォーメーションに表示される順序でカラム名を記述する必要があります。
- データベース予約語はすべて引用符で囲んでください。
- ドル記号 (\$) の前にエスケープ文字を追加します。\$ の前にスラッシュ (\) 文字がある場合、エスケープ文字 (\) をスラッシュとドル記号の両方に追加します。例えば、\$ を \\$ と入力し、\\$ を \\\$ と入力します。

カスタマイズデータオブジェクトを使用して自己結合を実行する場合、自己結合を含むカスタム SQL クエリを入力する必要があります。カスタムクエリが設定されたカスタマイズデータオブジェクトを、マッピングのソースとして使用できます。ソースデータベースは、データをデータ統合サービスに渡す前にクエリを実行します。ソースを空のカスタマイズデータオブジェクトに追加するカスタムクエリを作成できます。カスタムクエリを使用して、デフォルトの SQL クエリをオーバーライドすることもできます。

カスタムクエリの最適化

データ統合サービスは、パフォーマンスを向上させるために、カスタムクエリをプッシュしてリレーショナルデータオブジェクトで実行できます。クエリがデータベースに対して有効なサブクエリを形成する場合は、カスタムクエリをプッシュするようにしてください。

カスタムクエリを使用してリレーショナルデータオブジェクトのデータを読み取ると、データ統合サービスはデータベース内でクエリを実行することによりクエリを最適化できます。クエリがデータベースに対して有効なサブクエリを形成する場合、データ統合サービスはカスタムクエリをプッシュできます。カスタムクエリの SQL 構文がデータベースに対するサブクエリとして有効ではない場合、結果のクエリは実行に失敗します。

IBM DB2 以外のリレーショナルデータベースに対してカスタムクエリをプッシュする場合は、カラム参照ではない選択リストでそれぞれの式に対するエイリアスを指定する必要があります。これらのエイリアスにより、データ統合サービスが選択リスト内の式を参照できるようになります。

エイリアスとサブクエリに対する有効な SQL 構文に関する詳細は、データベースのマニュアルを参照してください。

デフォルトクエリ

Data Integration Service は、リレーショナルソースからデータを読み取るデフォルトの SQL クエリを生成します。カスタマイズ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インスタンスでデフォルトクエリをオーバーライドすることができます。

単純クエリまたは詳細クエリでデフォルトクエリをオーバーライドすることができます。単純クエリを使用して、個別の値の選択、ソースフィルタの入力、ポートのソート、またはユーザー定義ジョインの入力を行うことができます。ソースからデータを読み取るためのカスタム SQL クエリを作成するには、詳細クエリを使用します。カスタムクエリはデフォルトクエリおよび単純クエリをオーバーライドします。

テーブル名またはカラム名にデータベース予約語が含まれる場合、予約語ファイル reswords.txt を作成および管理できます。reswords.txt ファイルは、Data Integration Service がアクセスできる任意のマシンに作成します。

Data Integration Service は、マッピングの実行時に reswords.txt ファイルを検索します。ファイルが存在する場合、Data Integration Service はデータベースに対して SQL 文を実行する際に、一致した予約語を引用符で囲みます。デフォルトクエリをオーバーライドする場合、データベース予約語はすべて引用符で囲む必要があります。

Data Integration Service は、デフォルトクエリを生成するときに以下の文字を含むテーブル名およびフィールド名を二重引用符で区切ります。

/ + - = ~ ` ! % ^ & * () [] { } ' ; ? , < > \ | <space>

予約語ファイルの作成

カスタマイズデータオブジェクトにデータベース予約語が含まれる場合、予約語ファイルを作成します。

予約語ファイルを使用するように Data Integration Service を設定するには、管理者特権が必要です。

1. 「reswords.txt」というファイルを作成します。
2. 角括弧で囲んでデータベース名を入力して（例 [Oracle]）、各データベースのセクションを作成します。
3. 予約語をデータベース名の下に追加します。

以下に例を示します。

```
[Oracle]
OPTION
START
where
number
[SQL Server]
CURRENT
where
number
```

エントリでは、大文字と小文字は区別されません。

4. reswords.txt ファイルを保存します。
5. Informatica Administrator で、[Data Integration Service] を選択します。
6. カスタムプロパティを編集します。
7. 以下のカスタムプロパティを追加します。

名前	値
予約語ファイル	<path>\reswords.txt

8. Data Integration Service をリスタートします。

ヒント

データベースオブティマイザに対する指示として、ソースの SQL クエリにヒントを追加することができます。オブティマイザはヒントを使用して、ソースにアクセスするためのクエリ実行プランを選択します。

[ヒント] フィールドは、リレーショナルデータオブジェクトインスタンスまたはカスタマイズデータオブジェクトの【クエリ】ビューに表示されます。ソースデータベースは、Oracle、Sybase、IBM DB2、または Microsoft SQL Server でなければなりません。それ以外のデータベースタイプでは、[ヒント] フィールドは表示されません。

Data Integration Service によるソースクエリの生成時に、Developer ツールで入力した SQL ヒントがクエリに追加されます。ヒントは Data Integration Service では解析されません。ソースを含むマッピングを実行したときに、マッピングログにクエリとヒントが表示されます。

Data Integration Service によって SQL ヒントが挿入されるクエリ内の位置は、データベースタイプによって異なります。ヒントの構文については、データベースのマニュアルを参照してください。

Oracle

ヒントが SELECT/UPDATE/INSERT/DELETE キーワードのすぐ後に追加されます。

```
SELECT /*+ <hints> */ FROM ...
```

'+'はヒントの先頭を示します。

ヒントはコメント（/* ... */または--...行の末尾まで）に格納されます。

Sybase

ヒントがクエリのすぐ後に追加されます。ヒントでプラン名を設定します。

```
SELECT ... PLAN <plan>
```

```
select avg(price) from titles plan "(scalar_agg (i_scan type_price_ix titles ))"
```

IBM DB2

ヒントとして optimize-for 句を入力できます。この句はクエリの末尾に追加されます。

```
SELECT ... OPTIMIZE FOR <n> ROWS
```

optimize-for 句では、クエリで処理する行数をデータベースオプティマイザに示します。これは行数の制限ではありません。データベースで処理する行数が<n>行を超えると、パフォーマンスが低下する可能性があります。

Microsoft SQL Server

ヒントがクエリの末尾に OPTION 句の一部として追加されます。

```
SELECT ... OPTION ( <query_hints> )
```

ヒントに関するルールとガイドライン

SQL クエリのヒントを設定するときは、次のルールとガイドラインに従います。

- プッシュダウンの最適化を有効にする場合やリレーショナルデータオブジェクトで準結合を使用する場合は、元のソースクエリが変わります。変更後のクエリにはヒントは適用されません。
- ヒントはジョインやフィルタのオーバーライドと組み合わせることができますが、SQL のオーバーライドを設定した場合は SQL のオーバーライドが優先され、他のオーバーライドは適用されません。
- **【クエリ】** ビューには、簡易ビューと詳細ビューがあります。簡易ビューでフィルタ、ソート、またはジョインのオーバーライドを使用してヒントを入力した場合、完全なクエリオーバーライドは詳細ビューに表示されます。

ヒントの作成

データベースオプティマイザにクエリプランの決定に関する指示を伝えるには、ヒントを作成します。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。
3. **【出力トランスフォーメーション】** を選択します。

4. **【クエリ】** プロパティを選択します。
5. 単純クエリを選択します。
6. **【ヒント】** フィールドの横にある **【編集】** をクリックします。
【ヒント】 ダイアログボックスが表示されます。
7. **【SQL クエリ】** フィールドにヒントを入力します。
Developer ツールでは、ヒントは検証されません。
8. **【OK】** をクリックします。
9. データオブジェクトを保存します。

個別に選択

カスタマイズ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インスタンスのソースから、個別選択オプションを使用して一意の値を選択できます。個別に選択を有効にすると、SELECT DISTINCT 文がデフォルトの SQL クエリに追加されます。

個別選択オプションを使用して、ソースデータをフィルタします。例えば、**【個別に選択】** オプションを使用して、総売上高の一覧のテーブルから一意の顧客 ID を抽出できます。マッピングでリレーショナルデータオブジェクトを使用する場合、データフローの早い段階でデータがフィルタされるため、パフォーマンスが向上します。

個別に選択の使用

【個別選択】 プロパティを使用して、リレーショナルソースから一意の値を選択します。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。
3. **【出力トランスフォーメーション】** を選択します。
4. **【クエリ】** プロパティを選択します。
5. 単純クエリを選択します。
6. **【個別に選択】** オプションを選択します。
7. カスタマイズデータオブジェクトを選択します。

フィルタ

カスタムクエリにフィルタ値を入力できます。フィルタはクエリ SELECT 文の WHERE 句に変換されます。Data Integration Service がソースのテーブルから読み取る行数を減らすにはフィルタを使用します。

ソースフィルタの入力

Data Integration Service がリレーショナルリソースから読み取る行数を減らすには、ソースフィルタを入力します。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。
3. **【出力トランスフォーメーション】** を選択します。
4. **【クエリ】** プロパティを選択します。

5. 単純クエリを選択します。
6. **【フィルタ】** フィールドの横にある**【編集】** ボタンをクリックします。
【SQL クエリ】 ダイアログボックスが表示されます。
7. **【SQL クエリ】** フィールドにフィルタ条件を入力します。
【カラム】 リストからカラムを選択することもできます。
8. **【OK】** をクリックします。
9. **【検証】** をクリックしてフィルタ条件を検証します。
10. データオブジェクトを保存します。

ソート済みポート

カスタマイズ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インスタンスのデフォルトクエリで行をソートできます。ソート基準のポートを選択します。Data Integration Service はそのポートをデフォルトクエリの ORDER BY 句に追加します。

マッピングに以下のトランスフォーメーションを含めるときに、ソース行をソートしてパフォーマンスを向上させることができます。

- アグリゲータ。アグリゲータトランスフォーメーションをソート済み入力用に設定すると、ソート済みポートを使ってソート済みデータを送ることができます。アグリゲータトランスフォーメーションの Group By ポートは、カスタマイズデータオブジェクトのソート済みポートの順序と一致する必要があります。
- ジョイナ。ジョイナトランスフォーメーションをソート済み入力用に設定すると、ソート済みポートを使ってソート済みデータを送ることができます。各カスタマイズデータオブジェクト内のソート済みポートの順序は同一に設定してください。

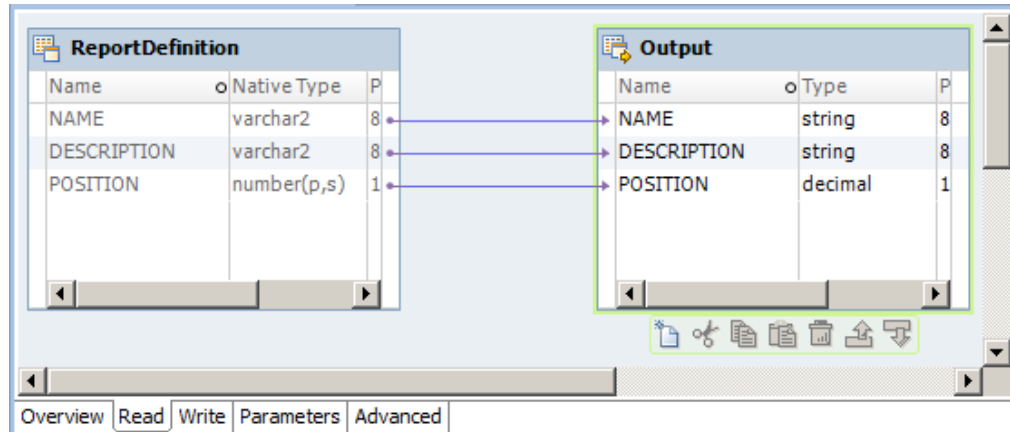
注: また、アグリゲータおよびジョイナトランスフォーメーションの前にソータートランスフォーメーションを使用して、リレーショナルデータおよびフラットファイルデータをソートすることもできます。

カラムデータのソート

ソート済みポートを使用して、カスタマイズ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インスタンスのカラムデータをソートします。このデータオブジェクトをマッピングまたはマップレットで読み取りトランスフォーメーションとして使用する場合、ソート済みデータを、読み取りトランスフォーメーションからトランスフォーメーションダウンストリームに送信できます。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。

以下の図に、エディタ内で開いているカスタマイズデータオブジェクトの【読み取り】ビューを示します。



3. 出力トランスフォーメーションを選択します。
4. 【クエリ】プロパティを選択します。
5. 単純クエリを選択します。
6. 【編集】ボタン（【ソート】フィールドの横）をクリックします。
【ソート】ダイアログボックスが表示されます。
7. ソート済みポートとしてカラムを指定するには、【新規】ボタンをクリックします。
8. カラムとソートタイプ（昇順または降順）を選択します。
9. 手順 7～8 を繰り返して、ソートする他のカラムを選択します。
【ソート】ダイアログボックスに表示される順序に従って、カラムがソートされます。
10. 【OK】をクリックします。
【クエリ】プロパティに、【ソート】フィールドのソースカラムが表示されます。
11. 【検証】をクリックしてソート構文を検証します。
12. データオブジェクトを保存します。

ユーザー定義ジョイン条件

ユーザー定義ジョインをカスタマイズ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インスタンスで設定できます。ユーザー定義ジョインは、複数のソースのデータを同じデータオブジェクトに結合するための条件を定義します。

ユーザー定義ジョインをカスタマイズ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インスタンスに追加すると、マッピングの読み取りトランスフォーメーションとしてそのデータオブジェクトを使用できます。ソースデータベースは、データを Data Integration Service に渡す前にジョインを実行します。ソーステーブルのインデックスを作成すると、マッピングのパフォーマンスが向上します。

関連ソースからのデータを結合するにはユーザー定義ジョインを作成します。ユーザー定義ジョインは、各ソースの関連キーに基づいて作成されるデフォルトのインナージョインをオーバーライドします。ユーザー定義ジョインを入力する場合、結合条件を指定する WHERE 句の内容を入力します。ユーザー定義ジョインがアウトジョインを実行すると、データベース構文に応じて WHERE 句または FROM 句にジョイン構文が挿入されます。

以下の環境では、ユーザー定義ジョインの入力が必要になることがあります。

- カラムがプライマリキーと外部キーの関係を持たない。
- 結合に使用するカラムのデータ型が一致しない。

- アウタージョインのような、別の種類の結合を指定したい。

カスタマイズ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にユーザー定義ジョインを入力する場合のガイドラインは以下のとおりです。

- ユーザー定義ジョインには WHERE キーワードを含めないでください。
- データベース予約語はすべて引用符で囲んでください。
- Informatica ジョイン構文を使用し、**【SQL 内の引用符を有効にする】** が接続に対して有効になっている場合、テーブル名や接続名を手動で入力するときは引用符で囲む必要があります。ユーザー定義ジョインを入力するときにテーブルやカラムを選択すると、Developer ツールによってテーブル名やカラム名が引用符で囲まれます。

ユーザー定義ジョインは、データベースの関連リソースのデータを結合します。異種データソースを結合する場合、ソースからデータを読み取るマッピングでジョイナトランスフォーメーションを使用します。自己結合を実行するには、自己結合を含むカスタム SQL クエリを入力する必要があります。

ユーザー定義ジョインの入力

ユーザー定義ジョインをカスタマイズ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で設定し、データオブジェクトソースの結合条件を定義します。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。
3. **【出力トランスフォーメーション】** を選択します。
4. **【クエリ】** プロパティを選択します。
5. 単純クエリを選択します。
6. **【結合】** フィールドの横にある **【編集】** ボタンをクリックします。
【SQL クエリ】 ダイアログボックスが表示されます。
7. **【SQL クエリ】** フィールドにユーザー定義ジョインを入力します。
【カラム】 リストからカラムを選択することもできます。
8. **【OK】** をクリックします。
9. **【検証】** をクリックしてユーザー定義ジョインを検証します。
10. データオブジェクトを保存します。

アウタージョインのサポート

カスタマイズデータオブジェクトを使用して、同じデータベース内の 2 つのソースのアウタージョインを実行できます。Data Integration Service では、アウタージョイン実行時に、1 つ目のソースリソースからすべての行を返し、2 番目のソースリソースからジョイン条件に一致する行を返します。

2 つのリソースを結合し、一方のリソースのすべての行を返したい場合には、アウタージョインを使用します。たとえば、登録済み顧客テーブルを月ごとの購入テーブルと結合して登録済み顧客の行動を調べたい場合に、アウタージョインを使用することができます。登録済み顧客テーブルを月ごとの購入テーブルと結合し、前月に何も購入しなかった顧客も含めて、登録済み顧客テーブルのすべての行を返すことができます。Normal ジョインを実行した場合、Data Integration Service はその月に何か購入した登録済み顧客のみ、および登録済み顧客による購入のみを返します。

アウタージョインの場合、ジョイナトランスフォーメーションの Master Outer（マスターアウター）ジョインまたは Detail Outer（明細アウター）ジョインとして同じ結果を生成することができます。しかし、アウタージョインを使用するとデータフロー内の行数が削減され、パフォーマンスが向上します。

2種類のアウタージョインを入力できます。

- レフト。Data Integration Service は、ジョイン構文の左辺のリソースからすべての行を、また、両方のリソースからジョイン条件に一致する行を返します。
- ライト。Data Integration Service は、ジョイン構文の右辺のリソースからすべての行を、また、両方のリソースからジョイン条件に一致する行を返します。

注: デフォルトクエリをオーバーライドする場合、ネストされたクエリ文でアウタージョインを使用します。

アウタージョインはユーザー定義ジョインまたはカスタム SQL クエリに入力できます。

Informatica ジョイン構文

ジョイン構文を入力する場合、Informatica ジョイン構文またはデータベース固有のジョイン構文を使用します。Informatica ジョイン構文を使用すると、Data Integration Service はマッピング実行中に構文を変換してソースデータベースに渡します。

注: ジョイン条件については、必ずデータベース固有の構文を使用してください。

Informatica ジョイン構文を使用する場合、結合文全体を「{Informatica 構文}」のようにブレースで囲みます。データベース構文を使用する場合、ソースデータベースがサポートしている構文をブレースで囲まずに使用します。

Informatica ジョイン構文を使用する場合、テーブル名をカラム名の前に付けます。たとえば、REG_CUSTOMER テーブルの FIRST_NAME という名前のカラムの場合、ジョイン構文には「REG_CUSTOMER.FIRST_NAME」と入力します。また、テーブル名にエイリアスを使用している場合、Data Integration Service がエイリアスを確実に認識するように、Informatica ジョイン構文内でエイリアスを使用します。

1つのデータオブジェクトでレフトアウタージョインまたはライトアウタージョインを Normal ジョインと組み合わせることができます。レフトアウタージョインとライトアウタージョインを組み合わせることはできません。複数の Normal ジョイン、および複数のレフトアウタージョインを使用します。一部のデータベースでは、ライトアウタージョインの使用に制限があります。

ジョインを組み合わせる場合、Normal ジョインを最初に入力します。

Normal ジョイン構文

カスタマイズ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インスタンスの結合条件を使用し、Normal ジョインを作成できます。

アウタージョインを作成する場合、デフォルトジョインをオーバーライドする必要があります。したがって、Normal ジョインをジョインオーバーライドに含める必要があります。Normal ジョインをジョインオーバーライドに含める場合、アウタージョインの前に Normal ジョインを記述します。ジョインオーバーライドには複数の Normal ジョインを入力することができます。

Normal ジョインを作成するには、以下の構文を使用します。

```
{ source1 INNER JOIN source2 on join_condition }
```

以下の表に、ジョインオーバーライド内の Normal ジョインの構文を示します。

構文	説明
<i>source1</i>	ソースリソース名。Data Integration Service は、このリソースからジョイン条件に一致する行を返します。
<i>source2</i>	ソースリソース名。Data Integration Service は、このリソースからジョイン条件に一致する行を返します。
<i>join_condition</i>	ジョイン条件。ソースデータベースがサポートしている構文を使用してください。複数のジョイン条件を AND 演算子によって組み合わせることができます。

たとえば、登録済み顧客のデータを含む REG_CUSTOMER テーブルがあるとします。

CUST_ID	FIRST_NAME	LAST_NAME
00001	Marvin	Chi
00002	Dinah	Jones
00003	John	Bowden
00004	J.	Marks

毎月更新される PURCHASES テーブルに以下のデータが含まれているとします。

TRANSACTION_NO	CUST_ID	DATE	AMOUNT
06-2000-0001	00002	6/3/2000	55.79
06-2000-0002	00002	6/10/2000	104.45
06-2000-0003	00001	6/10/2000	255.56
06-2000-0004	00004	6/15/2000	534.95
06-2000-0005	00002	6/21/2000	98.65
06-2000-0006	NULL	6/23/2000	155.65
06-2000-0007	NULL	6/24/2000	325.45

6 月の各取り引きの顧客名を表示する行を返すには、次の構文を使用します。

```
{ REG_CUSTOMER INNER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

Data Integration Service は以下のデータを返します。

CUST_ID	DATE	AMOUNT	FIRST_NAME	LAST_NAME
00002	6/3/2000	55.79	Dinah	Jones
00002	6/10/2000	104.45	Dinah	Jones
00001	6/10/2000	255.56	Marvin	Chi
00004	6/15/2000	534.95	J.	Marks

CUST_ID	DATE	AMOUNT	FIRST_NAME	LAST_NAME
00002	6/21/2000	98.65	Dinah	Jones

Data Integration Service は、顧客 ID が一致する行を返します。6 月に何も購入していない顧客は含まれません。非登録顧客による購入も含まれません。

レフトアウタージョイン構文

ジョインオーバーライドでレフトアウタージョインを作成することができます。1 つのジョインオーバーライドに複数のレフトアウタージョインを入力することができます。他のジョインと共にレフトアウタージョインを使用する場合、文においては、レフトアウタージョインをすべてまとめて Normal ジョインの後に一覧表示します。

レフトアウタージョインを作成するには、以下の構文を使用します。

```
{ source1 LEFT OUTER JOIN source2 on join_condition }
```

以下の表に、ジョインオーバーライド内のレフトアウタージョインの構文を示します。

構文	説明
<i>source1</i>	ソースリソース名。レフトアウタージョインの場合、Data Integration Service はこのリソースのすべての行を返します。
<i>source2</i>	ソースリソース名。Data Integration Service は、このリソースからジョイン条件に一致する行を返します。
<i>join_condition</i>	ジョイン条件。ソースデータベースがサポートしている構文を使用してください。複数のジョイン条件を AND 演算子によって組み合わせることができます。

たとえば、[「Normal ジョイン構文」 \(ページ 82\)](#)と同じ REG_CUSTOMER および PURCHASES テーブルを使用した場合、以下のジョインオーバーライドによって、6 月に何か購入した顧客の数を求めることができます。

```
{ REG_CUSTOMER LEFT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

Data Integration Service は以下のデータを返します。

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT
00001	Marvin	Chi	6/10/2000	255.56
00002	Dinah	Jones	6/3/2000	55.79
00003	John	Bowden	NULL	NULL
00004	J.	Marks	6/15/2000	534.95
00002	Dinah	Jones	6/10/2000	104.45
00002	Dinah	Jones	6/21/2000	98.65

Data Integration Service は REG_CUSTOMERS テーブル内の登録済み顧客をすべて返し、6 月に何も購入しなかった顧客には NULL 値を使用します。非登録顧客による購入は含まれません。

複数のジョイン条件を使用して、6 月において 1 回の購入金額が\$100.00 を超える登録済み顧客の人数を求めることができます。

```
{REG_CUSTOMER LEFT OUTER JOIN PURCHASES on (REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID AND PURCHASES.AMOUNT > 100.00) }
```

Data Integration Service は以下のデータを返します。

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT
00001	Marvin	Chi	6/10/2000	255.56
00002	Dinah	Jones	6/10/2000	104.45
00003	John	Bowden	NULL	NULL
00004	J.	Marks	6/15/2000	534.95

同じ期間の返品に関する情報を組み込みたい場合、複数のレフトアウタージョインを使用することができます。たとえば、RETURNS テーブルには以下のデータが含まれているとします。

CUST_ID	CUST_ID	RETURN
00002	6/10/2000	55.79
00002	6/21/2000	104.45

6 月に何か購入および返品した顧客の人数を求めるために、2 つのレフトアウタージョインを使用します。

```
{ REG_CUSTOMER LEFT OUTER JOIN PURCHASES on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID LEFT OUTER JOIN RETURNS on REG_CUSTOMER.CUST_ID = PURCHASES.CUST_ID }
```

Data Integration Service は以下のデータを返します。

CUST_ID	FIRST_NAME	LAST_NAME	DATE	AMOUNT	RET_DATE	RETURN
00001	Marvin	Chi	6/10/2000	255.56	NULL	NULL
00002	Dinah	Jones	6/3/2000	55.79	NULL	NULL
00003	John	Bowden	NULL	NULL	NULL	NULL
00004	J.	Marks	6/15/2000	534.95	NULL	NULL
00002	Dinah	Jones	6/10/2000	104.45	NULL	NULL
00002	Dinah	Jones	6/21/2000	98.65	NULL	NULL
00002	Dinah	Jones	NULL	NULL	6/10/2000	55.79
00002	Dinah	Jones	NULL	NULL	6/21/2000	104.45

Data Integration Service は、欠落値には NULL を使用します。

ライトアウタージョイン構文

ジョインオーバーライドでライトアウタージョインを作成することができます。ライトアウタージョインは、ジョイン構文でリソースの順序を逆にした場合、レフトアウタージョインと同じ結果を返します。ジョインオ

オーバーライドには、ライトアウタージョインを 1 つだけ使用します。複数のライトアウタージョインを作成する場合、ソースリソースの順序を逆にして、結合タイプをレフトアウタージョインに変更してみてください。

他のジョインと共にライトアウタージョインを使用する場合、ジョインオーバーライドの最後にライトアウタージョインを入力します。

ライトアウタージョインを作成するには、以下の構文を使用します。

```
{ source1 RIGHT OUTER JOIN source2 on join_condition }
```

以下の表に、ジョインオーバーライドでのライトアウタージョインの構文を示します。

構文	説明
<i>source1</i>	ソースリソース名。Data Integration Service は、このリソースからジョイン条件に一致する行を返します。
<i>source2</i>	ソースリソース名。ライトアウタージョインの場合、Data Integration Service はこのリソースのすべての行を返します。
<i>join_condition</i>	ジョイン条件。ソースデータベースがサポートしている構文を使用してください。複数のジョイン条件を AND 演算子によって組み合わせることができます。

PreSQL コマンドおよび PostSQL コマンド

SQL コマンドをカスタマイズ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インスタンスで作成できます。SQL コマンドを使用して、insert、update、delete などの SQL 文を実行できます。Data Integration Service は、ソースのリレーショナルリソースに対して SQL コマンドを実行します。

マッピングを実行すると、データ統合サービスはソースデータベースに対して PreSQL コマンドを実行した後、ソースを読み込みます。ソースを更新してからマッピングでソースを使用する必要がある場合に、PreSQL コマンドを使用できます。例えば、ソース内のレコードを削除してからソースデータベースの最新のレコードを読み込むようにソースを更新する PreSQL コマンドを構成できます。マッピングを実行する際、マッピングが使用するソースに最新のレコードが含まれます。

データ統合サービスは、ターゲットに書き込んでからソースデータベースに対して PostSQL コマンドを実行することもできます。例えば、PostSQL コマンドを使用して、レコードがマッピングターゲットに読み込まれた後でソース内のレコードを削除することができます。マッピングがターゲットに書き込んでからレコードを削除するように PostSQL コマンドを構成できます。

PreSQL および PostSQL コマンドを構成する場合は次のガイドラインに従ってください。

- そのデータベースタイプで有効な任意のコマンドを使用します。Data Integration Service では、データベース側で許可されている場合も、ネストされたコメントは使用できません。
- 複数の文を区切るにはセミコロン (;) を使用します。Data Integration Service は各文の後にコミットを発行します。
- Data Integration Service は、/* ... */内のセミコロンは無視されます。
- コメントの外部でセミコロンを使用する必要がある場合は、バックスラッシュ (\) でセミコロンをエスケープします。セミコロンをエスケープすると、Data Integration Service はバックスラッシュを無視し、セミコロンを文の区切り文字として使用しません。
- Developer tool は PreSQL および PostSQL コマンド内の SQL を検証しません。

マッピング実行前/実行後 SQL コマンドの追加

マッピング実行前および実行後の SQL コマンドを、カスタマイズ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インスタンスに追加できます。マッピングでこのデータオブジェクトを使用すると、Data Integration Service で SQL コマンドが実行されます。

1. カスタマイズデータオブジェクトを開きます。
2. **【読み取り】** ビューを選択します。
3. 出力トランスフォーメーションを選択します。
4. **【詳細】** プロパティを選択します。
5. **【PreSQL】** フィールドにマッピング実行前 SQL コマンドを入力します。
6. **【PostSQL】** フィールドにマッピング実行後 SQL コマンドを入力します。
7. カスタマイズデータオブジェクトを選択します。

カスタムクエリの作成

カスタムクエリを作成し、ソースからデータを読み取るための特別な SELECT 文を発行します。カスタムクエリは、データ統合サービスがソースデータを読み取るためのデフォルトクエリをオーバーライドします。

1. カスタマイズ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インスタンスを開きます。
2. **【読み取り】** ビューを選択します。
3. 出力トランスフォーメーションを選択します。
4. **【クエリ】** プロパティを選択します。
5. 詳細クエリを選択します。
6. **【カスタムクエリーを使用】** を選択します。
データ統合サービスがソースデータを読み取るために発行するクエリが表示されます。
7. クエリを変更するか、カスタムクエリで置き換えます。
8. カスタムクエリをリレーショナルデータソースにプッシュする場合は、**【カスタムクエリをデータベースにプッシュ】** を選択します。
データ統合サービスは、デフォルトではカスタムクエリをデータベースにプッシュしません。クエリがデータベースに対して有効なサブクエリを形成する場合は、カスタムクエリをプッシュするようにしてください。
9. データオブジェクトを保存します。

非リレーショナルデータオブジェクト

非リレーショナルデータオブジェクトは、インポートしてマッピング、マップレット、またはプロファイルで使用します。非リレーショナルデータオブジェクトは、非リレーショナルデータソースを使用する物理データオブジェクトです。

非リレーショナルデータオブジェクトは、以下の接続タイプにインポートできます。

- Adabas
- IMS
- シーケンシャル

- VSAM

非リレーショナルデータオブジェクトをインポートする場合、Developer ツールは PowerExchange データマップからオブジェクトのメタデータを読み取ります。データマップは非リレーショナルレコードをリレーショナルテーブルにマッピングし、データへのアクセスに SQL 言語を使用できるようにします。データマップを作成するには、PowerExchange Navigator を使用します。

オブジェクトをインポートした後、そのオブジェクトの非リレーショナル操作を読み取り、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としてマッピングおよびマップレットに追加できます。各非リレーショナル操作は、データマップが定義するリレーショナルテーブルに対応します。1 つ以上の非リレーショナルレコードのフィールドからリレーショナルテーブルのカラムへのマッピングを表示するには、**【オブジェクトエクスプローラ】** ビューで非リレーショナル操作をダブルクリックします。

データマップの詳細については、『*PowerExchange Navigator Guide*』を参照してください。

注意:

- Informatica 9.0.1 で作成した非リレーショナルデータオブジェクトを使用する場合、操作を実行する前にアップグレードする必要があります。非リレーショナルデータオブジェクトをアップグレードするには、`infacmd pwx UpgradeModels` コマンドを実行します。
- 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み、データを非リレーショナルターゲットに書き込むマッピングを実行する場合、マッピングはエラーメッセージとともに失敗する可能性があります。メッセージは、ターゲットのデータマップにはキー定義があるにもかかわらず、ターゲットテーブルにキーがないことを示しています。
- シーケンシャル接続タイプの非リレーショナルデータオブジェクトの場合は、ファイル名とデフォルト接続をパラメータ化できます。

非リレーショナルデータオブジェクトのインポート

マッピング、マップレット、またはプロファイルで使用する 1 つ以上のリレーショナルデータオブジェクトをインポートします。

非リレーショナルデータオブジェクトをインポートする前に、データベースまたはデータセットへの接続を設定する必要があります。オブジェクトのデータマップを作成する必要もあります。

1. **【Object Explorer】** ビューで、プロジェクトまたはフォルダーを選択します。
2. **【ファイル】** > **【新規】** > **【データオブジェクト】** をクリックします。
3. **【非リレーショナルデータオブジェクト】** を選択して、**【次へ】** をクリックします。
【新しい非リレーショナルデータオブジェクト】 ダイアログボックスが表示されます。
4. オプションで、物理データオブジェクトの名前を入力します。
5. **【接続】** オプションの横にある **【参照】** をクリックして、接続を選択します。
6. **【データマップ】** オプションの横にある **【参照】** をクリックします。
【データマップの選択】 ダイアログボックスが表示されます。
7. 必要に応じて、フィルタ条件を入力します。
8. **【リソース】** 領域で、必要に応じて接続とスキーマを展開し、目的のデータマップを選択します。
9. **【OK】** をクリックして **【データマップの選択】** ダイアログボックスを閉じ、**【完了】** をクリックします。

非リレーショナルデータオブジェクトと対応する非リレーショナル操作が、**【Object Explorer】** ビューのプロジェクトまたはフォルダにある **【物理データオブジェクト】** の接続名の下に表示されます。

注: **【接続エクスプローラ】** ビューを使用して、非リレーショナルデータオブジェクトをインポートすることもできます。

非リレーショナルデータ操作からの読み取り、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作成

非リレーショナルデータ操作は、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として、マッピングまたはマプレットに追加できます。

1. 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を作成するマッピングまたはマプレットを開きます。
 2. **[Object Explorer]** ビューで、非リレーショナルデータオブジェクトを 1 つ以上選択します。
 3. 非リレーショナルデータ操作をマッピングエディタにドラッグします。
[マッピングに追加] ダイアログボックスが表示されます。
 4. **[読み取り]**、**[書き込み]**、または **[ルックアップ]** オプションを選択します。
[独立したデータオブジェクトとして] が自動的に選択されます。
 5. **[OK]** をクリックします。
- 非リレーショナルデータ操作ごとに、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が作成されます。

WSDL データオブジェクト

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

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

WSDL をインポートするには、以下のガイドラインを考慮してください。

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

WSDL データオブジェクトの概要ビュー

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

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

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

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

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

WSDL データオブジェクトの詳細ビュー

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

以下の表に、WSDL データオブジェクトの詳細プロパティを示します。

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

WSDL データオブジェクトのインポート

WSDL から Web サービスを作成する場合、または Web サービスコンシューマトランスフォーメーションを作成する場合は、WSDL データオブジェクトをインポートします。WSDL データオブジェクトは、WSDL ファイルから、または WSDL の場所を指す URI からインポートできます。WSDL データオブジェクトは、SOAP 1.1 バインディング操作と SOAP 1.2DL バインディング操作またはそのいずれかを含む WSDL ファイルからインポートできます。

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

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

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

4. WSDL の名前を入力します。
5. **【場所】** オプションの横にある **【参照】** をクリックして、WSDL データオブジェクトをインポートするプロジェクトまたはフォルダの場所を選択します。
6. **【次へ】** をクリックして、WSDL 内の操作を表示します。
7. **【完了】** をクリックします。

Object Explorer ビューで、プロジェクトまたはフォルダー内の **【物理データオブジェクト】** の下にデータオブジェクトが表示されます。

WSDL の同期

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

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

名前属性が変更されていない場合、WSDL コンポーネントを参照するオブジェクトが更新されます。たとえば、WSDL ファイルを編集して、単純要素 "CustID" の型を xs:string から xs:integer に変更します。WSDL データオブジェクトを同期すると、その CustID 要素を参照するすべての Web サービスおよび Web サービスコンシューマトランスフォーメーションの要素型が更新されます。

名前属性が変更された場合、その WSDL コンポーネントを参照するオブジェクトを開くと変更済みとしてマークされています。例えば、WSDL を編集して、要素の名前を「Resp」から「RespMsg」に変更します。それから、WSDL の同期を実行します。その要素を参照する Web サービスを開くと、その要素名はエディタでアスタリスク付きになり、Web サービスに変更が含まれることがわかります。Developer ツールは Web サービスの要素名を更新しますが、新しい要素をポートにマップする方法を特定できません。Resp 要素が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のポートにマップされている場合、RespMsg 要素を適切なポートにマップする必要があります。

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

WSDL データオブジェクトの同期

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

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

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

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

3. WSDL の名前と場所を確認します。

4. **【次へ】** をクリックして、WSDL 内の操作を表示します。

5. **【完了】** をクリックします。

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

証明書管理

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

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

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

Informatica Developer の証明書プロパティ

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

以下の表に、証明書のプロパティを示します。

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

Informatica Developer への証明書の追加

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

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

同期化

物理データオブジェクトのソースが変更されたときに、同期させることができます。物理データオブジェクトの同期化では、選択したソースからオブジェクトメタデータが再インポートされます。

すべての物理データオブジェクトを同期できます。リレーショナルデータオブジェクトまたはカスタマイズデータオブジェクトを同期する際、Developer ツールで定義したキーリレーションを保持するか上書きすることができます。

ソースの変更時に同期するようにカスタマイズデータオブジェクトを設定できます。例えば、カスタマイズデータオブジェクトでリレーショナルデータオブジェクトがソースとして使用されている場合に、そのリレーショナルデータオブジェクトにカラムを追加すると、Developer ツールはそのカラムをカスタマイズデータオブジェクトに追加します。ソースの変更時にカスタマイズデータオブジェクトを同期するには、カスタマイズデータオブジェクトの【概要】プロパティで【入力と出力を同期】オプションを選択します。

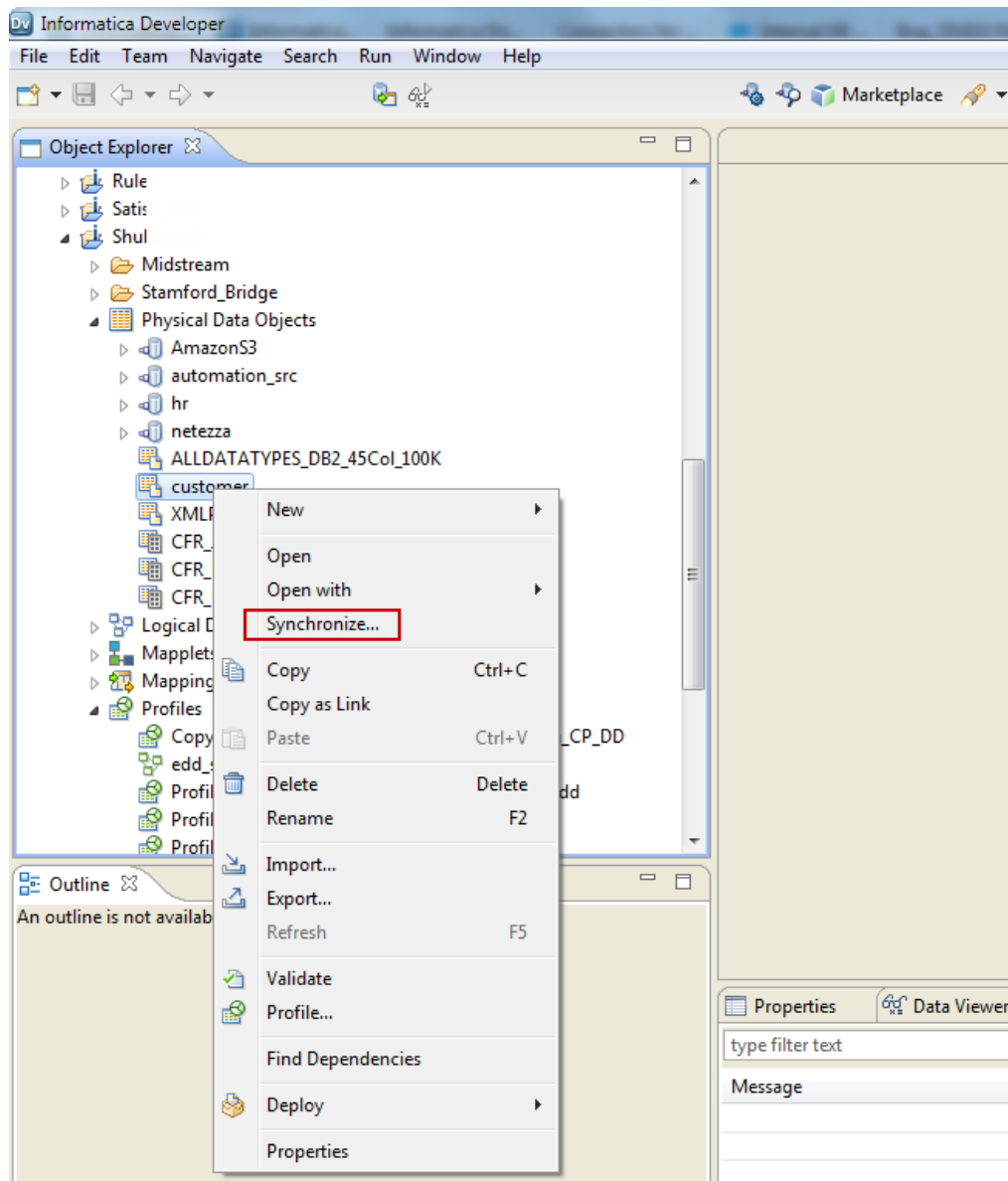
物理データオブジェクトを同期するには、【Object Explorer】ビューで右クリックして、【同期】を選択します。

Informatica Developer でのフラットファイルデータオブジェクトの同期

Informatica Developer で、外部フラットファイルデータソースに加えられた変更をそのデータオブジェクトに同期させることができます。データオブジェクトを同期させるには、【フラットファイルの同期】ウィザードを使用します。

1. 【オブジェクトエクスプローラ】ビューで、フラットファイルデータオブジェクトを選択します。
2. 右クリックし、【同期】を選択します。

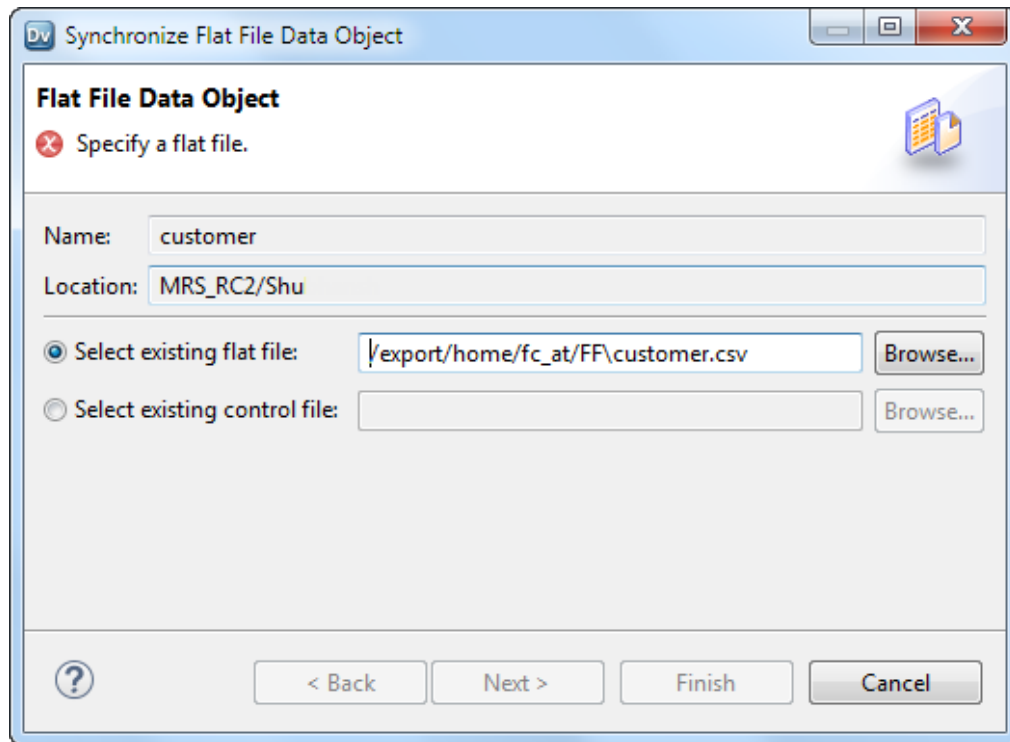
次の図は、データオブジェクトの同期オプションを示しています。



【フラットファイルデータオブジェクトの同期】ウィザードが表示されます。

3. 【フラットファイルデータオブジェクトの同期】ウィザードで、【既存のフラットファイルを選択】フィールドのフラットファイルパスを確認します。

次の図は、フラットファイルデータオブジェクトの同期ウィザードを示しています。



4. **【次へ】** をクリックします。
5. 必要に応じ、コードページ、形式、区切り形式プロパティ、およびカラムプロパティを選択します。
6. **【完了】** をクリックし、**【OK】** をクリックします。

Informatica Developer でのリレーショナルデータオブジェクトの同期

Informatica Developer で、リレーショナルデータソースの外部データソース変更をそのデータオブジェクトに同期させることができます。外部データソース変更には、カラムの追加、変更、削除や、ルールの変更などがあります。

1. **【Object Explorer】** ビューで、リレーショナルデータオブジェクトを選択します。
2. 右クリックし、**【同期】** を選択します。
操作の確認を求めるメッセージが表示されます。
3. 同期プロセスを実行するには、**【OK】** をクリックします。
同期プロセスのステータスメッセージが表示されます。
4. 「**同期が完了しました**」というメッセージが表示された時点で、**【OK】** をクリックします。
このメッセージには、データオブジェクトに加えられたメタデータ変更の概要が示されます。

物理データオブジェクトのトラブルシューティング

リレーショナルデータオブジェクトまたはカスタマイズデータオブジェクトのソーストランスフォーメーションのプレビューが失敗する。

リソースの所有者名が正しいことを確認します。

リレーショナルデータオブジェクトをインポートする際、ユーザー名とテーブルのインポート元のスキーマが一致しない場合、Developer tool は所有者名をインポートします。ユーザー名とテーブルのインポート元スキーマが一致しても、データベースのデフォルトスキーマ名が異なる場合、プレビューは失敗します。データ統合サービスが、テーブルが存在しないデータベースのデフォルトスキーマに対するクエリをプレビューするためです。

リレーショナルデータオブジェクトまたはソーストランスフォーメーションを更新し、正しいリソース所有者名を入力します。所有者名は、リレーショナルデータオブジェクトまたはソーストランスフォーメーションの【詳細】プロパティに表示されます。

Databricks 物理オブジェクトでマッピング検証が失敗する。

Databricks から物理データオブジェクトをインポートしたときにエスケープ文字を使用すると、マッピング検証が失敗します。オブジェクト名にエスケープ文字を使用しないでください。

第 6 章

フラットファイルデータオブジェクト

この章では、以下の項目について説明します。

- [フラットファイルデータオブジェクトの概要, 97 ページ](#)
- [ソースファイル名の生成, 98 ページ](#)
- [フラットファイルデータオブジェクトの概要プロパティ, 98 ページ](#)
- [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, 99 ページ](#)
- [制御ファイル, 107 ページ](#)
- [実行時のカラムの更新, 108 ページ](#)
- [制御ファイルからのカラムメタデータの生成, 109 ページ](#)
- [Excel からフラットファイルデータオブジェクトへのコピー, 111 ページ](#)
- [フラットファイルデータオブジェクトの作成, 113 ページ](#)

フラットファイルデータオブジェクトの概要

マッピング、マップレット、またはプロファイルに含めるフラットファイル物理データオブジェクトを作成します。フラットファイルデータオブジェクトは、読み取り、書き込み、およびルックアップトランスフォーメーションとして、マッピングおよびマップレットに追加できます。フラットファイルデータオブジェクトに対してプロファイルを作成し、実行できます。

フラットファイルデータオブジェクトは区切り形式または固定長です。バイナリデータを含まない固定長および区切り形式のフラットファイルからフラットファイルデータオブジェクトを作成できます。

実行時にフラットファイルデータソースの変更に対応する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。また、フラットファイルまたは制御ファイルの情報をを使用して、フラットファイルデータオブジェクトのカラム名を生成することもできます。制御ファイルには、カラム名、精度、スケール、処理バイト数などの情報が含まれます。

フラットファイルデータオブジェクトを作成したら、以下の各ビューを使用してフラットファイルプロパティを設定します。

【概要】 ビュー

フラットファイルデータオブジェクトの名前と説明を設定し、カラムプロパティを更新します。

【パラメータ】 ビュー

フラットファイルデータオブジェクトのパラメータを作成します。

【詳細】 ビュー

データ統合サービスがフラットファイルに対してデータの読み取りや書き込みを実行するときに使用する、フォーマットプロパティとランタイムプロパティを設定します。

フラットファイルデータオブジェクトを読み取りまたは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すると、**【プロパティ】** ビューでフラットファイルデータオブジェクトのフォーマットプロパティとランタイムプロパティを確認できます。マッピング内のフラットファイルプロパティは編集できません。

ソースファイル名の生成

ファイル名カラムをフラットファイルデータオブジェクトに追加できます。ファイル名カラムは、特定のデータレコードが含まれるソースファイルを特定するのに役立ちます。これは、データソースが複数ある場合に便利です。

ファイル名カラムはフラットファイルデータオブジェクトの**【概要】** ビューで設定できます。ファイル名カラムは、フラットファイルデータオブジェクトのオプションのカラムです。フラットファイルデータオブジェクトのファイル名カラムポートを使用して、ソースファイル名を各ソース行に書き込むようにマッピングを設定できます。ファイル名カラムには、完全修飾パスおよびファイル名が含まれます。

ファイル名カラムポートはソースデータオブジェクトの最後のカラムとして表示されます。ソースデータオブジェクトに追加できるファイル名カラムポートは1つのみです。ポートに「FileName」という名前が付いていて、ファイル名カラムポートの作成を試みると、ファイル名カラムポートは「FileName1」という名前になります。

例えば、マッピングでファイルのリストが含まれるソースファイルが使用されているとします。ファイルリスト内のファイル名は組織の部署名を参照しています。例えば、SYSA_Finance.txt ファイルには経理部のデータが含まれています。このマッピングでは、文字列関数を使用してファイル名カラムの出力から部署名を抽出できます。抽出された部署名を使用して、各部署ごとにデータを処理することができます。

フラットファイルデータオブジェクトの概要プロパティ

Data Integration Service は、フラットファイルに対してデータの読み取りや書き込みを行う際に概要プロパティを使用します。概要プロパティには全般プロパティが含まれ、フラットファイルデータオブジェクトに適用されます。カラムのプロパティも含まれ、フラットファイルデータオブジェクト内のカラムに適用されます。フラットファイルの概要プロパティは、Developer ツールにより**【概要】** ビューに表示されます。

以下の表に、フラットファイルに設定する全般プロパティを示します。

プロパティ	説明
名前	フラットファイルデータオブジェクトの名前。
説明	フラットファイルデータオブジェクトの説明。

以下の表に、フラットファイルに設定するカラムプロパティを示します。

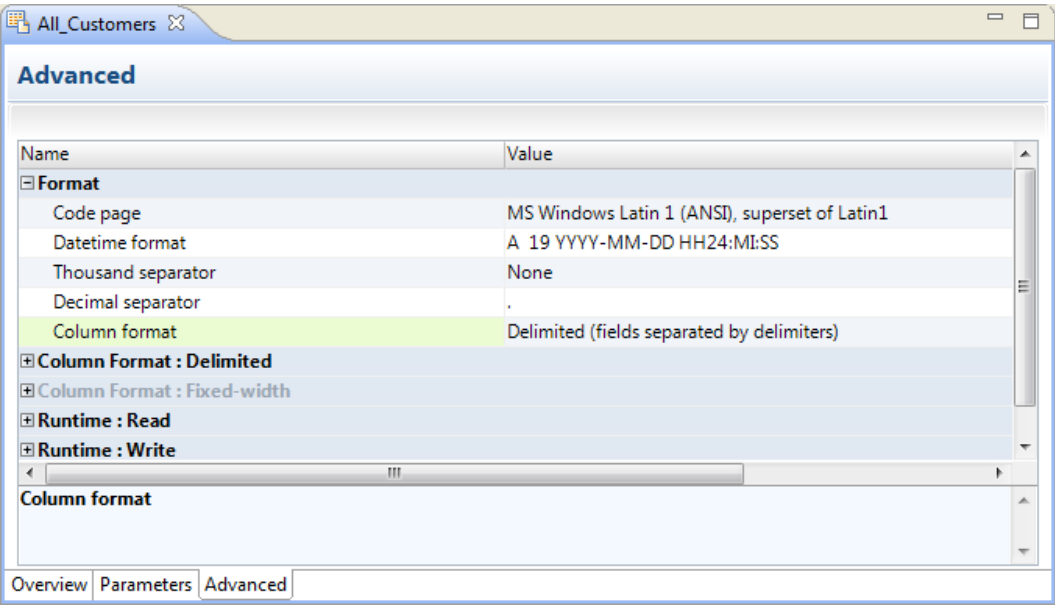
プロパティ	説明
名前	カラムまたはファイル名カラムポートの名前。
ネイティブタイプ	カラムのネイティブタイプ。
処理するバイト数（固定長フラットファイル）	データ統合サービスがカラムに対して読み取りまたは書き込みを行うバイト数。
精度	数値データ型の最大有効桁数、または文字列データ型の最大文字数です。数値データ型の場合、精度にはスケールが含まれます。
スケール	数値の小数点以下の最大桁数です。
形式	数値データ型および日時データ型のカラム形式。 数値データ型の場合、1000 ごとの区切りと小数点記号を定義します。デフォルトは 1000 ごとの区切りなしで、小数点記号はピリオドです (.)。 数値データ型の場合、年、月、日、および時間の表示形式を定義します。また、フィールド幅も定義します。デフォルトは、「A 19 YYYY-MM-DD HH24:MI:SS」です。
可視性	データ統合サービスがカラムに対してデータの読み取りおよび書き込みを実行できるかどうかを決定します。 例えば、可視性が「読み取り」の場合、データ統合サービスはカラムからデータを読み取ることができます。データをカラムに書き込むことはできません。 フラットファイルデータオブジェクトの場合、このプロパティは読み取り専用です。可視性は常に「読み取りおよび書き込み」です。
説明	カラムの説明。

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

データ統合サービスは、フラットファイルに対してデータの読み取りや書き込みを行う際に詳細プロパティを使用します。詳細プロパティには、フォーマットおよびランタイムプロパティが含まれています。フラットファイルの詳細プロパティは、Developer tool により **【詳細】** ビューに表示されます。

【詳細】 ビューには、折りたたみと展開が可能なプロパティセクションが含まれています。表示されるカラム形式セクションは、区切りカラム形式または固定長カラム形式のどちらを設定したかによって異なります。

次の図は、[詳細] ビューのプロパティセクションを示しています。



フォーマットのプロパティ

Developer tool では、[詳細] ビューの【フォーマット】セクションにフラットファイルデータオブジェクトのフォーマットプロパティが表示されます。

以下の表に、フラットファイルデータオブジェクトに設定するフォーマットプロパティを示します。

プロパティ	説明
コードページ	フラットファイルデータオブジェクトのコードページ。 ソースファイルの場合、ターゲットコードページのサブセットであるソースコードページを指定します。ルックアップファイルの場合、ソースコードページのスーパーセットかつターゲットコードページのサブセットであるコードページを指定します。ターゲットファイルの場合、ソースコードページのスーパーセットであるコードページを指定します。 デフォルトは、「MS Windows Latin 1 (ANSI), superset of Latin 1」です。
日時フォーマット	日時の値の表示形式およびフィールド幅を定義します。 デフォルトは、「A 19 YYYY-MM-DD HH24:MI:SS」です。
1000 ごとの区切り文字	数値に対する 1000 の区切り記号。 デフォルトは「なし」です。
小数位の区切り文字	数値の小数点記号。 デフォルトはピリオド「.」です。
カラム形式	フラットファイルのフォーマット。区切り形式または固定長です。

カラム形式: 区切りプロパティ

フラットファイルが区切り形式であれば、Developer tool は【詳細】ビューの【カラム形式: 区切り記号付き】セクションに区切りプロパティを表示します。

以下の表に、フラットファイルデータオブジェクトに設定する区切りプロパティを示します。

プロパティ	説明
区切り文字	データのカラムを区切るために使用される文字。 区切り文字フィールドをクリックして文字を選択するかパラメータをプロパティに割り当てます。区切り文字は印字可能であり、かつテキスト修飾子やエスケープ文字を選択した場合にはそれとは異なる文字である必要があります。 デフォルトは [カンマ] です。
テキスト修飾子	引用符はテキスト文字列の境界を定義します。引用符を選択すると、引用符で囲まれた文字列内の区切り文字が無視されます。 デフォルトは [引用符なし] です。
次の行からインポートを開始	データ統合サービスがデータのインポートを開始する行。ヘッダ行をスキップする場合には、このオプションを選択します。 デフォルトは 1 です。
行区切り文字	データ行を区切る文字の 8 進コード。デフォルトはラインフィード、\012 LF (\n) です。 注: 行区切り文字は、ソースデータの読み取りに適用されます。データ統合サービスは、ターゲットファイルに書き込むときには、常にデフォルトの区切り文字 \n を使用します。
エスケープ文字	区切り文字がエスケープ文字の直後にある場合に、引用符で囲まれていない文字列の区切り文字をエスケープする文字。エスケープ文字を指定すると、データ統合サービスは、文字列に含まれる通常の文字として区切り文字を読み込みます。 注: ソースファイルに引用符やエスケープ文字が含まれていないと、マッピングのパフォーマンスをある程度向上できます。
データのエスケープ文字を維持	出力文字列にエスケープ文字を含めます。 デフォルトでは無効になっています。
連続した区切り文字は 1 文字として扱う	データ統合サービスで、1 つ以上の連続したカラム区切り文字が 1 つのカラム区切り文字として扱われます。このオプションを選択しない場合、データ統合サービスは 2 つの連続した区切り文字を 1 つの NULL 値として読み取ります。 デフォルトでは無効になっています。

カラム形式: 固定幅プロパティ

フラットファイルが固定幅の場合、Developer tool の【詳細】ビューの【カラム形式: 固定幅】セクションに固定幅プロパティが表示されます。

以下の表に、フラットファイルデータオブジェクトに設定する固定幅プロパティを示します。

プロパティ	説明
NULL 文字タイプ	NULL 文字タイプ。テキストまたはバイナリです。
NULL 文字値	NULL 値を表す文字。NULL 文字には、ファイルコードページの任意の有効な文字、または 0 - 255 の任意のバイナリ値を指定できます。
NULL 文字を繰り返す	ソースファイルの場合、データ統合サービスは繰り返し NULL 文字を 1 つの NULL 値として単一フィールドで読み取ります。 ターゲットファイルの場合、データ統合サービスは可能な限り多くの NULL 文字をターゲットフィールドに書き込みます。このオプションを有効にしない場合、データ統合サービスはフィールドの最初に 1 つの NULL 文字を入力して NULL 値を表します。 デフォルトでは無効になっています。
次の行からインポートを開始	データ統合サービスがデータのインポートを開始する行。ヘッダ行をスキップする場合には、このオプションを選択します。 デフォルトは 1 です。
レコード間でスキップするバイト数	行の最終カラムと次の行の最初のカラムの間のバイト数。データ統合サービスは、指定されたバイト数を行の最後でスキップして、復帰文字や改行文字を読み込まないようにします。 UNIX ファイルの場合は「1」、DOS ファイルの場合は「2」を入力します。 デフォルトは 2 です。
行順序あり	データ統合サービスは、最後のカラムの改行文字または復帰文字をこのカラムの末尾として読み取ります。ファイルで改行または復帰を使用して各行の最後のカラムを短くしている場合に、このオプションを選択します。 デフォルトでは無効になっています。
後続のスペースを削除する	末尾の空白を文字列値から削除します。 デフォルトでは無効になっています。
ユーザー定義シフトステート	【カラム】プロパティのソースカラムのシフトステートを選択できるようにします。 ソースファイルにマルチバイトとシングルバイトデータの両方が含まれ、シフトインとシフトアウトキーが含まれない場合にこのオプションを選択します。 マルチバイトファイルソースにシフトキーが含まれない場合、フラットファイルデータオブジェクトで各カラムにシフトキーを選択する必要があります。データ統合サービスが各文字を正しく読み取れるように、各カラムにシフトキーを選択します。 デフォルトでは無効になっています。

ランタイム:読み取りプロパティ

Developer tool では、[詳細] ビューの [ランタイム: 読み取り] セクションに、フラットファイルソースのランタイムプロパティが表示されます。データ統合サービスは、フラットファイルからデータを読み取る際にこの情報を使用します。

以下の表に、フラットファイルデータオブジェクトに設定する読み取りプロパティを示します。

プロパティ	説明
入力タイプ	ソース入力のタイプ。次のソース入力のタイプを選択できます。 <ul style="list-style-type: none">- ファイル。フラットファイルのソースの場合に選択します。- コマンド。シェルコマンドによって生成されたソースデータまたはファイルリスト。
ソースタイプ	同じファイルプロパティを持つファイルのソースタイプを示します。次のいずれかのソースタイプを選択できます。 <ul style="list-style-type: none">- Direct。ソースデータを含むソースファイル。- Indirect。ファイルのリストを含むソースファイル。データ統合サービスはファイルリストを読み取り、ファイル順に読み取ります。- Directory。ディレクトリにあるソースファイル。ソースファイルのディレクトリプロパティでディレクトリの場所を指定する必要があります。データ統合サービスは、アルファベットの昇順にファイルを読み取ります。データ統合サービスはサブディレクトリのファイルを読み取りません。
ソースファイル名	フラットファイルソースのファイル名。
ソースファイルのディレクトリ	フラットファイルのソースが存在するディレクトリ。Informatica サービスをホストするマシンは、このディレクトリにアクセスできる必要があります。デフォルトは SourceDir システムパラメータです。
同時読み込みのパーティション化	複数のパーティションがソースファイルから入力行を読み取る順序。データ統合サービスがマッピングにパーティションを作成しない場合、この値は無視されます。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- スループットの最適化。データ統合サービスが入力行の順序を保持しません。- 相対順序を維持。データ統合サービスが、パーティションごとに読み込まれる行に対し、入力行の順序を保持します。- 絶対順序を維持。データ統合サービスは、すべてのパーティションで読み込まれるすべての行に対して、入力行の順序を保持します。
接続タイプ	接続のタイプ。次のオプションから選択します。 <ul style="list-style-type: none">- なし。ソースファイルは接続を必要としません。- Hadoop ファイルシステム。ソースファイルは HDFS 内に存在します。 デフォルトは [なし] です。

プロパティ	説明
コマンド	<p>ソースファイルデータを生成するために使用するコマンド。コマンドを使用してフラットファイルデータを生成または変換し、コマンドの標準出力をマッピングの実行時にフラットファイルリーダーに送信できます。フラットファイルリーダーは、標準出力をフラットファイルソースデータとして読み取ります。コマンドを使用してソースデータを生成すると、フラットファイルソースをステージングする必要がなくなります。</p> <p>コマンドまたはスクリプトを使用すると、マッピング前のコマンドを使用してフラットファイルソースを生成する代わりに、データ統合サービスに直接ソースデータを送信できます。コマンドを使用してファイルリストを生成することもできます。</p> <p>例えば、ファイルリストとしてディレクトリリストを使用するには、以下のコマンドを使用します。</p> <pre>cd MySourceFiles; ls sales-records-Sep-*-2005.dat</pre>
文字列の NULL の切り捨て	<p>最初の NULL 文字および最初の NULL 文字に続くすべての文字を文字列値から取り除きます。</p> <p>このオプションは、文字列に NULL 文字を含む区切りフラットファイルに対して有効にします。このオプションを有効にしない場合、データ統合サービスは文字列に NULL 文字を含む行に対して行エラーを生成します。</p> <p>デフォルトでは無効になっています。</p>
行順序ありのバッファ長	<p>データ統合サービスが読み取る 1 行のバイト数。</p> <p>データ統合サービスは、このプロパティと合計の行サイズを使用して、行を削除するかどうかを決定します。行が、行順序ありのバッファ長、または合計の行サイズのどちらか大きい方のサイズを超える場合、データ統合サービスはその行を削除してマッピングログファイルに書き込みます。行のサイズを決定するには、カラムの精度と区切り文字を追加して、その合計に各文字の最大バイト数を掛けます。</p> <p>デフォルトは 1024 です。</p>
ランタイムカラム名の生成	<p>実行時にカラムメタデータを生成する方法を決定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 自動 (Field1、Field2...)。データ統合サービスは、データ内のカラムの位置に基づいて、カラムメタデータを取り込みます。フラットファイルのカラムメタデータは、フラットファイルデータオブジェクトの形式プロパティの 【開始インポート行】 属性に指定された行を開始位置として、実行時に構築されます。 - データファイルヘッダーから (最初の行)。フラットファイルの先頭行にヘッダ行があり、データ統合サービスはこのヘッダ行を使用してカラムメタデータを決定します。 - 制御ファイルから。データ統合サービスは、カラム名、精度、データ型、スケール、処理バイト数など、制御ファイル内のデータに基づいてカラムメタデータを構築します。
制御ファイル名	制御ファイルの名前。制御ファイルからランタイムカラム名を生成する場合は必須。
制御ファイルディレクトリ	制御ファイルが存在するディレクトリ。制御ファイルからランタイムカラム名を生成する場合は必須。
デフォルトのフィールドタイプ	実行時に生成される追加ポートのデータ型。
デフォルトの精度	実行時に生成される追加ポートの精度。

プロパティ	説明
デフォルトのスケール	実行時に生成される追加ポートのスケール。
制約	<p>データ行の各値が満足する必要がある条件式。式エディタを使用して、TRUE に評価される式を入力します。データ統合サービスは、制約を読み取ると、TRUE に評価されない行を削除します。</p> <p>例えば、ソースフラットファイルに AGE カラムがあるとします。このとき、フラットファイルデータオブジェクトに対して「AGE < 70」という制約を設定できます。データ統合サービスは、ソースフラットファイルから制約「AGE < 70」を満たす行を読み取ります。データ統合サービスは「AGE >= 70」の行を読み取ると、その行を削除します。</p>

ランタイム:書き込みプロパティ

Developer tool では、[詳細] ビューの [ランタイム: 書き込み] セクションに、フラットファイルターゲットのランタイムプロパティが表示されます。データ統合サービスは、フラットファイルに対してデータの書き込みを行う際にこの情報を使用します。

以下の表に、フラットファイルデータオブジェクトに設定する書き込みプロパティを示します。

プロパティ	説明
存在する場合は追加	<p>ターゲットファイルおよび拒否ファイルに出力データを追加します。</p> <p>このオプションを選択しない場合、データ統合サービスはターゲットファイルおよび拒否ファイルを切り詰めてからデータを書き込みます。ファイルが存在しない場合、データ統合サービスによって作成されます。</p> <p>デフォルトでは無効になっています。</p>
存在しない場合はディレクトリを作成	<p>ターゲットディレクトリが存在しない場合は作成します。</p> <p>デフォルトでは無効になっています。</p>
ヘッダーのオプション	<p>ファイルターゲットにヘッダ行を作成します。以下のオプションを選択することができます。</p> <ul style="list-style-type: none"> - ヘッダなし。フラットファイルターゲットにヘッダ行を作成しません。 - 出力フィールド名。出力ポート名が指定されたファイルターゲットにヘッダ行を作成します。 - ヘッダコマンド出力を使用。ヘッダ行を生成するには、[ヘッダコマンド] フィールドでこのコマンドを使用します。たとえば、ファイルターゲットのヘッダ行にデータを追加するコマンドを使用できます。 <p>デフォルトは [ヘッダーなし] です。</p>
ヘッダーコマンド	ファイルターゲットにヘッダ行を生成するために使用するコマンド。
フッタのコマンド	ファイルターゲットにフッタ行を生成するために使用するコマンド。
出力タイプ	マッピングのターゲットのタイプ。ターゲットデータをフラットファイルに書き込むためのファイルを選択します。データをコマンドに出力するコマンドを選択します。

プロパティ	説明
コマンド	<p>ターゲットデータの処理に使用するコマンド。</p> <p>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。例えば、UNIX 上のターゲットデータから圧縮ファイルを作成するには、次のコマンドを使用します。</p> <pre>compress -c - > MyTargetFiles/MyCompressedFile.Z</pre> <p>Windows では、有効な任意の DOS コマンドまたはバッチファイルを使用します。フラットファイルライターは、フラットファイルターゲットではなくコマンドにデータを送信します。例えば、ファイルシステム内のステージングデータの回避やセキュリティ違反の回避には、Windows 上でターゲットコマンドとして cmd を使用します。</p> <p>データ統合サービスではなくコマンドにトランスフォーメーションタスクをプッシュすれば、マッピングのパフォーマンスを向上させることができます。コマンドを使用してターゲットデータのソートや圧縮を行うこともできます。</p>
マージコマンド	<p>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に使用するマージコマンド。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。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。</p>
出力ファイルディレクトリ	<p>フラットファイルターゲットの出力ディレクトリ。Informatica サービスをホストするマシンは、このディレクトリにアクセスする必要があります。</p> <p>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、パフォーマンスを向上させるために、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。</p> <p>デフォルトは TargetDir システムパラメータです。</p>
出力ファイル名	<p>フラットファイルターゲットのファイル名。</p> <p>複数のパーティションがフラットファイルのターゲットに書き込む場合でターゲットデータをマージしないことにした場合は、各パーティションが <output_file_name><partition_number>.out という形式で命名される出力ファイルに別々に書き込みを行います。</p>
マージタイプ	<p>データ統合サービスがパーティション化されたターゲットのデータに対して実行するマージのタイプ。データ統合サービスがターゲットにパーティションを作成しない場合、この値は無視されます。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 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。 - 連続。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、マッピングの最後にそれらを 1 つのマージファイルにマージします。 - 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。 - 同時進行。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。データ統合サービスによってすべてのパーティションに対しマージファイルに同時に書き込まれるため、統合ファイル内のデータのソート順は連続していない可能性があります。
マージファイルディレクトリ	<p>すべてのターゲットパーティションに関するマージファイルのディレクトリ。Informatica サービスをホストするマシンは、このディレクトリにアクセスする必要があります。</p> <p>デフォルトは TargetDir システムパラメータです。</p>

プロパティ	説明
マージファイル名	すべてのターゲットパーティションに関するマージファイルの名前。デフォルトは出力ファイル名です。
接続タイプ	接続のタイプ。次のオプションから選択します。 <ul style="list-style-type: none"> - なし。ターゲットファイルは接続を必要としません。ターゲットファイルの場所は、出力ファイルディレクトリによって指定されます。 - Hadoop ファイルシステム。ターゲットファイルは HDFS 内にあります。 デフォルトは [なし] です。

制御ファイル

データ統合サービスでは、制御ファイルに基づいてフラットファイルのカラムメタデータを更新できます。

制御ファイルは、フィールド名、データ型、精度、および位取りが含まれる単純なテキストファイルです。制御ファイルの各行には 1 つのカラムのデータが含まれます。

実行時に制御ファイルからカラム名を生成するには、フラットデータオブジェクトの [詳細] プロパティで **[ランタイムカラム名の生成]** プロパティの **[制御ファイルから]** を選択します。読み取りトランスフォーメーション設定して実行時にカラムメタデータを取得する必要があります。

制御ファイルに次のデータ型を指定できます。

- Bigint
- 日付/時刻
- 10 進型
- 倍精度浮動小数点数型
- 整数型
- 文字列型
- Text
- TimestampwithTZ

制御ファイルを使用してフラットファイルデータオブジェクトに基づく読み取りトランスフォーメーションのランタイムカラム名の生成または、フラットファイルデータオブジェクトの作成ができます。データ統合サービスは制御ファイルに存在する情報を使用して、フラットファイルのカラムメタデータを構築します。カラム名、精度、データ型、およびスケールの各属性はカンマで区切られます。カラム間の情報は改行文字で区切られます。

実行時のカラムの更新

実行時のソースメタデータの変更に对应できる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。データ統合サービスは、カラムの数または順序がフラットファイルの物理データオブジェクトのカラムの数または順序と異なる場合に、フラットファイルからデータを読み取ることができます。

例えば、マッピングのソースが別の部門から提供されているとします。ソースの提供元の部門は、ソースファイルのカラムの順序を保証できません。場合により、その部門がファイルのカラムを変更していたり、追加のカラムを含めていたりすることがあります。実行時のソースメタデータの変更を受け付ける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。

ランタイムカラム名は、ソースの任意の行、フラットファイルヘッダー、または制御ファイルから自動的に生成できます。

入力ルールを使用すると、フラットファイルデータオブジェクトに基づいて読み取りトランスフォーメーションからマッピング内の次のトランスフォーメーションにランタイムカラムを射影できます。

[データオブジェクト] タブの [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] を選択するように、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プロパティを設定できます。このプロパティを設定すると、フラットファイルデータオブジェクトのオプションによって、データ統合サービスによるデータの処理方法が決まります。

ランタイムカラム名の自動生成

データ統合サービスでは、カラム位置に基づいて自動的にフラットファイルのカラムメタデータを更新できます。

データ統合サービスは、フラットファイルの形式プロパティにインポート開始のために指定する行に基づいて、フラットファイルのカラムメタデータを動的に更新します。カラム位置を変更せずに、デフォルトのデータ型、精度、およびスケールがランタイムカラムのカラムメタデータに使用されます。

ランタイムに自動でカラム名を生成するには、フラットファイルデータオブジェクトの [詳細] プロパティの [ランタイムカラム名の生成] プロパティで [自動 (Field1、Field2...)] を選択します。読み取りトランスフォーメーション設定して実行時にカラムメタデータを取得する必要もあります。

例えば、フラットファイルデータオブジェクトを設定し、実行時に変更を受け入れます。カラム位置に基づいて、フラットファイルのデータによってカラムメタデータが決まります。カラム Dept、Name、Place を使用して、フラットファイルデータオブジェクトを定義しました。実行時にデータ統合サービスでフラットファイルのカラムメタデータを動的に構築する必要があります。

入力ファイルには、以下の情報が含まれます。

```
HR,Bob,Chicago,US,87675
Finance,Mary,California,US,65437,t567,4200
```

[形式] プロパティでフラットファイルデータオブジェクトを設定して、行 1 からデータのインポートが開始するようにしました。最初の行に 5 つのカラムがあるため、データ統合サービスは 5 つのカラムがあるフラットファイルデータオブジェクトを処理します。データ統合サービスは、位置に基づいて一意の名前 (Field4 とおよび Field5) で、フラットファイルデータオブジェクトに追加のランタイムカラムを作成します。2 番目の行で追加のカラムは処理されません。

マッピングにフラットファイルデータオブジェクトを追加する場合、読み取りトランスフォーメーションを設定して実行時にカラムメタデータを取得します。実行時にデータ統合サービスは、フラットファイルデータオブジェクトを構築し、次の値を使用してフラットファイルの行を処理します。

Dept	Name	Place	Field4	Field5
HR	Bob	Chicago	US	87675
Finance	Mary	California	US	65437

データファイルヘッダからのランタイムカラム名の生成

データ統合サービスでは、ヘッダー行を使用してカラムメタデータを決定できます。フラットファイルの最初の行がヘッダーです。

デフォルトのデータ型、精度、およびスケールがランタイムカラムのカラムメタデータに使用されますが、実行時にカラム位置は変更できます。

実行時にデータファイルヘッダーからカラム名を生成するには、フラットデータオブジェクトの [詳細] プロパティで **ランタイムカラム名の生成** プロパティの **データファイルヘッダーから (最初の行)** を選択します。読み取りトランスフォーメーション設定して実行時にカラムメタデータを取得する必要があります。

例えば、フラットファイルデータオブジェクトを設定し、実行時に変更を受け入れます。フラットファイルのデータで、データファイルヘッダーに基づいてカラムメタデータを決定します。

カラム Name、Age、Dept があるフラットファイルデータオブジェクトを定義したとします。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を設定し、データファイルヘッダーから実行時にカラム名を生成できます。

データフラットファイルには、以下の情報が含まれます。

Dept	Name	Place	Country	Phone_No.
HR	Bob	Chicago	US	87675
Finance	Mary	California	US	65437 t567 4200

マッピングにフラットファイルデータオブジェクトを追加する場合、読み取りトランスフォーメーションを設定して実行時にカラムメタデータを取得します。ランタイム中、データ統合サービスは、Dept、Name、Place、Country および Phone_No のカラムがあるフラットファイルデータオブジェクトを生成します。Age カラムがフラットファイルヘッダー行に存在しないため、実行時に構成されるフラットファイルデータオブジェクトには Age カラムがありません。

データ統合サービスは Country や Phone Number などのランタイムカラムをマッピングルールに基づいてダウンストリームのトランスフォーメーションにプロパゲートします。

制御ファイルからのカラムメタデータの生成

制御ファイルは単純なテキストファイルであり、それを基にしてフラットファイルデータオブジェクトを作成できます。制御ファイルを使用して、実行時にフラットファイルデータオブジェクトに基づいて読み取りトランスフォーメーションのカラムを更新することもできます。

データ統合サービスが実行時にカラムを生成するときは、フラットファイルデータオブジェクトのプロパティに指定されている小数点記号および桁区切り記号を使用します。フラットファイルデータオブジェクトにカラムが存在する場合、制御ファイルで指定する必要があるのは、カラムの名前のみです。

フラットファイルソースの [詳細] ビューのランタイムプロパティでソースタイプを間接として指定した場合、実行時に制御ファイルを使用してカラム名を生成できます。

区切りフラットファイルまたは固定長フラットファイルの制御ファイルを作成できます。

フラットファイルデータオブジェクトのランタイムプロパティの制御ファイル名と制御ファイルディレクトリはパラメータ化できます。

データ統合サービスは、制御ファイルおよび入力データに指定したカラムメタデータに基づいて制御ファイルを処理します。

実行時に制御ファイルからカラム名を生成し、ソースデータオブジェクトにファイル名カラムが含まれている場合、データ統合サービスは検証エラーを重複するファイル名でログに記録します。

制御ファイルの形式

区切りフラットファイルと固定長フラットファイルの制御ファイル形式は異なります。制御ファイルを使用して、実行時にカラムメタデータを生成したり、フラットファイルデータオブジェクトを作成したりできます。

以下のいずれかの形式を使用して、制御ファイルを作成します。

区切り制御ファイル

区切り制御ファイルの各行の形式は以下のとおりです。

```
[column name],<data type>,<precision>,<scale>
```

区切りフラットファイルの制御ファイルの例を以下に示します。

```
Dept,String,10
Name
Place,String,20
Country
Phone number,string,30
```

固定長制御ファイル

固定長制御ファイルの各行の形式は以下のとおりです。

```
[column name],<data type>,<precision>,<scale>,<number of bytes to process>
```

区切りフラットファイルの制御ファイルの例を以下に示します。

```
Dept,String,10
Name
Place,String,20
Country
Phone number,string,30
```

固定長ソースの場合、制御ファイルには、処理バイト数用の追加カラムが含まれます。処理バイト数を指定しない場合、データ統合サービスは、精度に指定された値を処理バイト数として使用します。

ランタイムプロパティのパラメータ化

追加のカラムのデフォルトのスケール、精度、およびデータ型の値は、フラットファイルデータオブジェクトのランタイムプロパティの制御ファイルからパラメータ化または指定できます。フラットファイルデータオブジェクトのランタイムプロパティの制御ファイル名と制御ファイルディレクトリは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。

制御ファイルのランタイム処理

マッピングを開発する際には、フラットファイルからデータを読み取る方法を決めるデータオブジェクト読み取りプロパティを定義します。データ統合サービスは、制御ファイルに基づくフラットファイルデータオブジェクトのカラムを処理できます。

データ統合サービスは、制御ファイルに基づいてカラムメタデータを構築する際に、以下のデータ処理基準を適用します。

フラットファイルデータオブジェクトに指定されていないカラムプロパティについては、デフォルト値を適用する

データ統合サービスは、制御ファイルにランタイムカラムプロパティが指定されていない場合、フラットファイルデータオブジェクトのデフォルトのランタイムプロパティを適用します。また、制御ファイルの追加カラムにデータ型、精度、スケールが指定されていない場合は、デフォルトのデータ型、精度、スケールを使用します。

制御ファイルには現れないがフラットファイルデータオブジェクトには存在するカラムについては、データを NULL として処理する

制御ファイルにフラットファイルデータオブジェクトのカラムが存在しない場合、データ統合サービスは、データプレビュー時にデータを NULL として処理します。これらのカラムは制御ファイルに存在しないため、データ統合サービスが実行時に処理することはできません。

データ統合サービスは、ソースメタデータが制御ファイルと一致しない場合、マッピングに失敗する

制御ファイルに基づくソースメタデータが入力データと一致しない場合、データ統合サービスはデータの処理に失敗します。例えば、ソースに制御ファイルのデータ型が含まれており、それがソース内のデータ型と互換性がない場合、データ統合サービスはマッピングに失敗します。

制御ファイルに関するルールおよびガイドライン

制御ファイルを使用する場合、以下のガイドラインに従います。

- 制御ファイルは、データ統合サービスが実行されているマシンにアクセスする必要があります。
- 制御ファイルをフラットファイルデータオブジェクトとしてインポート後、データオブジェクトのカラムのデータ型、精度、およびスケールは変更できません。制御ファイルから追加のカラムのカラム位置およびカラムメタデータは変更できません。
- 制御ファイルに基づいて、実行時にカラム名を生成する場合、ルックアップキャッシュに制御ファイルのメタデータが反映されます。マッピングを一度実行した後、追加のカラムがある同じ制御ファイルを更新する場合、ルックアップトランスフォーメーションの「ランタイム」プロパティから「ルックアップソースからの再キャッシュ」を選択する必要があります。
- 制御ファイルで日付形式のカラムを指定することはできません。データ統合サービスは、「詳細」ビューの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されているフラットファイルの日付形式を使用します。フラットファイルソースに複数の日付形式がある場合、データ統合サービスはすべての日付の「詳細」ビューに指定されている 1 つの日付形式のみを使用します。
- 制御ファイルでバイナリデータ型は指定できません。制御ファイルに TimestampwithTZ データ型が含まれる場合、そのデータ型の精度が 36、スケールが 9 に設定されていることを確認します。

Excel からフラットファイルデータオブジェクトへのコピー

Excel でフラットファイルプロパティを設定してから、Developer tool のフラットファイルデータオブジェクトにコピーできます。フラットファイルプロパティには、カラム名、ネイティブタイプ、精度、スケールなどがあります。多くのカラムのあるフラットファイルデータオブジェクトを作成または編集する必要がある場合に、このような作業をお勧めします。

注: 各セルの値をフラットファイルデータオブジェクトにコピーする前に、その値が有効であることを確認する必要があります。例えば、文字列タイプには「0」以外のスケール値を指定できません。精度値には文字を指定できず、タイプ値には数値を指定できません。情報が適切でない場合は、エラーメッセージが表示されます。

Excel でのフラットファイルデータオブジェクトの編集

フラットファイルデータオブジェクトの大部分を編集する場合、Developer tool ですべての値を変更する必要はありません。代わりに、フラットファイルのカラムを Excel にコピーし、オートフィル機能を使用して、すべての値を同時に変更してから、フラットファイルをもう一度 Developer tool に貼り付け（置換）できます。

1. Developer tool からメタデータをコピーするには、フラットファイルデータオブジェクト内を右クリックして、**[すべて選択]** をクリックします。
2. メタデータを Excel スプレッドシートにコピーします。
3. Excel スプレッドシート内で変更します。
4. Excel からメタデータをコピーします。
5. 変更した内容でフラットファイルデータオブジェクトを更新するには、フラットファイルカラム内を右クリックして、**[貼り付け（置換）]** をクリックします。

フラットファイルデータオブジェクトへのメタデータのコピー

Excel でメタデータを作成し、それを Developer tool のフラットファイルデータオブジェクトにコピーできます。

1. フラットファイルデータオブジェクトを含むマッピングを作成します。
2. Excel でフラットファイルデータオブジェクトのメタデータを定義します。
3. Excel からメタデータをコピーします。
4. メタデータをフラットファイルデータオブジェクトに移動するには、カラム内を右クリックして、**[貼り付け（置換）]** をクリックします。次の図は、Excel テーブルの例と、メタデータを Developer tool にコピーした後に作成されるフラットファイルデータオブジェクトを示しています。

	A	B	C	D
1	Name	Native type	Precision	Scale
2	busn_addr_zip_code	string	60	0
3	cust_day_phone	string	40	0
4	busn_addr_cntry_code	string	3	0
5	pmry_email_addr	string	127	0
6	busn_srvc_email	string	127	0
7	cust_acct_type_code	number	1	0

	Name	Native Type	Precision	Scale
1	busn_addr_zip_code	string	60	0
2	cust_day_phone	string	40	0
3	busn_addr_cntry_code	string	3	0
4	pmry_email_addr	string	127	0
5	busn_srvc_email	string	127	0
6	cust_acct_type_code	number	1	0

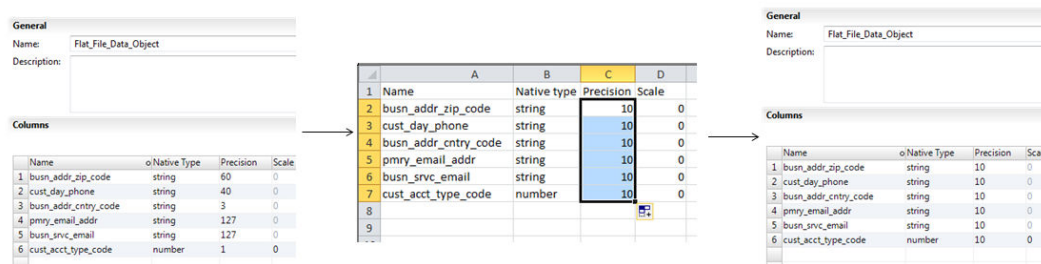
メタデータを Developer tool にコピーした後も、フラットファイルデータオブジェクトを変更できます。

Excel でのデータオブジェクトの編集例

フラットファイルデータオブジェクトを作成しており、すべての精度値を 10 に変更する必要があります。各フィールドを個別に変更するのではなく、Excel でグローバルに変更を行います。

メタデータを Developer tool から Excel にコピーし、それを変更してから、Developer tool のフラットファイルカラムにコピーします。Excel を使用すると、各フィールドを個別に変更する必要がなくなります。

次の図は、フラットファイルを Excel に移動し、オートフィル機能を使用して特定の値を変更してから、メタデータを Developer tool のフラットファイルデータオブジェクトにコピーするプロセスを示しています。



フラットファイルデータオブジェクトの作成

フラットファイルデータオブジェクトは、空のデータオブジェクトとして、既存のフラットファイルまたは制御ファイルから作成できます。区切り形式または固定長のフラットファイルデータオブジェクトを任意の方法で作成できます。

フラットファイルデータオブジェクトを作成するには、以下の方法でデータオブジェクトを作成できます。

空のデータオブジェクトとして作成

Developer tool でデータオブジェクトのカラムと行を定義する場合は、空のフラットファイルデータオブジェクトを作成します。

既存のフラットファイルから作成

データオブジェクトに含めるメタデータを定義したフラットファイルが用意されている場合は、その既存のフラットファイルからフラットファイルデータオブジェクトを作成します。

制御ファイルから作成

制御ファイルに基づいてデータオブジェクトのカラムと行を定義する場合は、制御ファイルからフラットファイルデータオブジェクトを作成します。

空のフラットファイルデータオブジェクトの作成

Developer tool でデータオブジェクトのカラムと行を定義する場合は、空のフラットファイルデータオブジェクトを作成します。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】** > **【新規】** > **【データオブジェクト】** をクリックします。
3. **【物理データオブジェクト】** > **【フラットファイルデータオブジェクト】** を選択し、**【次へ】** をクリックします。
【新しいフラットファイルデータオブジェクト】 ダイアログボックスが表示されます。
4. **【空として作成】** を選択します。
5. データオブジェクトの名前を入力します。
6. 必要に応じて、**【参照】** ボタンをクリックして、データオブジェクトのプロジェクトまたはフォルダを選択します。
7. **【次へ】** をクリックします。
8. 処理するデータのコードページに一致するコードページを選択します。

9. **【区切り】** または **【固定長】** を選択します。
10. **【固定長】** を選択した場合、**【完了】** をクリックします。**【区切り】** を選択し、**【次へ】** をクリックします。
11. 次の区切り形式のプロパティを設定します。

プロパティ	説明
区切り文字	データのカラムを区切るために使用される文字。エスケープ文字またはテキスト修飾子と同じ区切り文字を入力すると、予期しない結果が生成されることがあります。
引用符	引用符はテキスト文字列の境界を定義します。引用符を選択すると、引用符で囲まれた文字列内の区切り文字が無視されます。

12. **【完了】** をクリックします。
 エディタに空のデータオブジェクトが開きます。**【概要】** ビューでデータオブジェクトのカラムを定義します。

既存のフラットファイルからのフラットファイルデータオブジェクトの作成

データオブジェクトに含めるメタデータを定義したフラットファイルが用意されている場合は、その既存のフラットファイルからフラットファイルデータオブジェクトを作成します。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】** > **【新規】** > **【データオブジェクト】** をクリックします。
3. **【物理データオブジェクト】** > **【フラットファイルデータオブジェクト】** を選択し、**【次へ】** をクリックします。
【新しいフラットファイルデータオブジェクト】 ダイアログボックスが表示されます。
4. **【既存のフラットファイルから作成】** を選択します。
5. **【参照】** をクリックし、ファイルを含むディレクトリに移動します。
6. **【開く】** をクリックします。
 選択したファイルと同じ名前がデータオブジェクトに付けられます。
7. 必要に応じて、データオブジェクト名を編集します。
8. 必要に応じて、**【参照】** ボタンをクリックして、データオブジェクトのプロジェクトまたはフォルダを選択します。
9. **【次へ】** をクリックします。
10. 処理するデータのコードページに一致するコードページを選択します。
11. **【区切り】** または **【固定長】** を選択します。
12. 必要に応じて、プレビューする行の最大数を編集します。
13. **【次へ】** をクリックします。
14. フラットファイルが区切りまたは固定長のどちらであるかに応じて、フォーマットプロパティを設定します。

- 区切りフラットファイルの場合は、次のプロパティを設定します。

プロパティ	説明
区切り文字	データのカラムを区切るために使用される文字。エスケープ文字またはテキスト修飾子と同じ区切り文字を入力すると、予期しない結果が生成されることがあります。
引用符	引用符はテキスト文字列の境界を定義します。引用符を選択すると、引用符で囲まれた区切り文字が無視されます。
1 行目からカラム名をインポート	選択すると、Developer tool は最初の行のデータをカラム名として使用します。カラム名が最初の行に含まれている場合に、このオプションを選択します。有効でないフィールド名にはプレフィックス「FIELD_」が付きます。
行区切り文字	改行文字を指定します。リストから選択するか、文字を入力します。8 進コードの前にはバックスラッシュ (\) をつけます。1 文字を使用するには、その文字を入力します。 入力した文字の先頭にバックスラッシュが付いていない場合は、データ統合サービスは最初の文字だけを使用します。文字は 1 バイト文字とします。コードページ内のそれ以外の文字には、そのバイトが含まれてはいけません。デフォルトは、改行、\012 LF (\n) です。
エスケープ文字	引用符で囲まれていない文字列におけるカラム区切り文字の直前の文字、または引用符で囲まれた文字列内における引用符の直前の文字。エスケープ文字を指定すると、データ統合サービスは、区切り文字を普通の文字として読み込みます。
次の行からインポートを開始	データ統合サービスがファイルをインポートする際に、読み込みを開始する行番号。例えば 2 行目から開始するように指定すると、1 行目をスキップしてから読み込みを開始します。
連続した区切り文字を 1 文字として扱う	選択した場合、データ統合サービスは 1 つ以上の連続したカラム区切り文字を 1 つのカラム区切り文字として読み取ります。このオプションを選択しない場合、データ統合サービスは 2 つの連続した区切り文字を 1 つの NULL 値として読み取ります。
データ内のエスケープ文字を維持	出力文字列にエスケープ文字を含めます。

- 固定長フラットファイルの場合は、次のプロパティを設定します。

プロパティ	説明
1 行目からカラム名をインポート	選択すると、Developer tool は最初の行のデータをカラム名として使用します。カラム名が最初の行に含まれている場合に、このオプションを選択します。
次の行からインポートを開始	データ統合サービスがファイルをインポートする際に、読み込みを開始する行番号。例えば 2 行目から開始するように指定すると、1 行目をスキップしてから読み込みを開始します。
カラムの区切り	固定長ファイルのカラムの区切りを設定します。【区切りの編集】をクリックしてカラムの境界線を編集します。または、表示される指示に従い、ファイルのプレビューウィンドウでカラム間の境界線を操作します。カラムの境界線はドラッグして移動できます。境界線を削除するには、これをダブルクリックします。

15. **【次へ】** をクリックして、フラットファイルデータオブジェクトをプレビューします。
16. **【完了】** をクリックします。
データオブジェクトがエディタに表示されます。

制御ファイルからのフラットファイルデータオブジェクトの作成

制御ファイルに基づいてデータオブジェクトのカラムと行を定義する場合は、制御ファイルからフラットファイルデータオブジェクトを作成します。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】** > **【新規】** > **【データオブジェクト】** をクリックします。
3. **【物理データオブジェクト】** > **【フラットファイルデータオブジェクト】** を選択し、**【次へ】** をクリックします。
【新しいフラットファイルデータオブジェクト】 ダイアログボックスが表示されます。
4. **【制御ファイルから作成】** を選択します。
5. **【参照】** をクリックし、制御ファイルが含まれるディレクトリに移動します。
6. **【開く】** をクリックします。
選択した制御ファイルと同じ名前がデータオブジェクトに付けられます。
7. 必要に応じて、データオブジェクト名を編集します。
8. 必要に応じて、**【参照】** ボタンをクリックして、データオブジェクトのプロジェクトまたはフォルダを選択します。
9. **【次へ】** をクリックします。
10. 処理するデータのコードページに一致するコードページを選択します。
11. **【区切り】** または **【固定長】** を選択します。
12. **【固定長】** を選択した場合、**【完了】** をクリックします。**【区切り】** を選択し、**【次へ】** をクリックします。
13. 次の区切り形式のプロパティを設定します。

プロパティ	説明
区切り文字	データのカラムを区切るために使用される文字。エスケープ文字またはテキスト修飾子と同じ区切り文字を入力すると、予期しない結果が生成されることがあります。
引用符	引用符はテキスト文字列の境界を定義します。引用符を選択すると、引用符で囲まれた文字列内の区切り文字が無視されます。

14. **【完了】** をクリックします。
データオブジェクトがエディタに表示されます。

第 7 章

データの論理ビュー

この章では、以下の項目について説明します。

- [データの論理ビューの概要, 117 ページ](#)
- [データの論理ビューの開発, 119 ページ](#)
- [論理データオブジェクトモデル, 119 ページ](#)
- [論理データオブジェクトモデルのプロパティ, 121 ページ](#)
- [論理データオブジェクト, 127 ページ](#)
- [論理データオブジェクトマッピング, 131 ページ](#)

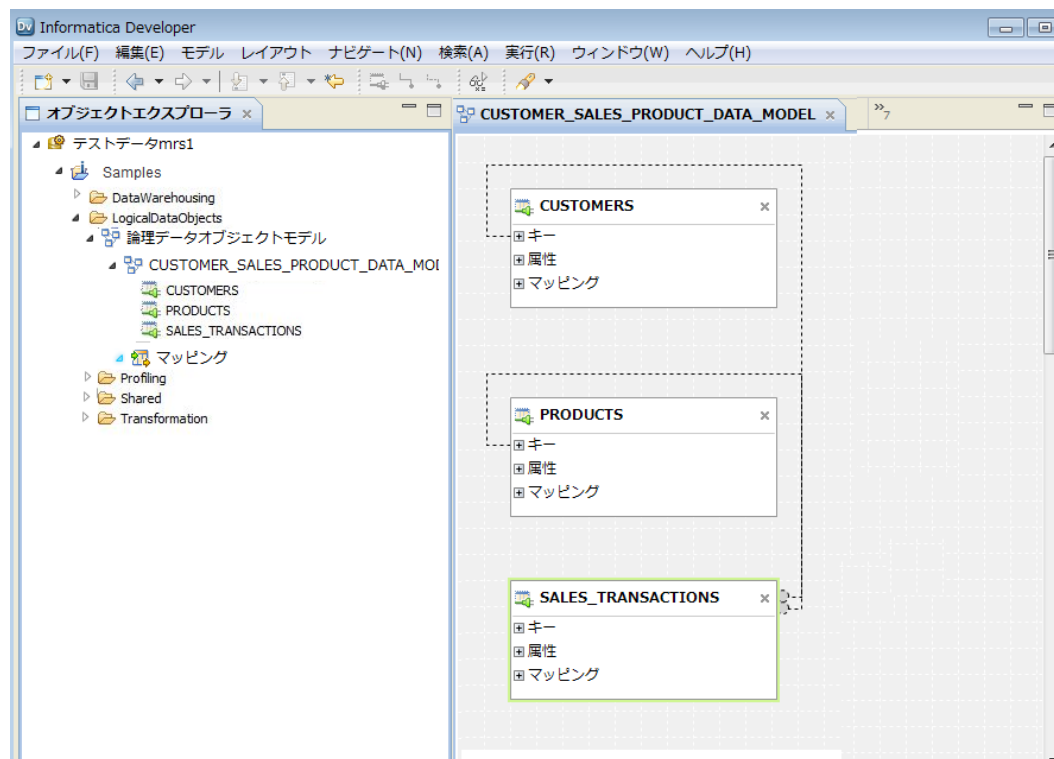
データの論理ビューの概要

データの論理ビューは、企業内に存在するデータを表すビューです。データの論理ビューには、論理データモデル、論理データオブジェクト、および論理データオブジェクトマッピングが含まれます。

データの論理ビューは、以下の目的で使します。

- 個々のビジネスニーズに対応しようとしてデータを定義しなおす手間を省くため、企業全体で共通のデータモデルを使用する。これは、データ属性に変更があったときに変更の適用が 1 回で済むことを意味しています。つまり、1 つのマッピングを使用して、このデータを使用するすべてのデータベースに変更を反映できるのです。
- 関連するデータのソースを見つけて、単一のビューに表示する。データは企業内のさまざまな場所（リレーショナルデータベースやフラットファイルなど）に存在しますが、1 つのビューからすべてのデータソースにアクセスして表示することができます。
- 論理データをリレーショナルテーブルとして公開し、再利用を促進する。

以下の図に、関連する論理データオブジェクト例を示します。



論理データオブジェクトモデルの例

論理データオブジェクトモデルを作成して、企業内の論理エンティティを示します。例えば、論理データオブジェクトモデルを作成して、異なるソースからのアカウントデータを単一のビューに表示します。

American Bank が California Bank を買収しました。買収後、American Bank は以下の操作が必要になりました。

- 両行のデータを、上位 10 位の顧客レポートなどのビジネスインテリジェンスレポートに表示する。
- 両行のデータを中央のデータウェアハウスに統合する。

従来、American Bank は、開発環境でデータを中央のデータウェアハウスに統合し、データを検証して、そのデータウェアハウスをプロダクション環境に移動していました。このプロセスには数か月以上を要していました。また、American Bank では、ビジネスインテリジェンスレポートはプロダクション環境のデータウェアハウスに対して実行していました。

American Bank の開発者は Developer ツールを使用して、顧客、口座、支店、およびその他の企業内データのモデルを作成できます。American Bank と California Bank のリレーショナルソースを、顧客の単一ビューにリンクします。その結果、中央のデータウェアハウスを作成する前に、ビジネスインテリジェンスレポートに使用可能なデータを作成できます。

データの論理ビューの開発

企業におけるデータへのアクセス方法とデータの使用方法を表す、データの論理ビューを開発します。

データの論理ビューを開発した後、データサービスに追加して、仮想データをエンドユーザーが使えるようにします。

データの論理ビューを開発する前に、論理データオブジェクトマッピングで使用する物理データオブジェクトを定義できます。データ品質を分析するために物理データソースをプロファイルすることもできます。

1. 論理データモデルを作成またはインポートします。
2. 必要に応じて、論理データオブジェクトを論理データオブジェクトモデルに追加して、オブジェクト間のリレーションを定義します。
3. 論理データオブジェクトに対してデータの読み取りまたは書き込みを行うための論理データオブジェクトマッピングを作成します。論理データオブジェクトマッピングには、データを変換するためのトランスフォーマーメソッドロジックを含めることができます。このトランスフォーメーションには、データを検証およびクレンジングするための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。
4. 論理データオブジェクトマッピングの出力を表示します。

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は、企業内のデータの構造および使用方法を表します。論理データオブジェクトモデルは論理データオブジェクトを含み、オブジェクト間のリレーションを定義します。

論理データオブジェクトモデルを定義して、企業内のデータの統合モデルを作成します。企業のデータは、リレーショナルデータベースやフラットファイルなど、複数の異なるソースシステム内に存在します。論理データオブジェクトモデルは、ソースシステムにかかわらず、ビジネスの観点からデータを表します。論理データオブジェクトモデルを作成して、データの検証、データ属性の記述、および属性間のリレーションの定義を行います。

例えば、American Bank の顧客口座データが Oracle データベースに存在し、California Bank のデータが IBM DB2 データベースに存在する場合に、顧客と口座のリレーションを定義する、顧客口座の統合モデルを作成するには、このリレーションを定義する論理データオブジェクトモデルを作成します。

モデリングツールから論理データオブジェクトモデルをインポートすることができます。モデリングツールで作成した XSD ファイルから論理データオブジェクトモデルをインポートすることもできます。または、Developer ツールで論理データオブジェクトモデルを手動で作成することもできます。

論理データオブジェクトモデルはプロジェクトまたはフォルダーに追加して、モデルリポジトリ内に格納します。

エンドユーザーが論理データオブジェクトに対して SQL クエリを実行できるようにするには、論理データオブジェクトを SQL データサービスに含めます。論理データオブジェクトを仮想テーブルのソースにします。エンドユーザーが Web 経由で論理データオブジェクトにアクセスできるようにするには、論理データオブジェクトを Web サービスに含めます。論理データオブジェクトを操作のソースにします。

論理データオブジェクトモデルの作成

論理データオブジェクトモデルを作成して、企業内のデータの構造および使用方法を定義します。論理データオブジェクトモデルを作成する際には、論理データオブジェクトを追加できます。物理データオブジェクトを

各論理データオブジェクトに関連付けます。Developer ツールで、モデルの各論理データオブジェクトに対する論理データオブジェクト読み取りマッピングを作成します。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】 > 【新規】 > 【論理データオブジェクトモデル】** をクリックします。
【新規】 ダイアログボックスが表示されます。
3. **【論理データオブジェクトモデル】** を選択し、**【次へ】** をクリックします。
【新しい論理データオブジェクトモデル】 ダイアログボックスが表示されます。
4. 論理データオブジェクトモデルの名前を入力します。
5. 論理データオブジェクトを作成するには、**【次へ】** をクリックします。空の論理データオブジェクトモデルを作成するには、**【完了】** をクリックします。
【次へ】 をクリックすると、論理データオブジェクトをモデルに追加するように求められます。
6. 論理データオブジェクトを作成するには、**【新規】** ボタンをクリックします。
論理データオブジェクトがリストに追加されます。
7. **【名前】** カラムに名前を入力します。
8. 必要に応じて、**【データオブジェクト】** カラムの **【オープン】** ボタンをクリックして、物理データオブジェクトを論理データオブジェクトに関連付けます。
【データオブジェクトの選択】 ダイアログボックスが表示されます。
9. 物理データオブジェクトを選択し、**【OK】** をクリックします。
10. 手順 6～9 を繰り返して、論理データオブジェクトを追加します。
11. **【完了】** をクリックします。
論理データオブジェクトモデルがエディタに表示されます。

モデリングツールからの論理データオブジェクトモデルのインポート

モデリングツールまたは XSD ファイルから論理データオブジェクトモデルをインポートすることができます。企業内の既存の構造のモデルおよびデータを利用するには、論理データオブジェクトモデルをインポートします。

1. 論理データオブジェクトモデルのインポート先の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】 > 【論理データオブジェクトモデル】** をクリックします。
【新しい論理データオブジェクトモデル】 ダイアログボックスが表示されます。
3. **【データモデルから論理データオブジェクトモデル】** を選択します。
4. **【次へ】** をクリックします。
5. **【モデルタイプ】** フィールドで、論理データオブジェクトモデルのインポート元のモデリングツールを選択します。
6. 論理データオブジェクトモデルの名前を入力します。
7. **【参照】** をクリックして、論理データオブジェクトモデルの場所を選択します。
8. **【次へ】** をクリックします。
9. インポートするファイルを参照し、選択して **【オープン】** をクリックします。
10. インポートプロパティを設定します。
11. **【次へ】** をクリックします。

12. 論理データオブジェクトを論理データオブジェクトモデルに追加します。
13. **【完了】** をクリックします。
論理データオブジェクトがエディタに表示されます。

論理データオブジェクトモデルのプロパティ

論理データオブジェクトモデルをモデリングツールからインポートする場合、このツールに関連付けられたプロパティを指定します。

CA ERwin Data Modeler のインポートプロパティ

論理データオブジェクトモデルを CA ERwin Data Modeler からインポートする場合、インポートプロパティを設定します。

以下の表に、モデルを CA ERwin Data Modeler からインポートする場合に設定するプロパティを示します。

プロパティ	説明
UDP のインポート	ユーザー定義プロパティのインポート方法を指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- メタデータとして: 明示的な値をプロパティ値オブジェクトとしてインポートします。明示的な値はエクスポートされません。- メタデータとして既定値を移行: 明示的な値および黙示的な値をプロパティ値オブジェクトとしてインポートします。- 説明に既定値を移行: 黙示的な場合にも、プロパティ名と値をオブジェクトの説明プロパティに追加します。- 両方で既定値を移行: 黙示的な場合にも、UDP 値をメタデータとして、またオブジェクトの説明にインポートします。 デフォルトは [メタデータとして] です。
リレーション名のインポート	ERwin からのリレーション名のインポート方法を指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- リレーション名から- リレーション説明から デフォルトは [リレーション名から] です。
ID のインポート	オブジェクトの一意の ID をネイティブ ID プロパティとして設定するかどうかを指定します。
サブジェクト領域のインポート	ERwin からのサブジェクト領域のインポート方法を指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- 図としてインポート- パッケージおよび図としてインポート- パッケージおよび図としてインポート、各エンティティに対して 1 つのサブジェクト領域を想定しています- サブジェクト領域をインポートしない デフォルトは [図としてインポート] です。

プロパティ	説明
カラム順序のインポート形式	<p>テーブル内のカラムの位置のインポート方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - カラム順序: ERwin 物理ビューに表示されるカラムの順序。 - 物理的順序: SQL DDL で生成されたときの、データベース内のカラムの順序。 <p>デフォルトは [物理的順序] です。</p>
所有者スキーマのインポート	所有者スキーマをインポートするかどうかを指定します。

IBM Cognos Business Intelligence Reporting - Framework Manager のインポートプロパティ

IBM Cognos Business Intelligence Reporting - Framework Manager から論理データオブジェクトモデルをインポートする場合、インポートプロパティを設定します。

以下の表に、モデルを IBM Cognos Business Intelligence Reporting - Framework Manager からインポートする場合に設定するプロパティを示します。

プロパティ	説明
フォルダの表示	<p>Framework Manager のフォルダの表示方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 無視: フォルダを無視します。 - フラット: フォルダを図として表しますが、階層は維持しません。 - 階層: フォルダを図として表し、階層も維持します。 <p>デフォルトは [無視] です。</p>
パッケージの表示	<p>Cognos Framework Manager のパッケージの表示方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 無視: サブジェクト領域を無視します。 - サブジェクト領域: パッケージをサブジェクト領域として表します。 - モデル: パッケージをモデルとして表します。 <p>デフォルトは [無視] です。</p>
リバースエンジニアリングのリレーション	2 つの dbQueries 間のリレーションを参照整合性制約として計算するかどうかを指定します。
テーブルデザインレベル	<p>インポートしたテーブルのデザインレベルの制御方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 論理および物理: テーブルは、モデルの論理ビューと物理ビューの両方で表示されます。 - 物理: テーブルはモデルの物理ビューで表示されます。 <p>デフォルトは [物理] です。</p>
使用方法プロパティを無視	queryItem の使用方法プロパティを使用するかどうかを指定します。

SAP BusinessObjects Designer のインポートプロパティ

論理データオブジェクトモデルを SAP BusinessObjects Designer からインポートする場合、インポートプロパティを設定します。

以下の表に、モデルを SAP BusinessObjects Designer からインポートする場合に設定するプロパティを示します。

プロパティ	説明
System	BusinessObjects リポジトリの名前。 BusinessObjects versions 11.x および 12.x (XI)の場合、Central Management Server の名前を入力します。 BusinessObjects version 5.x および 6.x の場合、Supervisor アプリケーションによって定義されたリポジトリの名前を入力します。
認証モード	ログイン認証モード。 このパラメータは、SAP BusinessObjects Designer 11.0 以降に適用されます。 次のいずれかの認証モードを選択します。 <ul style="list-style-type: none">- エンタープライズ: Business Objects Enterprise ログイン- LDAP: LDAP サーバー認証- Windows AD: Windows Active Directory サーバー認証- Windows NT: Windows NT ドメインサーバー認証- スタンドアロン: スタンドアロン認証 デフォルトは [エンタープライズ] です。
ユーザー名	BusinessObjects サーバーのユーザー名。 バージョン 11.x および 12.x (XI)の場合、BusinessObjects グループのメンバーであることが必要です。
パスワード	BusinessObjects サーバーのパスワード。
サイレント実行	対話式モードで実行するか、サイレントモードで実行するかを指定します。 デフォルトは [サイレント] です。
実行後に閉じる	Developer Tool がモデルのインポートを完了した後に BusinessObjects を閉じるかどうかを指定します。
テーブルデザインレベル	インポートしたテーブルのデザインレベルを指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- 論理および物理: テーブルは、モデルの論理ビューと物理ビューの両方で表示されます。- 物理: テーブルはモデルの物理ビューで表示されます。 デフォルトは [物理] です。
結合を外部キーに変換	モデル内の単純な SQL 結合を外部キーリレーションに変換します。 データベースデザインツールなど、構造化リレーショナルメタデータのみをサポートするツールにモデルをエクスポートする場合、このパラメータを選択します。

プロパティ	説明
クラス表示	<p>クラスおよびサブクラスのツリー構造をインポートする方法を指定します。Developer ツールは、CWM OLAP 標準の定義どおりに各クラスをディメンションとしてインポートします。また、Developer ツールは、クラスおよびサブクラスを CWM 標準および UML 標準の定義どおりにパッケージのツリーとしてインポートします。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - フラット構造: Developer ツールはパッケージを作成しません。 - 単純ツリー構造: Developer ツールは、サブクラスを持つ各クラスのパッケージを作成します。 - 完全ツリー構造: Developer ツールは、各クラスのパッケージを作成します。 <p>デフォルトは [フラット構造] です。</p>
値のリストを含める	<p>オブジェクトに関連付けられている値のリストをインポートする方法を制御します。</p>
次元プロパティのトランスフォーメーション	<p>次元名、説明、およびロールを、基本テーブルに変換する方法、および属性名、説明、データ型を基本カラムに変換する方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 無効: プロパティ変換は発生しません。 - 有効: 次元オブジェクトとリレーションオブジェクト間に直接一致がある場合にプロパティ変換が発生します。Developer ツールは、次元名をリレーション名に移行します。 - 有効 (名前を保持): 次元オブジェクトとリレーションオブジェクト間に直接一致がある場合にプロパティ変換が発生します。Developer ツールはリレーション名を保持します。 <p>デフォルトは [無効] です。</p>

SAP PowerDesigner CDM のインポートプロパティ

論理データオブジェクトモデルを SAP PowerDesigner CDM からインポートする場合、インポートプロパティを設定します。

次の表に、SAP PowerDesigner CDM からモデルをインポートする場合に設定するプロパティを示します。

プロパティ	説明
UDP のインポート	<p>ユーザー定義プロパティのインポート方法を指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - メタデータとして: 明示的な値をプロパティ値オブジェクトとしてインポートします。明示的な値はエクスポートされません。 - メタデータとして既定値を移行: 明示的な値および黙示的な値をプロパティ値オブジェクトとしてインポートします。 - 説明に既定値を移行: 黙示的な場合にも、プロパティ名と値をオブジェクトの説明プロパティに追加します。 - 両方で既定値を移行: 黙示的な場合にも、UDP 値をメタデータとして、またオブジェクトの説明にインポートします。 <p>デフォルトは [メタデータとして] です。</p>
関連クラスのインポート	<p>関連クラスをインポートするかどうかを指定します。</p>
ID のインポート	<p>オブジェクトの一意の ID をネイティブ ID プロパティとして設定するかどうかを指定します。</p>

プロパティ	説明
説明フィールドに容量情報を追加する	オカレンス数の情報をインポートして説明プロパティに追加します。
テキストの書式の削除	リッチテキスト書式を削除するか保持するかを指定します。 モデルが PowerDesigner 7.0 または 7.5 で生成された場合は、このオプションを選択します。 モデルが PowerDesigner 8.0 以上で生成された場合は、このオプションをクリアします。

SAP PowerDesigner PDM のインポートプロパティ

SAP PowerDesigner PDM から論理データオブジェクトモデルをインポートする場合、インポートプロパティを設定します。

次の表に、SAP PowerDesigner PDM からモデルをインポートする場合に設定するプロパティを示します。

プロパティ	説明
UDP のインポート	ユーザー定義プロパティのインポート方法を指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none"> - メタデータとして: 明示的な値をプロパティ値オブジェクトとしてインポートします。明示的な値はエクスポートされません。 - メタデータとして既定値を移行: 明示的な値および黙示的な値をプロパティ値オブジェクトとしてインポートします。 - 説明に既定値を移行: 黙示的な場合にも、プロパティ名と値をオブジェクトの説明プロパティに追加します。 - 両方で既定値を移行: 黙示的な場合にも、UDP 値をメタデータとして、またオブジェクトの説明にインポートします。 デフォルトは「メタデータとして」です。
ID のインポート	オブジェクトの一意の ID をネイティブ ID プロパティとして設定するかどうかを指定します。
説明フィールドに容量情報を追加する	オカレンス数の情報をインポートして説明プロパティに追加します。
テキストの書式の削除	リッチテキスト書式を削除するか保持するかを指定します。 モデルが PowerDesigner 7.0 または 7.5 で生成された場合は、このオプションを選択します。 モデルが PowerDesigner 8.0 以上で生成された場合は、このオプションをクリアします。

XSD のインポートプロパティ

モデリングツールでエクスポートした XSD ファイルから論理データオブジェクトモデルをインポートすることができます。

以下の表に、モデルを XSD ファイルからインポートする場合に設定するプロパティを示します。

プロパティ	説明
要素のコンテンツ名	XSD ファイル内の#PCDATA などのテキストコンテンツを保持するための属性。 デフォルトは [メタデータとして] です。
縮小レベル	クラスを縮小する状況を指定します。選択した値によって、XSD ファイルの要素と属性のすべてがインポートされるか、または一部がインポートされるかが決まります。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- なし: すべての XSD 要素はクラスになり、すべての XSD 属性は属性になります。- 空: 空のクラスのみが、親クラスに縮小されます。- 単一属性: 属性が 1 つで、子がない XSD 要素のみが親クラスに縮小されます。- 子なし: 子がない XSD 要素のみが親クラスに縮小されます。- すべて: すべての縮小可能な XSD 要素が親クラスに縮小されます。 デフォルトは [すべて] です。
星印を縮小	入力 xlink を持つ XML 要素を親クラスに縮小するかどうかを指定します。
クラスタイプ	要素を親要素に縮小する場合にクラスタイプを作成するかどうかを指定します。
Any	「xs:any」疑似要素に対するクラスまたはエンティティを作成するかどうかを指定します。
ID の生成	プライマリキーと外部キーを作成するための追加の属性を生成するかどうかを指定します。デフォルトでは、追加の属性は生成されません。
置換グループのインポート方法	継承の表示方法を指定します。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- 汎化: 継承を汎化として表します。- ロールダウン: サブクラスの継承された属性を複製します。 デフォルトは [ロールダウン] です。
インクルードパス	インクルードされたスキーマファイルを含むディレクトリのパス（該当する場合）。
UDP 名前空間	ユーザー定義プロパティとしてインポートされる属性を含む名前空間。

論理データオブジェクト

論理データオブジェクトは論理データオブジェクトモデル内のオブジェクトであり、企業の論理エンティティを表します。論理データオブジェクトには属性とキーがあり、属性間のリレーションを表します。

相互に関係する論理データオブジェクトをデータオブジェクトモデルに含めます。例えば、国立銀行の論理データオブジェクトモデルに、論理データオブジェクト「Customer（顧客）」と「Account（口座）」があるとします。論理データオブジェクトモデルは、顧客と口座のリレーションを表します。

このモデルでは、論理データオブジェクト Account は、属性 Account_Number（口座番号）を含みます。口座を一意に識別する Account_Number がプライマリキーです。Customer データオブジェクトは各顧客の口座を参照する必要があるため、Account には論理データオブジェクト Customer とのリレーションがあります。

物理データオブジェクトを論理データオブジェクトモデルエディタにドラッグして、論理データオブジェクトを作成できます。または、論理データオブジェクトを作成して、属性とキーを定義します。

論理データオブジェクトのプロパティ

論理データオブジェクトには、データオブジェクトを定義するプロパティと、論理データオブジェクトモデル内の他の論理データオブジェクトへのリレーションが含まれます。

論理データオブジェクトのプロパティは、エディタ内のエディタタブにまとめられています。以下の図に、論理データオブジェクトエディタを示します。

The screenshot shows the 'Department' logical data object editor. It has a tabbed interface with 'Overview' selected. The 'General' tab contains fields for 'Name' (Department) and 'Description'. Below these are 'Read Mapping' and 'Write Mapping' sections, each with a text input and 'Add...' and 'Delete' buttons. A small preview table on the right shows the attributes: DEPTNO (decimal), DNAME (string), LOC (string), FIRSTNAME (string), and LASTNAME (string). The 'Attributes' tab is also visible, showing a table with columns: Name, Type, Primary Key, Precision, Scale, Nullable, and Description. The table lists the five attributes with their respective values and checkboxes for primary key and nullable status.

	Name	Type	Primary Key	Precision	Scale	Nullable	Description
1	DEPTNO	decimal	<input type="checkbox"/>	2	0	<input checked="" type="checkbox"/>	
2	DNAME	string	<input type="checkbox"/>	10	0	<input type="checkbox"/>	
3	LOC	string	<input type="checkbox"/>	8	0	<input checked="" type="checkbox"/>	
4	FIRSTNAME	string	<input type="checkbox"/>	20	0	<input type="checkbox"/>	
5	LASTNAME	string	<input type="checkbox"/>	20	0	<input checked="" type="checkbox"/>	

Navigation tabs at the bottom: Overview | Keys | Relationships | Access | Read Mapping | Advanced

次の表では、論理データオブジェクトエディタに表示される情報について説明します。

タブ名	説明
概要	[全般] 領域には、論理データオブジェクトの名前、説明、読み取りおよび書き込みマッピング（該当する場合）が含まれます。 [属性] 領域には、論理データオブジェクト内のデータの構造が表示されます。
キー	論理データオブジェクト内の 1 つ以上の属性をプライマリキーまたは一意キーに指定できます。
リレーション	論理データオブジェクト間の関連付け。
アクセス	論理データオブジェクトとデータオブジェクトの各属性に対するアクセスの種類。
読み取りマッピング	論理データオブジェクトに関連付けられている論理データオブジェクト読み取りマッピング。論理データオブジェクトに読み取りマッピングが含まれていない場合は、[読み取りマッピング] タブは表示されません。
書き込みマッピング	論理データオブジェクトに関連付けられている論理データオブジェクト書き込みマッピング。論理データオブジェクトに書き込みマッピングが含まれていない場合は、[書き込みマッピング] タブは表示されません。

属性のリレーション

リレーションは、1 つ以上の論理データオブジェクトのプライマリキー属性または外部キー属性間の関連付けです。

属性間には以下のタイプのリレーションを定義できます。

識別

ある属性が他の属性との関連付けによって識別される、2 つの属性間のリレーション。

例えば、論理データオブジェクト Branch（支店）の Branch_ID（支店 ID）属性と、論理データオブジェクト Customer（顧客）の Branch_Location（支店の場所）属性間のリレーションは「識別」です。支店 ID は、支店の場所に対して一意だからです。

非識別

一方の属性を他方の属性に依存せずに識別する、2 つの属性間のリレーション。

例えば、論理データオブジェクト Account（口座）の Account_Type（口座種別）属性と、論理データオブジェクト Customer（顧客）の Account_Number（口座番号）属性間のリレーションは「非識別」です。口座種別は、口座番号と関連付けなくても識別可能です。

リレーションを定義する際、論理データオブジェクトモデルでは、識別リレーションが実線で表されます。非識別リレーションは属性間の点線で表されます。

論理データオブジェクトの作成

論理データオブジェクトを論理データオブジェクトモデル内に作成し、企業の論理エンティティを定義できます。

1. [ファイル] > [新規] > [論理データオブジェクト] をクリックします。
2. 論理データオブジェクト名を入力します。

- 論理データオブジェクトの論理データオブジェクトモデルを選択し、**【完了】** をクリックします。
論理データオブジェクトが論理データオブジェクトモデルエディタに表示されます。
- 論理データオブジェクトを選択し、**【プロパティ】** ビューをクリックします。
- [全般] 領域の **【概要】** タブで、必要に応じて論理データオブジェクトの名前と説明を編集できます。
- [属性] 領域の **【概要】** タブで、属性を作成し、そのデータ型と精度を指定できます。

ツールバーのオプションを使用して、属性値を更新できます。使用可能なオプションは、[新規]、[切り取り]、[コピー]、[貼り付け]、[削除]、[上に移動]、[下に移動]、[すべての NULL 可能を有効化]、および [すべての NULL 可能を無効化] です。[すべての NULL 可能を有効化] または [すべての NULL 可能を無効化] アイコンを使用して、属性のすべての NULL 可能値を選択またはクリアできます。

Department

Overview

General

Name:

Description:
Read Mapping: [Department Read Mapping](#) Add... Delete

Write Mapping: Add... Delete

Department

Name	Type
DEPTNO	decimal
DNAME	string
LOC	string
FIRSTNAME	string
LASTNAME	string

Attributes

	Name	Type	Primary Key	Precision	Scale	Nullable	Description >>
1	DEPTNO	decimal	<input type="checkbox"/>	2	0	<input checked="" type="checkbox"/>	
2	DNAME	string	<input type="checkbox"/>	10	0	<input type="checkbox"/>	
3	LOC	string	<input type="checkbox"/>	8	0	<input checked="" type="checkbox"/>	
4	FIRSTNAME	string	<input type="checkbox"/>	20	0	<input type="checkbox"/>	
5	LASTNAME	string	<input type="checkbox"/>	20	0	<input checked="" type="checkbox"/>	

Overview | **Keys** | Relationships | Access | Read Mapping | Advanced

7. **【キー】** タブで、必要に応じてデータオブジェクトのプライマリキーと一意キーを指定します。

Keys

Keys: Key

Add... **Remove**

Name: Key

Description:

Available Attributes:

Status
LastName
FirstName

Selected Attributes:

CUSTID

> <

Overview Keys Relationships Access Read Mapping Advanced

8. **【リレーション】** タブで、必要に応じて論理データオブジェクト間のリレーションを作成します。

Relationships

Relationships: Relationship

Add... **Remove**

Name: Relationship

Description:

Relationship Type

☒ Identifying
☐ Non-Identifying

Referenced Key

Key: [customer.Key](#) **Browse...**

Attributes:

CUSTID

Logical Data Object

Available Attributes:

CUSTID
Status
LastName
FirstName

Selected Attributes:

CUSTID1

> <

Overview Keys Relationships Access Read Mapping Advanced

9. **【アクセス】** タブで、必要に応じて論理データオブジェクトとデータオブジェクトの各属性のアクセスの種類を編集します。

デフォルトは「読み取り専用」です。

Access

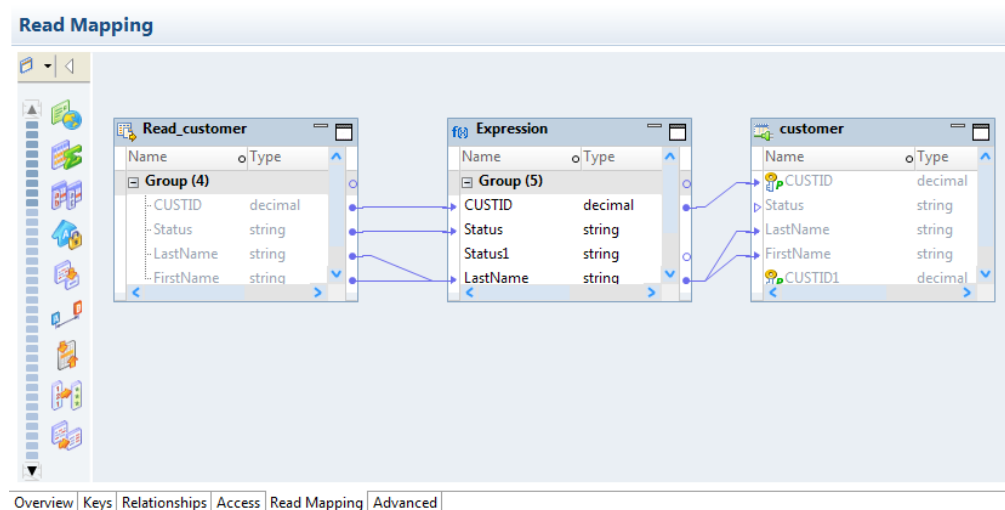
Write Access: Read Only

Attributes:

	Name	Type	Write Access
1	CUSTID	decimal	Read Only
2	Status	string	Read Only
3	LastName	string	Read Only
4	FirstName	string	Read Only
5	CUSTID1	decimal	Read Only

Overview | Keys | Relationships | Access | Read Mapping | Advanced

10. 「読み取りマッピング」タブで、必要に応じて、論理データオブジェクト読み取りマッピングを作成します。



11. 「書き込みマッピング」タブで、必要に応じて、論理データオブジェクト書き込みマッピングを作成します。
12. 論理データオブジェクトを保存します。

論理データオブジェクトマッピング

論理データオブジェクトマッピングは、論理データオブジェクトを1つ以上の物理データオブジェクトにリンクするマッピングです。論理データオブジェクトマッピングには、トランスフォーメーションロジックを含めることができます。

論理データオブジェクトマッピングのタイプは以下のいずれかです。

- 読み取り
- 書き込み

各論理データオブジェクトを、1つの論理データオブジェクト読み取りマッピングまたは1つの論理データオブジェクト書き込みマッピングに関連付けることができます。

論理データオブジェクト読み取りマッピング

論理データオブジェクト読み取りマッピングには、入力として1つ以上の物理データオブジェクトが、また出力として1つの論理データオブジェクトが含まれます。マッピングには、データを変換するためのトランスフォーマーメソッドロジックを含めることができます。

論理データオブジェクト読み取りマッピングは、基のデータソースにアクセスせずに、データにアクセスする方法を提供します。また、複数のソースのデータを1つのビューにまとめて表示することを可能にします。

例えば、American Bankに顧客の口座の論理データオブジェクトモデルがあり、この論理データオブジェクトモデルにはCustomers（顧客）論理データオブジェクトが含まれているとします。

Customers 論理データオブジェクトにある2つのリレーショナルデータベースから顧客データを表示する場合、論理データオブジェクト読み取りマッピングを使用してこのタスクを実行し、出力を【データビュー】ビューで表示することができます。

論理データオブジェクト書き込みマッピング

論理データオブジェクト書き込みマッピングは、入力として論理データオブジェクトを含みます。このマッピングにより、論理データオブジェクトからターゲットへの書き込みが可能になります。

マッピングには、データを変換するためのトランスフォーマーメソッドロジックを含めることができます。このマッピングは基になるデータターゲットにアクセスすることなく実行されます。ターゲットへ書き込むことなく、変換されたデータの単一ビューを提供します。

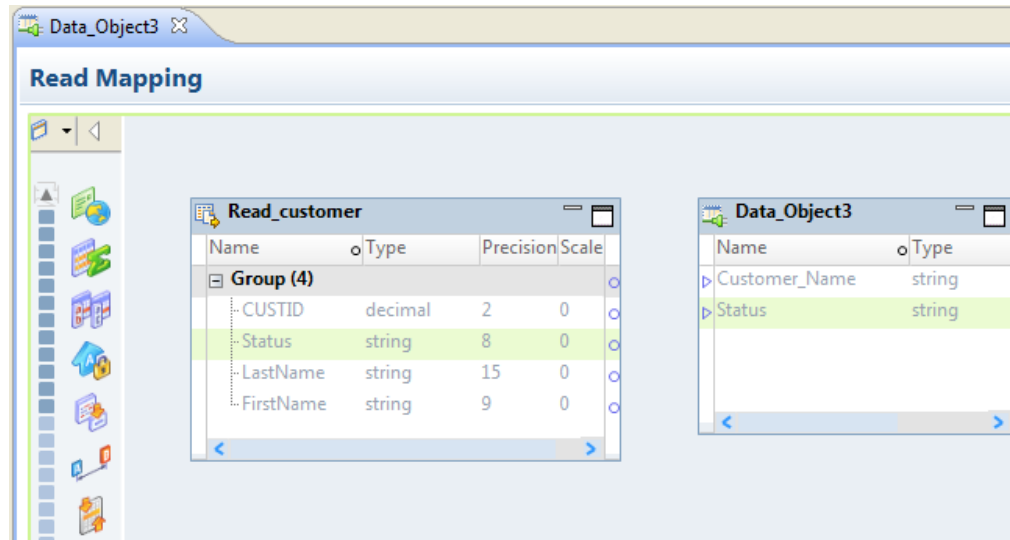
論理データオブジェクトマッピングの作成

論理データオブジェクトマッピングを作成して、物理データオブジェクトから論理データオブジェクトにデータをリンクし、データを変換することができます。

1. 【オブジェクトエクスプローラ】ビューで、マッピングを物理データオブジェクトに追加する論理データオブジェクトをダブルクリックします。
論理データオブジェクトエディタが開きます。
2. [全般] 領域の【概要】タブで、【追加】をクリックして読み取りマッピングまたは書き込みマッピングを追加します。
3. マッピングの名前を入力して【完了】をクリックします。
マッピングが読み取りマッピングか書き込みマッピングかに応じて、論理データオブジェクトがマッピング入力または出力としてエディタに表示されます。
4. 物理データオブジェクトから論理データオブジェクトにデータをリンクします。
 - a. 【読み取りマッピング】タブまたは【書き込みマッピング】タブをクリックしてマッピングを編集します。
マッピングエディタパレットが開きます。
 - b. 【オブジェクトエクスプローラ】ビューで、リンクする物理データオブジェクトを参照し、物理データオブジェクトをマッピングエディタパレットにドラッグします。
【マッピングに追加】ダイアログボックスが開きます。
 - c. 読み取りマッピング、書き込みマッピング、またはルックアップマッピングの作成を選択します。

- d. 読み取りマッピングの作成を選択する場合は、読み取りマッピングのオブジェクトアクセスを関連データオブジェクトまたは独立したデータオブジェクトとして指定します。
- e. **【OK】** をクリックします。

以下の画像は、**【顧客】** フラットファイルデータオブジェクトをマッピングエディタパレットにドラッグし、読み取りマッピングとして指定した後の状態です。



5. 必要に応じて、マッピングに再利用可能なオブジェクトを追加します。
論理データオブジェクトまたは他のモデルリポジトリオブジェクトを追加できます。
 - a. マッピングエディタを右クリックして **【再利用可能なオブジェクトの追加】** を選択します。
 - b. モデルリポジトリからオブジェクトを 1 つ選択して **【OK】** をクリックします。
 - c. 再利用可能なオブジェクトを読み取りマッピング、書き込みマッピング、またはルックアップマッピングとして指定します。
 - d. 再利用可能なオブジェクトを読み取りマッピングとして指定する場合は、読み取りマッピングのオブジェクトアクセスを関連データオブジェクトまたは独立したデータオブジェクトとして指定します。
 - e. **【OK】** をクリックします。
6. 必要に応じて、追加オブジェクトとトランスフォーメーションをマッピングに追加し、マッピングオブジェクト間にリンクを作成してから、**【データビューア】** ビューをクリックしてマッピングを実行します。
結果が **【出力】** セクションに表示されます。

第 8 章

データの表示

この章では、以下の項目について説明します。

- [データの表示の概要, 134 ページ](#)
- [構成, 135 ページ](#)
- [データのエクスポート, 143 ページ](#)
- [オブジェクトの依存関係, 143 ページ](#)
- [ログ, 146 ページ](#)
- [検証設定, 146 ページ](#)
- [Developer tool からのジョブの監視, 147 ページ](#)

データの表示の概要

ユーザーは、マッピングの実行、プロファイル結果の表示、ソースデータの表示、トランスフォーメーションのデータのプレビュー、SQL クエリの実行、Web サービスのメッセージのプレビュー、またはオブジェクトに関する依存関係の表示を行うことができます。

データとマッピングを操作および表示するには、次のビューを使用します。

【マッピングエディタ】 ビュー

マッピングエディタを使用して、マッピングを実行し、プロファイル結果を表示します。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。コマンドラインまたは【実行】ダイアログボックスからマッピングを実行できます。

【データビューア】 ビュー

【データビューア】 ビューを使用して、ソースデータの表示、トランスフォーメーションのデータのプレビュー、SQL クエリの実行、Web サービスのメッセージのプレビューを行います。

データを表示する前に、ドメインに複数のサービスが含まれている場合は、デフォルトのデータ統合サービスを選択する必要があります。他のデータ統合サービスを追加して、データを表示するときに使用することもできます。設定を作成して、データを表示するときに Developer ツールで適用される設定を制御できます。

データ統合サービスまたは Spark エンジンのいずれかで実行されるマッピング内のデータをプレビューできます。ソースとトランスフォーメーションをマッピング内のプレビューポイントとして選択できます。

データ統合サービスは、プレビューポイントに基づいて、ネイティブ環境内と Spark エンジン上のどちらでジョブを実行するかを決定します。プレビューポイントは、データを表示するよう選択したマッピング内のオブジェクトです。Spark エンジンでのデータプレビューに関する詳細については、*Data Engineering Integration ユーザーガイド*を参照してください。

注: データビューアでプレビューできる最大行数は 100,000 行です。

データを **【データビューア】** ビューで表示する場合、データをファイルにエクスポートできます。ログイベントを表示するログにアクセスすることもできます。

【オブジェクトの依存関係】 ビュー

【オブジェクトの依存関係】 ビューを使用して、モデルリポジトリオブジェクトを表示、変更、または削除するときにオブジェクトの依存関係を表示します。

構成

構成とは、マッピングの実行、データのプレビュー、SQL クエリの実行、または Web サービスメッセージのプレビュー時に、Developer ツールにより適用されるひとまとまりの設定です。

構成は、デフォルトの Data Integration Service、ソースから読み取る行数、デフォルトの日付/時刻フォーマット、および最適化レベルなどの設定を制御します。作成した構成は、Developer ツールのインストールに適用されます。

以下の構成を作成できます。

- データビューアの構成。出力を **【データビューア】** ビューでプレビューする場合に Developer ツールが適用する設定を制御します。
- マッピング構成。マッピングを **【実行設定】** ダイアログボックスから、またはコマンドラインから実行する場合に Developer ツールが適用する設定を制御します。
- Web サービス設定。**【データビューア】** ビューで Web サービスの出力をプレビューする場合に、Developer ツールにより適用される設定を制御します。

構成プロパティ

Developer ツールは、出力のプレビューまたはマッピングの実行時に構成プロパティを適用します。 **【データビューア】** ビューまたはマッピングの構成プロパティは、**【実行】** ダイアログボックスで設定します。

Data Integration Service のプロパティ

Developer ツールでは、データビューア、マッピング、および Web サービス設定用の [Data Integration Service] タブが表示されます。

以下の表に、Data Integration Service に対して構成するプロパティを示します。

プロパティ	説明
デフォルトのデータ統合サービスの使用	マッピングの実行にデフォルトのデータ統合サービスを使用します。 デフォルトでは有効になっています。
データ統合サービス	デフォルトのデータ統合サービスを使用しない場合に、マッピングを実行するデータ統合サービスを指定します。
使用可能なオペレーティングシステムプロファイル	データ統合サービスでオペレーティングシステムのプロファイルを使用できる場合、マッピングを実行するためのオペレーティングシステムのプロファイルを指定します。 Developer tool でこのプロパティが表示されるのは、管理者がオペレーティングシステムのプロファイルを少なくとも 1 つユーザーに割り当てた場合のみです。データ統合サービスはユーザーに割り当てられたデフォルト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。オペレーティングシステムのプロファイルは、使用可能なオペレーティングシステムのプロファイルのリストから変更できます。

ソースのプロパティ

Developer ツールでは、データビューア、マッピング、および Web サービス設定用の [ソース] タブが表示されます。

以下の表に、ソースに設定するプロパティを示します。

プロパティ	説明
すべての行の読み取り	ソースからすべての行を読み取ります。 デフォルトでは有効になっています。
次の行数まで読み取る	すべての行を読み取らない場合にソースから読み取る最大行数を指定します。 注: カスタマイズデータオブジェクトに書き込むマッピングにこのオプションを指定すると、Data Integration Service はターゲットに書き込む前にターゲットテーブルを切り詰めません。 デフォルトは 1000 です。
すべての文字を読み取る	カラム内のすべての文字を読み取ります。 デフォルトでは無効になっています。
次の文字数まで読み取る	すべての文字を読み取らない場合に各カラムに読み取る最大文字数を指定します。SAP ソースの場合、このプロパティは無視されます。 デフォルトは 4000 です。

結果のプロパティ

Developer ツールでは、データビューアおよび Web サービス設定用の【結果】タブが表示されます。

以下の表に、【データビューア】ビューの結果に対して構成するプロパティを示します。

プロパティ	説明
すべての行を表示	【データビューア】ビューにすべての行を表示します。 デフォルトでは無効になっています。
次の行数まで表示	すべての行を表示しない場合に表示する、最大行数を指定します。 デフォルトは 1000 です。
すべての文字を表示	カラム内のすべての文字を表示します。 デフォルトでは無効になっています。
次の文字数まで表示	すべての文字を表示しない場合に各カラムで表示する、最大文字数を指定します。 デフォルトは 4000 です。

メッセージのプロパティ

Developer ツールでは、Web サービス設定用の【メッセージ】タブが表示されます。

以下の表に、メッセージに設定するプロパティを示します。

プロパティ	説明
要求メッセージで読み取られる最大文字数	入力メッセージで処理する最大文字数を指定します。
要求メッセージで表示される最大文字数	出力メッセージで表示する最大文字数を指定します。

詳細プロパティ

Developer ツールでは、データビューア、マッピング、および Web サービス設定用の【詳細】タブが表示されます。

以下の表に詳細プロパティを示します。

プロパティ	説明
デフォルトの日時形式	マッピングで文字列が日付に変換される場合に、データ統合サービスが使用する日付/時刻形式。 デフォルトは MM/DD/YYYY HH24:MI:SS です。
トレースレベルのオーバーライド	マッピングの各トランスフォーメーションのトレースレベルをオーバーライドします。トレースレベルは、データ統合サービスがマッピングログファイルに送信する情報量を決定します。 以下のいずれかのトレースレベルを選択します。 <ul style="list-style-type: none">- なし。データ統合サービスは、マッピングに設定されたトレースレベルを使用します。- Terse。データ統合サービスは、初期化情報、エラーメッセージ、および却下されたデータの通知をログに記録します。- ノーマル。データ統合サービスは、初期化情報、ステータス情報、検出されたエラー、およびトランスフォーメーション行エラーが原因でスキップされた行をログに記録します。マッピングの結果を要約しますが、個別行のレベルでは行いません。- Verbose initialization。ノーマルトレースに加え、データ統合サービスにより、追加の初期化の詳細、インデックスおよび使用されたデータファイルの名前、および詳細なトランスフォーメーション統計もログに記録されます。- 冗長データ。データ統合サービスにより、Verbose 初期化トレースに加え、マッピングに渡された各行がログに記録されます。また、カラムの精度に合わせて文字列データを切り詰めた場所と、詳細なトランスフォーメーション統計も記録します。 デフォルトは [なし] です。
ソート順	データ統合サービスがマッピングで文字データをソートする順番。 デフォルトは [バイナリ] です。

プロパティ	説明
最適化レベル	<p>データ統合サービスがマッピングに適用する最適化の方法を制御します。</p> <p>自動</p> <p>データ統合サービスは、実行モードとマッピングコンテンツに基づいて最適化を適用します。</p> <p>なし</p> <p>データ統合サービスは最適化は適用されません。</p> <p>最小</p> <p>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。</p> <p>ノーマル</p> <p>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。</p> <p>完全</p> <p>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。</p> <p>デフォルトは「自動」です。</p>
高精度	<p>マッピングを高精度で実行します。</p> <p>「高精度」を選択すると、データ値の精度が高くなります。15 桁以上の精度など、マッピングが大きな数値を生成する場合や、正確な値を必要とする場合に「高精度」を有効にします。「高精度」を有効にすると、桁数の多い数値で精度の損失を防ぐことができます。</p> <p>デフォルトでは有効になっています。</p>
クライアントにログを送信	<p>Developer ツールでログファイルを表示できるようにします。このオプションを無効にした場合、ログファイルは Administrator ツールで表示する必要があります。</p> <p>デフォルトでは有効になっています。</p>

データビューアの構成

データビューア構成は、出力を【データビューア】ビューでプレビューする場合に Developer ツールが適用する設定を制御します。

以下のオブジェクトに対する出力をプレビューする場合に、データビューア構成を選択できます。

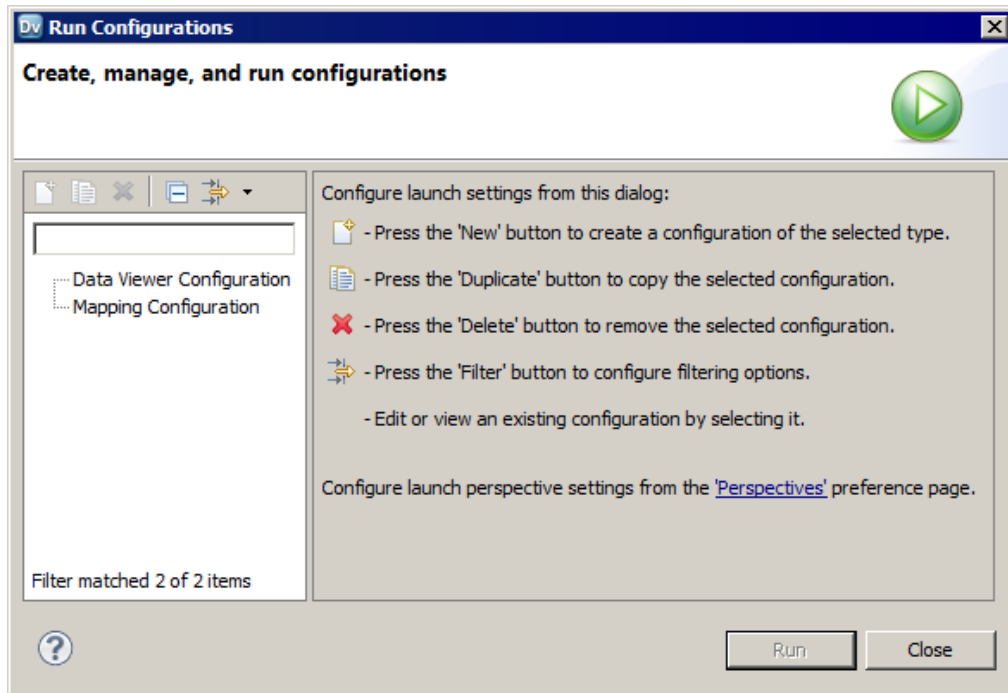
- カスタムデータオブジェクト
- 論理データオブジェクト
- 論理データオブジェクト読み取りマッピング
- 物理データオブジェクト
- マッピング内のソースとトランスフォーメーション
- 仮想ストアドプロシージャ
- 仮想テーブル
- 仮想テーブルマッピング


データビューア構成の作成

データビューア構成を作成し、出力を【データビューア】ビューでプレビューする場合に Developer ツールが適用する設定を制御します。

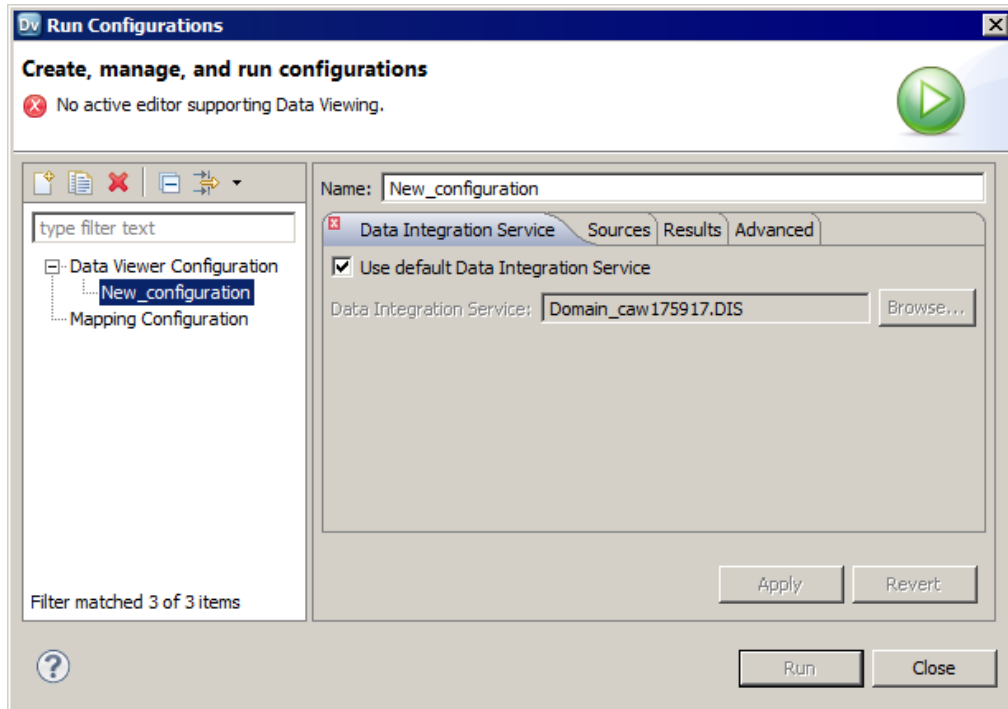
1. 【実行】 > 【実行ダイアログを開く】 をクリックします。

【実行設定】 ダイアログボックスが表示されます。



2. 【実行】 をクリックします。
3. 【新規】 ボタンをクリックします ()。

【実行設定】ダイアログボックスの右側のパネルには、データビューアの設定プロパティが表示されます。



4. データビューアの設定の名前を入力します。
5. データビューアの設定プロパティを設定します。
6. 【適用】をクリックします。
7. 【閉じる】をクリックします。

データビューア構成が作成されます。

マッピング構成

マッピング設定は、マッピングを【実行設定】ダイアログボックスまたはコマンドラインから実行する場合に Developer ツールが使用するマッピングのデプロイメントのプロパティを制御します。

Developer tool から実行しているマッピングにマッピング設定を適用するには、マッピングを【実行設定】ダイアログボックスから実行するか、詳細オプションを使用して実行します。【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、Developer tool は、デフォルトのマッピングのデプロイメントのプロパティを使用してマッピングを実行します。

コマンドラインから実行するマッピングにマッピングのデプロイメントのプロパティを適用するには、マッピングをアプリケーションに追加するときにマッピング構成を選択します。選択したマッピング構成は、アプリケーション内のすべてのマッピングに適用されます。

マッピングのデプロイメントのプロパティは、アプリケーションを編集するときに変更できます。管理者は Administrator ツールを使用してマッピングのデプロイメントのプロパティを変更することもできます。変更内容を有効にするには、アプリケーションを再デプロイする必要があります。

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、『Developer マッピングガイド』を参照してください。

マッピング構成の作成

マッピング構成を作成し、マッピングを実行するときに Developer tool が使用するマッピングのデプロイメントプロパティを制御します。

1. **【実行】** > **【実行ダイアログを開く】** をクリックします。
【実行設定】 ダイアログボックスが表示されます。
2. **【マッピング設定】** をクリックします。
3. **【新規】** ボタンをクリックします。
【実行設定】 ダイアログボックスの右側のパネルには、マッピング設定のプロパティが表示されます。
4. マッピング設定の名前を入力します。
5. マッピング設定のプロパティを構成します。
6. **【適用】** をクリックします。
7. **【閉じる】** をクリックします。
マッピング構成が作成されます。

Web サービスの設定

Web サービス設定では、**【データビューア】** ビューで Web サービスの出力をプレビューする場合に Developer ツールによって適用される設定を制御します。

Web サービス設定を作成して、特定の Web サービスに使用する設定を制御します。操作マッピングまたは操作マッピングのトランスフォーメーションの出力をプレビューする際に、Web サービス設定を選択できます。

注: プレビューするすべての Web サービスに適用する Web サービス設定を作成するには、**【設定】** ダイアログボックスを使用してデフォルトの Web サービス設定を更新します。

Web サービス設定の作成

Web サービス設定を作成して、**【データビューア】** ビューで Web サービスの出力をプレビューする場合に Developer ツールで適用される設定を制御します。

1. **【実行】** > **【実行ダイアログを開く】** をクリックします。
【実行】 ダイアログボックスが表示されます。
2. **【Web サービスの設定】** をクリックします。
3. **【新規】** をクリックします。
4. Web サービス設定の名前を入力します。
5. Web サービス設定のプロパティを設定します。
6. **【適用】** をクリックします。
7. **【閉じる】** をクリックします。

デフォルトの構成プロパティの更新

データビューア、マッピング、および Web サービスのデフォルトの設定プロパティを更新できます。

1. **【ウィンドウ】** > **【設定】** をクリックします。
【設定】 ダイアログボックスが表示されます。
2. **【Informatica】** > **【実行設定】** をクリックします。

3. **【データビューア】**、**【マッピング】**、または **【Web サービス】** 設定を選択します。
4. データビューア、マッピング、または Web サービスのデフォルトの設定プロパティを設定します。
5. **【OK】** をクリックします。

デフォルトの構成プロパティが更新されます。

構成のトラブルシューティング

同じ名前で大文字と小文字が異なる 2 つの設定を作成したが、Developer ツールを閉じて再度開くと、1 つの設定が見つからない。

データビューア構成名とマッピング構成名では大文字と小文字が区別されません。同じ名前で大文字と小文字が異なる複数の構成を作成すると、Developer ツールはユーザーがセッションを終了した時点で構成の 1 つを削除します。Developer ツールはこの 2 つの構成名を一意とは見なしません。

長い名前の構成を作成すると、ファイルを書き込めないというエラーメッセージが表示される。

Developer ツールは、ビューア構成とマッピング構成を、Developer ツールを実行するマシン上のファイルに格納します。例えば 100 文字以上の長い名前の構成を作成すると、このファイルはハードウェアドライブに保存されません。

この問題を回避するには、構成名を短くします。

カスタマイズされたデータオブジェクトのデータのマイプレビューが失敗しました。

カスタマイズされたデータオブジェクトがデータ型変換を実行し、プッシュダウン最適化がソースに設定されていない場合、Developer tool はデータビューアービューにデータを表示できません。

この問題を回避するには、プッシュダウン最適化を設定して、処理をソースにプッシュします。

データのエクスポート

【データビューア】 ビューに表示するデータを、TXT ファイルや CSV ファイルなどのタブ区切りのフラットファイルにエクスポートできます。階層的データ型のデータを JSON ファイルにエクスポートできます。データのローカルコピーを作成するには、データをエクスポートします。

1. **【データビューア】** ビューで、結果を右クリックして **【データのエクスポート】** 選択します。
2. ファイル名と拡張子を入力します。
3. ファイルの保存先を選択します。
4. **【OK】** をクリックします。

オブジェクトの依存関係

オブジェクトを変更または削除する前に、モデルリポジトリ内でそのオブジェクトの影響を受けるすべてのオブジェクトに関して、オブジェクトの依存関係を確認することができます。選択したオブジェクトに依存して

いるすべてのオブジェクトと、そのオブジェクトが依存しているオブジェクトに関して、オブジェクトの依存関係を確認することができます。

例えば、複数のアプリケーションでデプロイされたデータオブジェクトを削除する必要があります。しかし、このデータオブジェクトを削除することでデプロイ済みのアプリケーションに影響が及ぶかが分かっていません。このような場合に、オブジェクトの依存関係を表示することで、このデータオブジェクトがデプロイ済みのアプリケーションに影響を及ぼすかどうかを判定することができます。データオブジェクトを削除したら、アプリケーションをデプロイしなおすことができます。

【オブジェクトの依存関係】 ビューで、オブジェクトの依存関係を表示することができます。読み取り権限のないプロジェクト内に依存オブジェクトがある場合、Developer ツールでは **【オブジェクトの依存関係】** ビューにそのオブジェクトが表示されません。Developer ツールに、Developer ツールがオブジェクトにアクセスできないことを伝えるメッセージが表示されます。

オブジェクトの依存関係の表示

【オブジェクトの依存関係】 ビューの中のオブジェクトに関して、オブジェクトの依存関係を表示することができます。ユーザーは、**【Object Explorer】** ビューから選択するオブジェクトに関して、または **【接続エクスプローラ】** ビューから選択した接続に関して、そのオブジェクトの依存性を表示できます。

次のタスクを実行すると、オブジェクトの依存性を表示することができます。

循環依存の関係の表示。

2 つのオブジェクトが相互依存している場合、循環依存性が生じます。例えば、オブジェクト A はオブジェクト B に依存し、オブジェクト B はオブジェクト A に依存するとします。オブジェクトの依存関係ツリーで同じオブジェクトが 2 回出てくることを検知した場合、Developer ツールはオブジェクトの 2 回目以降のインスタンスを表示しません。その代わりに、Developer ツールは循環依存が発生していることを示す丸のアイコンをオブジェクトに追加します。

ダウンストリームまたはアップストリームの依存関係の表示。

選択されたオブジェクトに依存するオブジェクトを確認するには、ダウンストリームの依存関係を表示します。デフォルトでは Developer ツールにダウンストリームの依存関係が表示されます。

選択されたオブジェクトが依存するオブジェクトを確認するには、アップストリームの依存関係を表示します。

オブジェクトの依存関係のフィルタリング。

オブジェクトの依存関係にフィルタリングを実行して依存するオブジェクトのリストを絞り込むことができます。オブジェクトタイプでのフィルタリングと、プロジェクトでのフィルタリングを選ぶことができます。例えば、特定のオブジェクトがワークフローに影響を及ぼすかを知る必要があるとします。この場合は、オブジェクトタイプでフィルタリングしてワークフローを選択します。

オブジェクトの依存関係の履歴の表示。

オブジェクトの依存関係のうち直近の 10 件に関する履歴を表示します。

オブジェクトの依存関係の表示

依存関係を表示する際、フィルタを適用したり、履歴をクリアしたり、特定の依存性に関する詳細を選択したりすることができます。

1. **【Object Explorer】** ビューまたは **【接続エクスプローラ】** ビューで、オブジェクトまたは接続を右クリックしてから **【依存関係の検索】** をクリックします。

【オブジェクトの依存関係】 ビューには、オブジェクトの依存関係ツリーの中で選択されたオブジェクトまたは接続に関するオブジェクトの依存関係がリスト表示されます。

2. オブジェクトの依存関係ツリー内のオブジェクトまたは接続を選択すると、オブジェクトまたは接続のプロパティを表示できます。
3. 必要に応じて依存性をフィルタリングする場合は、**【フィルタ】** アイコンをクリックして、オブジェクトまたはプロジェクトのタイプでフィルタリングするように選択します。
4. 次のタイプの依存関係を表示できます。
 - 循環。Developer ツールは循環依存が発生していることを示す丸のアイコンをオブジェクトに追加します。
 - アップストリーム。アップストリームの依存関係を表示する場合は、**【アップストリームの依存関係】** アイコンをクリックします。
 - ダウンストリーム。ダウンストリームの依存関係を表示する場合は、**【ダウンストリームの依存関係】** アイコンをクリックします。
5. 必要に応じて依存関係の履歴を表示する場合は、**【履歴】** アイコンをクリックして、オブジェクトの依存関係の履歴を表示します。必要に応じて、**【履歴のクリア】** をクリックして履歴をクリアします。

オブジェクトの依存関係のフィルタリング

オブジェクトに依存関係ツリーを **【タイプ】** および **【プロジェクト】** でフィルタリングできます。フィルタされた結果で、オブジェクトの一部が親オブジェクトに直接的に依存しないものであることがあります。これらのオブジェクトにはプラス (+) アイコンが表示され、オブジェクトの依存関係ツリー内の親オブジェクトへの絶対パスが非表示になります。このパスを拡大して親オブジェクトまでのすべてのオブジェクトレベルを挿入することもできます。

1. **【オブジェクトの依存関係】** ビューで、**【フィルタ】** アイコンをクリックします。
【フィルタ】 ダイアログボックスが開きます。
2. **【タイプ】** タブでのデフォルトの選択を受け入れて、オブジェクトタイプでフィルタ表示します。
3. 必要があれば、**【プロジェクト】** タブをクリックして、モデルリポジトリ内のすべてのプロジェクトでフィルタリングするか、または各プロジェクトでフィルタリングするかを選択します。
 - モデルリポジトリ内のすべてのプロジェクトでフィルタリングする場合は、すべてのプロジェクトを含めるオプションを選択します。
 - 各プロジェクトでフィルタリングする場合は、開いたプロジェクトを含めるオプションを選択します。
4. オブジェクトタイプまたはプロジェクトを個別に選択するのか、すべてのオブジェクトタイプまたはプロジェクトを選択するのか、あるいは選択してあったオブジェクトタイプとプロジェクトをクリアするのかを指定します。
 - オブジェクトタイプまたはプロジェクトを個別に選択する場合は、オブジェクトタイプまたはプロジェクトをそれぞれ選択します。
 - すべてのオブジェクトタイプまたはプロジェクトを選択する場合は、**【すべて選択】** をクリックします。
 - 選択してあったオブジェクトタイプとプロジェクトをクリアする場合は、**【すべて選択解除】** をクリックします。
5. **【OK】** をクリックします。
フィルタリングされたオブジェクトの依存関係ツリーに **【オブジェクトの依存関係】** ビューが表示されます。
6. 必要に応じて、オブジェクトの名前に対してプラス (+) のアイコンが表示された場合に、そのオブジェクトを右クリックし、**【パスを拡大】** をクリックして、親のオブジェクトまでのすべてのオブジェクトレベルを挿入します。

ログ

Data Integration Service は、マッピングやプロファイルの実行、データのプレビュー、または SQL クエリの実行時に、ログイベントを生成します。ログイベントには Data Integration Service によって実行されるタスク、エラー、ロード要約についての情報とトランスフォーメーション統計が含まれます。

Developer ツールで生成されたログを表示して、ローカルディレクトリに保存できます。

【データビューア】ビューの【ログの表示】ボタンからログイベントを表示できます。

【実行】 > 【マッピングの実行】からマッピングを実行する場合、【進捗状況】ビューでログイベントを表示できます。Developer ツールでログイベントを開くには、マッピング実行のリンクをクリックして、【ログに移動】を選択します。

プロファイルを実行する場合、Monitoring ツールからログイベントを表示できます。

ログをファイルに保存するには、【ファイル】 > 【名前を付けてコピーを保存】をクリックして、ディレクトリを選択します。デフォルトでは、ログファイルは次のディレクトリに保存されます。c:\[TEMP]\AppData\Local\Temp

ログファイル形式

ログファイルの情報は、実行時のイベントのシーケンスによって異なります。ログに送信される情報量は、トレースレベルによって決まります。

Data Integration Service は、マッピングやプロファイルの実行、データのプレビュー、または SQL クエリの実行時に、以下の情報でログファイルを更新します。

論理 DTM メッセージ

マッピングのコンパイル、最適化、変換の準備に関する情報が含まれます。ログイベントおよび情報量は、設定されている構成のプロパティによって異なります。

データ変換マネージャ (DTM) メッセージ

ソースへの接続の確立、データの読み取り、データのトランスフォーム、ターゲットへのデータのロードに関する情報が含まれます。

ロード要約およびトランスフォーメーション統計メッセージ

ソースから読み取られた行数、ターゲットに出力された行数、拒否された行数、および実行時間に関する情報が含まれます。

検証設定

【検証ログ】ビューに表示されるエラーメッセージの数を制限できます。【検証ログ】ビューでオブジェクトまたはオブジェクトタイプ別にエラーメッセージをグループ化することもできます。

エラーメッセージのグループ化

【検証ログ】ビューのエラーメッセージをグループ化して、メッセージをオブジェクトまたはオブジェクトタイプ別に分類します。これを行わなかった場合、メッセージはアルファベット順に表示されます。

【検証ログ】ビューのエラーメッセージをグループ化するには、【メニュー】 > 【グループ別】を選択し、【オブジェクト】または【オブジェクトタイプ】を選択します。

エラーメッセージのグループを削除するには、**【メニュー】** > **【グループ別】** > **【なし】** を選択します。エラーメッセージはグループ化が解除され、**【検証ログ】** ビューでアルファベット順に一覧表示されます。

エラーメッセージの制限

【検証ログ】 ビューに表示されるエラーメッセージの数を制限できます。この制限により、グループに表示されるメッセージ数や **【検証ログ】** ビューに表示されるメッセージの合計数が決まります。エラーメッセージはアルファベット順に一覧表示され、制限を適用すると、下から上に向かって削除されます。

1. **【ウィンドウ】** > **【設定】** をクリックします。
【設定】 ダイアログボックスが表示されます。
2. **【Informatica】** > **【検証】** を選択します。
3. 必要に応じて、エラーの上限を設定し、表示される項目数を設定します。
デフォルトは 100 です。
4. デフォルト値に戻すには、**【デフォルトの復元】** をクリックします。
5. **【適用】** をクリックします。
6. **【OK】** をクリックします。

Developer tool からのジョブの監視

Developer tool から Monitoring ツールにアクセスし、アプリケーションとジョブのステータスを監視できます。管理者は、Administrator ツールでアプリケーションとジョブを監視することもできます。

アプリケーションとジョブを監視して、オブジェクトのプロパティ、ランタイム統計、およびランタイムレポートを表示します。

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】** ビューにドメイン内のジョブのステータス概要が図で表示されます。

【実行統計】 ビューにジョブの全般のプロパティおよびステータス情報が表示されます。例えば、ジョブの開始者やジョブの完了に要した時間を表示できます。マッピングジョブを監視する場合、ジョブ実行のスループットおよびリソースの使用状況統計も表示できます。

Developer tool からアプリケーションとジョブを監視するには、**【進行状況】** ビューを開き、**【表示】** メニュー > **【ジョブの監視】** をクリックします。アプリケーション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。

第 9 章

アプリケーションデプロイメント

この章では、以下の項目について説明します。

- [アプリケーションデプロイメントの概要, 148 ページ](#)
- [n アプリケーションの作成, 149 ページ](#)
- [アプリケーションのプロパティ, 150 ページ](#)
- [アプリケーションへのデプロイメント, 152 ページ](#)
- [アーカイブファイルへのデプロイメント, 154 ページ](#)
- [リソースパラメータを使用したデプロイメント, 154 ページ](#)
- [ランタイムオブジェクトの表示, 156 ページ](#)
- [アプリケーションの再デプロイメント, 156 ページ](#)
- [アプリケーションの状態情報, 157 ページ](#)
- [アプリケーションの作成、デプロイ、更新の方法, 158 ページ](#)

アプリケーションデプロイメントの概要

アプリケーションは、物理データオブジェクト、論理データオブジェクト、データサービス、マッピング、マップレット、トランスフォーメーション、Web サービス、ワークフローを含むことができるデプロイ可能なオブジェクトです。

Developer tool の外部からアプリケーションオブジェクトにアクセスできるようにするには、そのオブジェクトを含むアプリケーションをデプロイします。新規アプリケーションを作成するか、オブジェクトをデプロイするときにアプリケーションを作成できます。アプリケーションをデプロイすると、そのアプリケーションオブジェクトにはデータ構造の変更は反映されなくなります。

アプリケーションオブジェクトは次の方法でデプロイできます。

データ統合サービスにデプロイする。

アプリケーションをデータ統合サービスにデプロイすると、アプリケーションが実行され、エンドユーザーがアプリケーションに接続できるようになります。

アプリケーション内のオブジェクトのタイプに応じて、適切な権限のあるエンドユーザーはオブジェクトに対してクエリを実行したり、Web サービスにアクセスしたり、コマンドラインからマッピングやワークフローを実行したりすることができます。また、ユーザーがサードパーティのクライアントツールでオブジェクトをクエリできるようにオブジェクトをデプロイすることもできます。

ネットワークファイルシステムにデプロイする。

オブジェクトをネットワークファイルシステムにデプロイすると、Developer tool によってアプリケーションアーカイブファイルが作成されます。バージョン管理システムにオブジェクトをアーカイブする場合は、オブジェクトをアーカイブファイルにデプロイします。

管理者がオブジェクトをデータ統合サービスにデプロイすることが組織で必要とされている場合、管理者は Administrator ツールを使用してアプリケーションアーカイブファイルをデータ統合サービスにデプロイできます。

また、アプリケーションアーカイブファイルからモデルリポジトリ内のプロジェクトまたはフォルダへオブジェクトをインポートすることもできます。

アプリケーションオブジェクトまたはアプリケーションアーカイブファイルをデータ統合サービスにデプロイ後、Developer tool を使用してランタイムアプリケーションに接続し、ランタイムオブジェクトの読み取り専用コピーを表示できます。

n アプリケーションの作成

アプリケーションを作成してデプロイし、アプリケーションに含まれるマッピングおよびその他のオブジェクトを実行します。

オブジェクトをデータ統合サービスにデプロイするときにアプリケーションを作成するか、新規アプリケーションを作成できます。新規アプリケーションの作成時に、アプリケーションに含めるオブジェクトを選択します。

マッピング、ワークフロー、データオブジェクト、パラメータセットなどのオブジェクトを追加できます。論理データオブジェクトをアプリケーションに直接追加することはできません。しかし、論理データオブジェクトを含む Web サービスおよび SQL データサービス用のアプリケーションは作成できます。

アプリケーションの作成時に、完全なアプリケーションか差分アプリケーションを作成できます。完全なアプリケーションとは、アプリケーションを更新するたびに再デプロイする必要があるアプリケーションです。差分アプリケーションとは、アプリケーションパッチをデプロイすることによって更新できるアプリケーションです。パッチでは、他のアプリケーションオブジェクトが実行されている状態でアプリケーションオブジェクトを更新できますが、完全なアプリケーションを再デプロイする場合と比べて実行に時間がかかることがあります。

アプリケーションを作成すると、Developer tool がアプリケーションを検証します。アプリケーションの有効性は、アプリケーションに含まれるオブジェクトの有効性によって左右されます。例えば、マッピングを含むアプリケーションが有効であるためには、マッピング内のオブジェクト間のリンク設定が有効である必要があります。アプリケーションが有効でない場合は、[検証ログ] ビューまたはエラーダイアログボックスにエラーが表示されます。

完全なアプリケーション

完全なアプリケーションとは、データ統合サービスにデプロイできるアプリケーションです。アプリケーションまたはアプリケーションオブジェクトを編集するたびに、アプリケーションを再デプロイして変更を適用する必要があります。

注: アプリケーションを再デプロイしている間は、アプリケーションに含まれるすべてのオブジェクトにアクセスできません。

例

検索機能を実行するマッピングと、ショッピングカートに選択した検索結果を入れる別のマッピングを作成します。この2つのマッピングを含むアプリケーションを作成し、データ統合サービスにアプリケーションをデ

プロイします。アプリケーションオブジェクトの出力をテストしたら、オブジェクトに変更を加えます。アプリケーションを更新するには、アプリケーションを再デプロイします。アプリケーションはアプリケーションアーカイブファイルにもデプロイします。管理者はアーカイブファイルをバージョン管理システムにチェックインします。

差分アプリケーション

差分アプリケーションとは、データ統合サービスにデプロイでき、アプリケーションパッチをデプロイすることによって更新できるアプリケーションです。

パッチには、更新するオブジェクトやデプロイされた差分アプリケーションに追加するオブジェクトが含まれます。パッチをデプロイすると、データ統合サービスがパッチのコンテンツに基づいて、デプロイされたアプリケーションを更新します。

パッチを使用することで、アプリケーションのアップタイムが向上します。例えば、ストリーミングマッピングを含むアプリケーションがあるとします。このストリーミングマッピングがデータを処理し続けている状態で、アプリケーション内の他のオブジェクトを更新できます。

アプリケーションパッチの詳細については、[第 10 章, 「アプリケーションパッチのデプロイメント」 \(ページ 167\)](#)を参照してください。

注: 差分アプリケーションには、Web サービス、SQL データサービス、REST Web サービスを含めることはできません。

例

100 個のマッピングと、それらのマッピングを含む差分アプリケーションを作成します。次に、そのアプリケーションをデータ統合サービスにデプロイします。アプリケーションオブジェクトの出力をテストした後、いくつかのオブジェクトを変更します。

ランタイムアプリケーション内のオブジェクトを更新するために、変更したオブジェクトを含むアプリケーションパッチを作成します。そのアプリケーションパッチをデータ統合サービスにデプロイすると、サービスがパッチを差分アプリケーションに適用してオブジェクトを更新します。この間、他のユーザーが実行している可能性があるその他のアプリケーションオブジェクトを停止する必要はありません。アプリケーションとアプリケーションパッチはアプリケーションアーカイブファイルにもデプロイします。管理者はアーカイブファイルをバージョン管理システムにチェックインします。

アプリケーションのプロパティ

アプリケーションを作成すると、**アプリケーションエディタ**でプロパティを編集できます。

アプリケーション全般のプロパティ

以下の表に、**【概要】** タブで設定できるアプリケーション全般のプロパティを示します。

プロパティ	説明
名前	アプリケーションの名前。
説明	オプション。アプリケーションの説明。
アプリケーションタイプ	アプリケーションのタイプ。完全または差分。

マッピングのデプロイメントのプロパティ

以下の表では、アプリケーションがマッピングを含む場合に【詳細】タブで設定できるマッピングのデプロイメントのプロパティを示します。

プロパティ	説明
デフォルトの日時形式	マッピングで文字列が日付に変換される場合に、データ統合サービスが使用する日付/時刻形式。 デフォルトは MM/DD/YYYY HH24:MI:SS です。
トレースレベルのオーバーライド	マッピングの各トランスフォーメーションのトレースレベルをオーバーライドします。トレースレベルは、データ統合サービスがマッピングログファイルに送信する情報量を決定します。 以下のいずれかのトレースレベルを選択します。 <ul style="list-style-type: none">- なし。トランスフォーメーションごとに設定したトレースレベルはオーバーライドされません。- 簡易。データ統合サービスは、初期化、エラーメッセージ、および拒否されたデータの通知に関する情報にログに記録します。- ノーマル。データ統合サービスは、初期化、ステータス、検出されたエラー、およびトランスフォーメーション行エラーが原因でスキップされた行に関する情報をログに記録します。データ統合サービスは、個別の行レベルではなくマッピング結果を要約します。- 詳細 - 初期化。ノーマルトレースに加え、データ統合サービスにより、追加の初期化の詳細、インデックスおよび使用されたデータファイルの名前、および詳細なトランスフォーメーション統計もログに記録されます。- 詳細 - データ。データ統合サービスにより、詳細 - 初期化トレースに加え、マッピングに渡された各行がログに記録されます。ログには、データ統合サービスがカラムの精度に合わせて文字列データを切り詰めた場所を記録します。このログには、詳細なトランスフォーメーション統計が含まれます。データ統合サービスではトランスフォーメーションを処理する際にブロック中のすべての行の行データが書き込まれます。 <p>注: ワークフローログ内の Spark エンジンの情報を表示するには、Developer tool で【詳細 - 初期化】または【詳細 - データ】を選択します。</p> <p>デフォルトは【なし】です。</p>
ソート順	データ統合サービスがマッピングで文字データをソートする順番。 デフォルトは【バイナリ】です。

プロパティ	説明
最適化レベル	<p>データ統合サービスがマッピングに適用する最適化の方法を制御します。</p> <p>自動</p> <p>データ統合サービスは、実行モードとマッピングコンテンツに基づいて最適化を適用します。</p> <p>なし</p> <p>データ統合サービスは最適化は適用されません。</p> <p>最小</p> <p>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。</p> <p>ノーマル</p> <p>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。</p> <p>完全</p> <p>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。</p> <p>デフォルトは「自動」です。</p>
高精度	<p>マッピングを高精度で実行します。</p> <p>「高精度」を選択すると、データ値の精度が高くなります。15桁以上の精度など、マッピングが大きな数値を生成する場合や、正確な値を必要とする場合に「高精度」を有効にします。「高精度」を有効にすると、桁数の多い数値で精度の損失を防ぐことができます。</p> <p>デフォルトでは有効になっています。</p>

アプリケーションへのデプロイメント

アプリケーションとして、またはアプリケーションの一部であるデータサービスとしてオブジェクトをデプロイすることができます。

Web サービスまたは SQL データサービスとしてデプロイできるオブジェクトもあります。最初に、アプリケーションを作成してオブジェクトを追加します。次に、アプリケーションをデプロイすると、Developer tool によってオブジェクトに基づいてサービスを作成するように求められます。サービスがアプリケーションに追加されます。

また、ユーザーがサードパーティのクライアントツールでオブジェクトをクエリできるようにオブジェクトをデプロイすることもできます。

データ統合サービスにアプリケーションをデプロイすると、ユーザーは、マッピング、ワークフロー、およびアプリケーションに含まれる他のオブジェクトにアクセスできるようになります。

アプリケーションをデプロイすると、データ統合サービスによってアプリケーションオブジェクトが実行されます。

マッピングまたはワークフローをデプロイすると、Developer tool によってアプリケーションを作成するように求められます。

デプロイするアプリケーションにパラメータセットを追加することができます。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で使用するパラメータとパラメータ値を含みます。データ統合サービスでワークフローまたはマッピングを実行するときにパラメータ値を使用するには、ワークフローまたはマッピングと一緒にパラメータセットをデプロイします。アプリケーションに複数のパラメータセットを追加すると、マッピングまたはワークフローの実行ごとに異なるパラメータセットを使用できます。

同じ名前の完全なアプリケーションを同じデータ統合サービスにデプロイすると、データ統合サービスは、デプロイされたアプリケーションとアプリケーション内のすべてのオブジェクトを上書きします。

Web サービスとしてのオブジェクトのデプロイ

以下のオブジェクトを Web サービスとしてデプロイできます。

- マップレット
- Web サービスコンシューマトランスフォーメーションを除くトランスフォーメーション
- フラットファイル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
- 論理データオブジェクト

Web サービスとしてオブジェクトをデプロイすると、Developer tool によって、オブジェクトに基づいて Web サービスを作成し、この Web サービスを含むアプリケーションを作成するように求められます。

Web サービスとしてオブジェクトをデプロイする場合は、以下の情報を入力します。

プロパティ	説明
名前	Web サービスの名前。
場所	アプリケーションを格納するモデルリポジトリプロジェクトフォルダ。
名前空間	ユーザーがアクセスできる Web サービスの URL。データ統合サービスは、WSDL ファイルで定義されている URL をオーバーライドします。
プレフィックス	ターゲット名前空間のプレフィックス。

SQL データサービスとしてのオブジェクトのデプロイ

以下のデータオブジェクトを SQL データサービスとしてデプロイできます。

- 物理データオブジェクト
- 論理データオブジェクト

SQL データサービスとしてデータオブジェクトをデプロイすると、Developer tool によって、オブジェクトに基づいて SQL データサービスを作成し、このサービスを含むアプリケーションを作成するように求められます。

SQL データサービスとしてオブジェクトをデプロイする場合は、以下の情報を入力します。

プロパティ	説明
名前	Web サービスの名前。
場所	アプリケーションを格納するモデルリポジトリプロジェクトフォルダ。

アーカイブファイルへのデプロイメント

アーカイブファイルには、アプリケーションオブジェクトとメタデータが XML 形式で保存されます。

次のタイプのアーカイブファイルを作成できます。

アプリケーションアーカイブファイル

アプリケーションアーカイブファイルは、アプリケーションのアプリケーションオブジェクトとメタデータについてのすべての情報を保存する XML ファイルです。ファイルの拡張子は.iar です。

アプリケーションパッチアーカイブファイル

アプリケーションパッチアーカイブファイルは、アプリケーションパッチに関する情報を保存する XML ファイルです。このファイルには、パッチに含まれるアプリケーションオブジェクト、パッチの説明、対応する差分アプリケーションの名前、パッチが状態情報を保持するか破棄するかに関する情報が保存されます。ファイルの拡張子は.piar です。

以下の理由により、アーカイブファイルを作成する場合があります。

- アプリケーションまたはアプリケーションパッチをデプロイする。組織で、アプリケーションまたはアプリケーションパッチをデータ統合サービスにデプロイする機能が管理者に限定されている場合、管理者はアーカイブファイルをデータ統合サービスにデプロイできます。

アプリケーションをデプロイする場合は、管理者は Administrator ツールまたは infacmd deployApplication を使用できます。アプリケーションパッチをデプロイする場合は、管理者が infacmd tools patchApplication を使用できます。

- アプリケーションまたはアプリケーションパッチをモデルリポジトリにインポートする。管理者は infacmd tools importObjects を使用してアーカイブファイルをモデルリポジトリにインポートできます。モデルリポジトリ内にすでにあるオブジェクトを含むアーカイブファイルをインポートする場合、Developer tool によって新しいプロジェクトフォルダが作成されます。

注: アプリケーションをインポートするモデルリポジトリのリリースバージョンには、そのアプリケーションをアーカイブファイルにエクスポートした元のバージョンとの互換性が必要です。

- 別のシステムにファイルをアーカイブします。例えば、モデルリポジトリがバージョン管理システムに統合されていない場合、管理者はプロジェクト監査やオブジェクトのリカバリを行うためにアーカイブファイルをバージョン管理システムにチェックインできます。

リソースパラメータを使用したデプロイメント

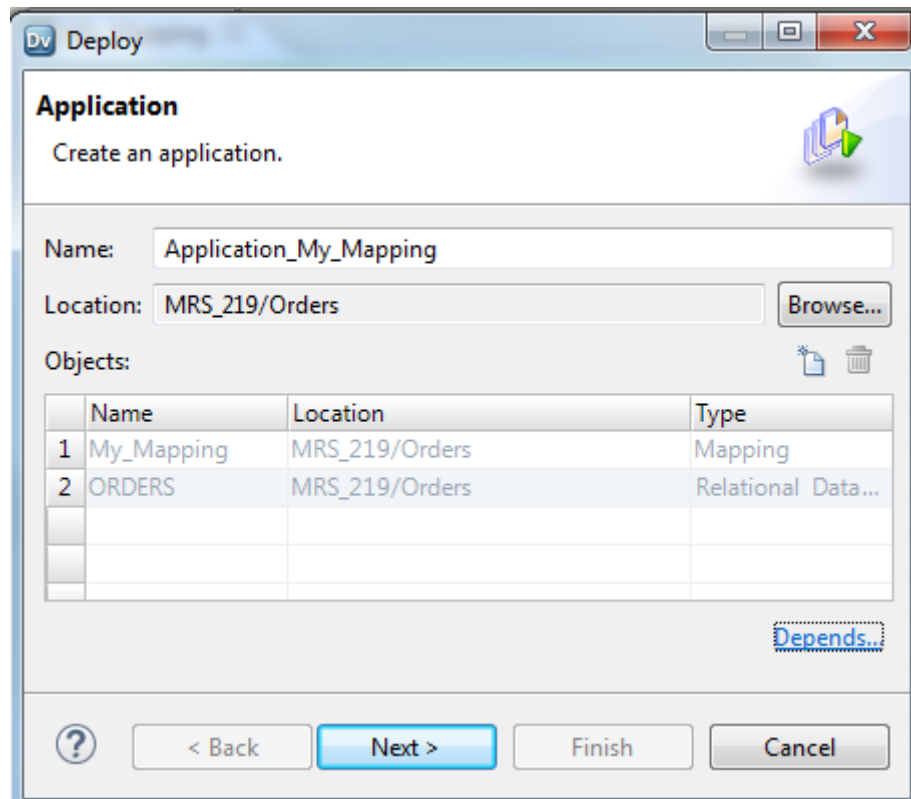
ソース、ルックアップ、またはターゲットトランスフォーメーションにリソースパラメータを持つマッピングをデプロイする場合、アプリケーションがマッピングを実行するために必要なデータオブジェクトのリストを表示できます。

リソースパラメータ値のデータオブジェクトは、アプリケーションを作成したり、マッピングをデプロイしたりするときに確認できます。データオブジェクトを表示するには、**【アプリケーションの作成】** ダイアログボックスまたは **【デプロイメント】** ダイアログボックスの **【依存】** リンクをクリックします。データオブジェクトを表示するときに、アプリケーションからデータオブジェクトを除外することを選択できます。ランタイムに、リソースパラメータの値を別のデータオブジェクトに変更します。

アプリケーションからデータオブジェクトを除外するには、**【含む/除外する】** チェックボックスをクリアします。

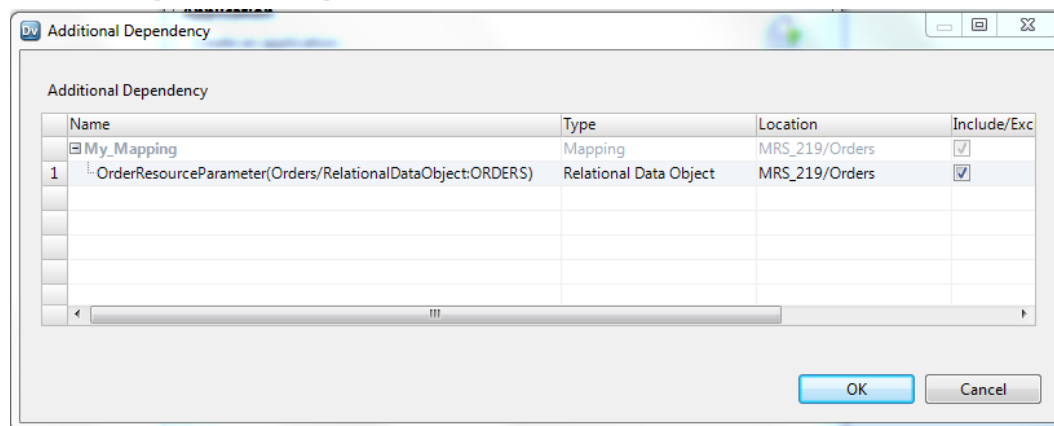
マッピングには複数のマップレットを含めることができ、マップレットには他のマップレットを含めることができます。別のパラメータがパラメータ階層内でリソースパラメータ値をオーバーライドする場合は、依存関係リストにリソースパラメータ値は表示されません。

以下の図は、[デプロイメント] ダイアログボックスの [依存] リンクを示しています。



[依存] リンクをクリックすると、[追加の依存関係] ダイアログボックスが表示されます。Developer tool には、アプリケーションがマッピングを実行するために必要なデータオブジェクトのリストが表示されます。

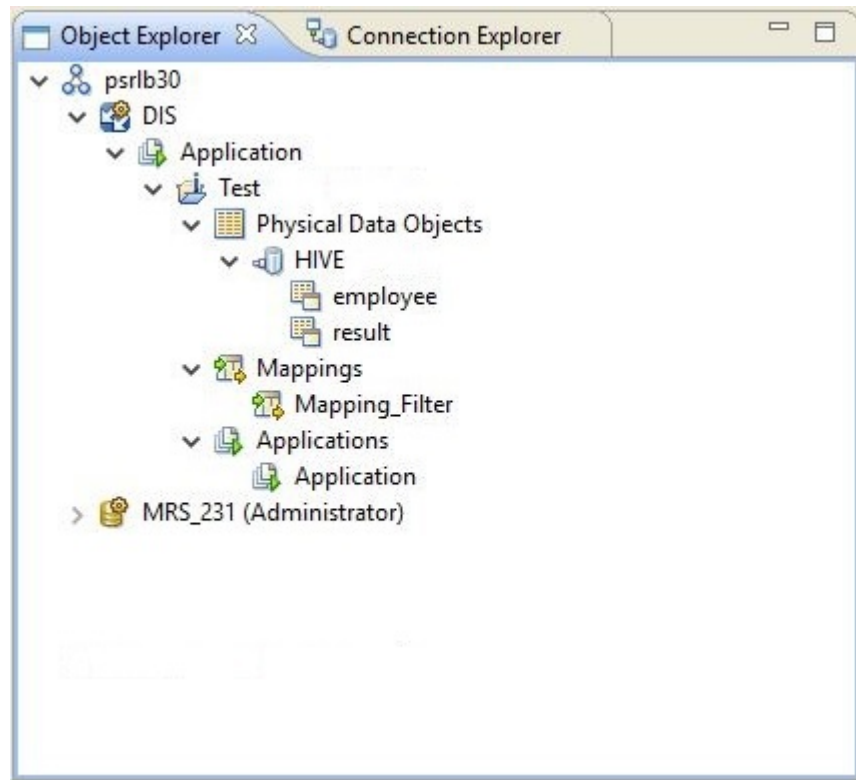
以下の図は、[追加の依存関係] ダイアログボックスを示しています。



ランタイムオブジェクトの表示

ランタイムアプリケーションに接続して、デプロイ済みアプリケーション内のランタイムオブジェクトの読み取り専用コピーを表示します。

次の図は、[Object Explorer] ビューのランタイムアプリケーションを示しています。



ランタイムオブジェクトを【エディタ】で開くと、アプリケーションまたはアプリケーションパッチが正常にデプロイされたことを確認できます。アプリケーションの再デプロイまたはアプリケーションパッチのデプロイ後に、ランタイムアプリケーションを更新して、更新されたオブジェクトを表示できます。

テストの結果または新しいリリースの要件に基づいて、対応する設計時オブジェクトの編集中にランタイムオブジェクトを参照することもできます。ランタイムオブジェクトは対応する設計時オブジェクトのデータ構造内の変更から分離されているため、ランタイムオブジェクトを変更せずに設計時オブジェクトの編集を続けることができます。

アプリケーションの再デプロイメント

アプリケーションのコンテンツを変更した後で、変更を適用するアプリケーションを再デプロイできます。

以下の理由により、アプリケーションを再デプロイする場合があります。

- 完全なアプリケーション内のアプリケーションオブジェクトを編集、追加、または削除し、それらの変更をデプロイされた完全なアプリケーションに適用したい。
- 差分アプリケーション内のアプリケーションオブジェクトを削除し、それらの変更をデプロイされた差分アプリケーションに適用したい。

- 差分アプリケーションのアプリケーションパッチに多数のオブジェクトが含まれており、ユーザーがパッチをデプロイするのは非効率である。パッチに多数のオブジェクトが含まれている場合、パッチの継承、あるいはランタイムアプリケーション内の影響を受けるオブジェクトの数が多いことが原因で、パッチのデプロイが非効率になる可能性があります。

編集したアプリケーションをデプロイする場合は、アプリケーションの状態情報の保持または破棄を選択します。状態情報は、マッピングプロパティ、マッピング出力、およびランタイムオブジェクトのプロパティを参照しています。

アプリケーションを再デプロイする場合は、アプリケーションを停止するよう選択する必要があります。実行中のオブジェクトを強制終了する代わりに、アプリケーションの名前を変更するか、アプリケーションを別のサービスにデプロイすることもできます。

アプリケーションを更新してネットワークファイルシステムにエクスポートする場合、アプリケーションアーカイブファイルを置換するか、デプロイメントを取り消すことができます。アプリケーションアーカイブファイルを上書きする場合は、アプリケーション内のオブジェクトが置換され、オブジェクトのプロパティがリセットされます。

アプリケーションの状態情報

アプリケーションを再デプロイまたはアプリケーションパッチをデプロイすることにより、デプロイ済みアプリケーションを更新するときに、デプロイ済みアプリケーションの状態情報を保持するか破棄するかを選択できます。

状態情報は、マッピングプロパティと、マッピング出力やシーケンスジェネレータトランスフォーメーションなどのランタイムオブジェクトのプロパティを参照しています。状態情報を保持すると、デプロイしたアプリケーションでこれらの設定およびプロパティが保持されます。状態情報を破棄すると、デプロイしたアプリケーションでこれらの設定およびプロパティの状態が破棄されます。

例: 設定可能なプロパティの保持または破棄

アプリケーションには、設定可能なランタイムプロパティを使用したマッピングが含まれます。[高精度] プロパティを True に設定します。アプリケーションをデプロイしたら、マッピングを編集して [高精度] プロパティを False に変更します。アプリケーションを再デプロイし、**【状態情報を保持する】** にチェックを入れると、データ統合サービスはデプロイしたアプリケーションの状態情報を保持し、ランタイムプロパティへの変更が認識されません。プロパティへの変更を有効にするには、**【状態情報を保持する】** のチェックを解除する必要があります。

例: シーケンスの保持または破棄

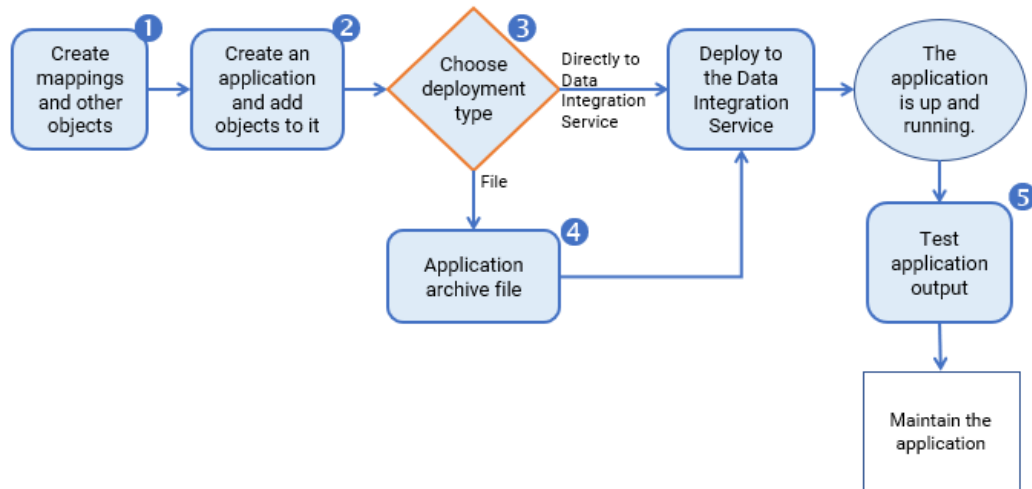
マッピングには、ターゲットテーブルの行に対して一意のキーを生成するシーケンスジェネレータトランスフォーメーションが含まれます。マッピングを含むアプリケーションをデプロイしたら、マッピングが実行され、シーケンスジェネレータトランスフォーメーションは 1~2000 の行に対するキーを生成します。次にマッピングが実行されると、シーケンスジェネレータトランスフォーメーションは 2001 以降のキーを生成しようとしていますが、シーケンスが 10000 から開始されるように設定します。新しい開始値を指定できるようにシーケンスジェネレータトランスフォーメーションを編集します。アプリケーションを再デプロイし、**【状態情報を保持する】** にチェックを入れると、データ統合サービスはデプロイしたアプリケーションの状態情報を保持し、設定への変更が認識されません。マッピングが再度実行されると、シーケンスは 2001 から再開します。設定への変更を有効にするには、**【状態情報を保持する】** のチェックを解除する必要があります。

注: アプリケーションの状態を保持または破棄しても、シーケンスジェネレータトランスフォーメーションの [リセット] プロパティには影響しません。

アプリケーションの作成、デプロイ、更新の方法

マッピングと他のオブジェクトを作成および編集し、アプリケーションにデプロイしてエンドユーザーが使用できるようにします。

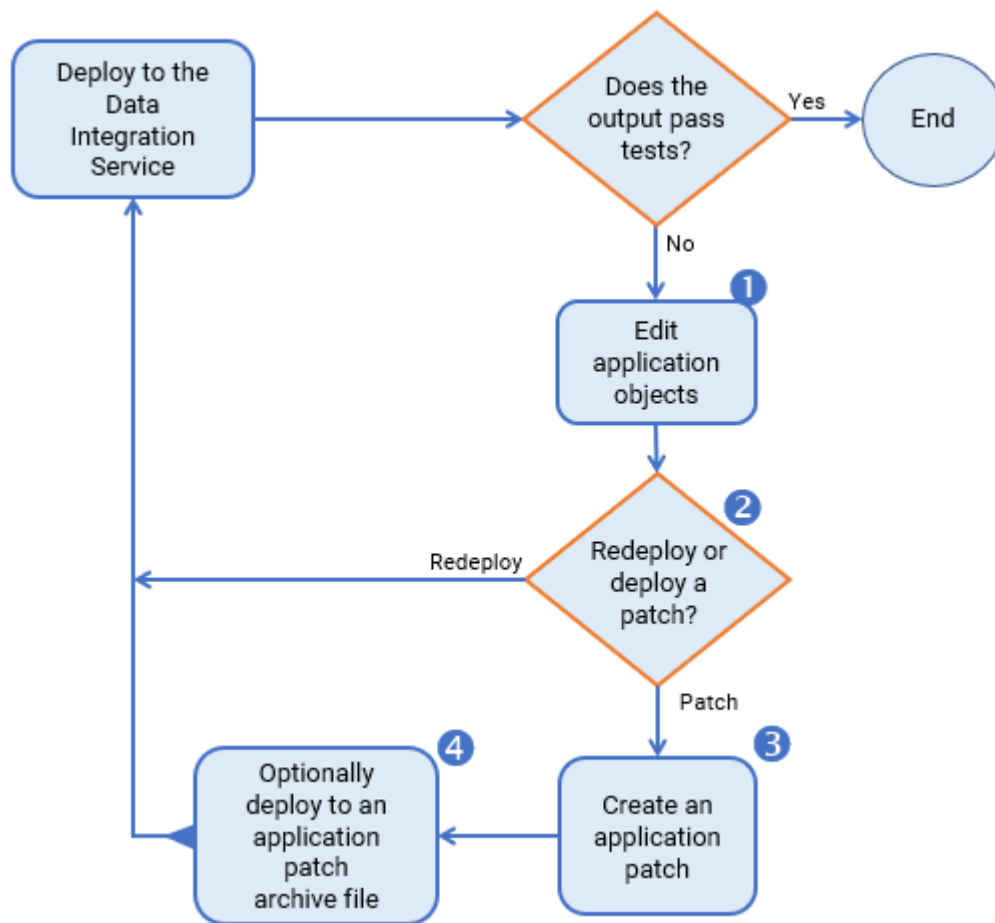
以下の画像に、アプリケーションの開発およびデプロイのプロセスを示します。



1. マッピング、ワークフロー、トランスフォーメーション、および他のオブジェクトを作成し、データにアクセスしてデータを変換します。
2. アプリケーションを作成し、そのアプリケーションにオブジェクトを追加します。
3. アプリケーションをデータ統合サービスに直接デプロイする、またはアプリケーションアーカイブファイルにデプロイするよう選択します。
 - データ統合サービスでオブジェクトを実行できるように、アプリケーションをデータ統合サービスに直接デプロイします。
 - 管理者がファイルをアーカイブできるように、アプリケーションをアプリケーションアーカイブファイルにデプロイします。
4. このアーカイブファイルをデータ統合サービスにデプロイします。
5. アプリケーションをテストして、組織の要件を満たしていることを確認します。

アプリケーションをテストしたら、アプリケーションオブジェクトを編集し、デプロイされたアプリケーションを更新して、アプリケーションをメンテナンスできます。アプリケーションオブジェクトを編集しやすくするために、ランタイムアプリケーションに接続して、ランタイムオブジェクトの設計方法を確認できます。

次の図に、アプリケーションがどのようにメンテナンスされるかを示します。



1. 出力が要件を満たしていない場合、アプリケーション内のオブジェクトを編集します。
2. アプリケーションの再デプロイまたはアプリケーションパッチのデプロイによりアプリケーションのオブジェクトを更新するよう選択します。アプリケーションが差分アプリケーションである場合のみ、アプリケーションパッチをデプロイできます。
3. パッチをデプロイするよう選択する場合、差分デプロイメントウィザードを使用してパッチを作成します。
4. パッチをデータ統合サービスにデプロイし、必要に応じてアプリケーションパッチをアーカイブファイルにデプロイします。

アプリケーションパッチの作成とデプロイ方法の詳細については、「アプリケーションパッチのデプロイメント」の章を参照してください。

注: また、管理者はアプリケーションをデプロイし、コマンドラインまたは Administrator ツールを使用して他の管理タスクを実行できます。これらのタスクの詳細については、『*Informatica アプリケーションサービスガイド*』を参照してください。

アプリケーションの作成

アプリケーションの作成時に、アプリケーションに含めるオブジェクトを選択します。1 つ以上のオブジェクトをデプロイして、エンドユーザーがサードパーティのツールを使用してデータにアクセスできるように、アプリケーションを作成します。

1. **【オブジェクトエクスプローラ】** ビューで、プロジェクトまたはフォルダを選択します。

2. [ファイル] > [新規] > [アプリケーション] をクリックします。

[新しいアプリケーション] ダイアログボックスが表示されます。

Name	Location	Type

3. アプリケーションの名前を入力します。

アプリケーション名は、アプリケーションをデプロイするときのセッションログファイル名の一部です。アプリケーションによっては、セッションログファイル名に、アプリケーション名、機能名、接続 ID、および日付と時刻が含まれる場合があります。

セッションログファイル名は、ログファイルのパスのログディレクトリのパスに含まれています。Windows オペレーティングシステムでは、ログファイルのパスは 259 文字に制限されます。アプリケーションに名前を付ける場合は長さの制限を考慮してください。

4. [参照] をクリックし、アプリケーションの場所を選択します。

アプリケーションはプロジェクトまたはフォルダ内に作成する必要があります。

5. アプリケーションのタイプを選択します。[完全] または [差分] を選択できます。

アプリケーションを作成した後は、アプリケーションタイプを編集することはできません。

6. [追加] ボタンをクリックし、アプリケーションにオブジェクトを追加します。

[オブジェクトの追加] ダイアログボックスが表示されます。

7. 1 つ以上のオブジェクトを選択し、[OK] をクリックします。

選択したオブジェクトが [新しいアプリケーション] ダイアログボックスに一覧表示されます。

8. [完了] をクリックし、アプリケーションを作成します。

アプリケーションを作成したら、そのアプリケーションをデータ統合サービスまたはアプリケーションアーカイブファイルにデプロイします。

アプリケーションのデプロイ

アプリケーションを作成したら、そのアプリケーションをデータ統合サービスにデプロイしてアプリケーションオブジェクトを実行します。または、アプリケーションをアプリケーションアーカイブファイルにデプロイします。

1. **【オブジェクトエクスプローラ】** ビューでアプリケーションを右クリックし、**【デプロイメント】** を選択します。

【デプロイメント】 ダイアログボックスが表示されます。

The dialog box is titled "Deploy" and contains the following elements:

- Deployment Method**: Choose how to deploy the application.
- Application**: MyApplication (Incremental)
- Deploy to Service** (Selected):
 - Domain**: dxt101 (with a **Browse...** button)
 - Available Services**:

	Service	Target Name	Action
<input checked="" type="checkbox"/>	DIS	MyApplication (Running)	Add
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
- Export as application archive file** (Unselected):
 - Location**: (with a **Browse...** button)
- Navigation Buttons**: < Back, Next >, **Finish** (highlighted), Cancel

2. アプリケーションをデプロイする際は次のオプションのいずれかを選択します。
 - アプリケーションをデータ統合サービスにデプロイするには、**【サービスにデプロイ】** を選択します。
 - 必要に応じて、**【参照】** をクリックしてドメインを選択します。
 - アプリケーションをデプロイするデータ統合サービスを選択します。
 - アプリケーションをアプリケーションアーカイブファイルにデプロイするには、**【アプリケーションアーカイブファイルとしてエクスポート】** を選択します。**【参照】** ボタンをクリックし、ファイルの保存先となるディレクトリを選択します。

アプリケーションをデータ統合サービスにデプロイした場合、データ統合サービスはそのアプリケーションとそれに含まれるオブジェクトを実行します。アプリケーションオブジェクトの出力をテストし、要件に対する

出力を検証できます。必要に応じて、アプリケーションオブジェクトを編集し、デプロイされたアプリケーションを更新します。

アプリケーションをアプリケーションアーカイブファイルにデプロイした場合、そのファイルを保存して、別のモデルリポジトリにインポートできるようにしたり、後でそのファイルからアプリケーションをデータ統合サービスにデプロイしたりすることができます。

オブジェクトのデプロイ

オブジェクトをデータ統合サービスにアプリケーションとしてデプロイするか、オブジェクトをアプリケーションアーカイブファイルにデプロイしてオブジェクトをアプリケーションとしてエクスポートします。

1. 実行可能なオブジェクトまたはフラットファイルデータオブジェクトのデプロイを選択します。
 - マッピング、ワークフロー、または他の実行可能なオブジェクトをデプロイするには、オブジェクトを右クリックして **【デプロイメント】** を選択します。
 - フラットファイルオブジェクトをデプロイするには、オブジェクトを右クリックして、オブジェクトを Web サービス、SQL データサービス、REST Web サービスとしてデプロイします。
アプリケーションを作成するように求められます。
2. アプリケーション名を入力します。
3. アプリケーションの場所を選択するには、デフォルトの場所を受け入れるか、**【参照】** をクリックして別の場所を選択します。
4. アプリケーションのタイプを選択します。[完全] または [差分] を選択できます。
フラットファイルオブジェクトを Web サービス、SQL データサービス、REST Web サービスとしてデプロイする場合、アプリケーションは差分アプリケーションにすることはできません。
アプリケーションを作成した後は、アプリケーションタイプを編集することはできません。
5. アプリケーションにオブジェクトを追加するには、[追加] ボタンをクリックしてオブジェクトを選択します。
6. **【次へ】** をクリックします。
7. アプリケーションをデプロイする際は次のオプションのいずれかを選択します。
 - アプリケーションをデータ統合サービスにデプロイするには、**【サービスにデプロイ】** を選択します。
 - Developer tool に複数ドメインの接続情報が含まれている場合、**【参照】** をクリックしてドメインを選択します。
 - アプリケーションをデプロイするデータ統合サービスを選択します。
 - アプリケーションをアプリケーションアーカイブファイルにデプロイするには、**【アプリケーションアーカイブファイルとしてエクスポート】** を選択します。**【参照】** ボタンをクリックし、ファイルの保存先となるディレクトリを選択します。
8. Web サービスにオブジェクトをデプロイするには、以下の手順を実行します。
 - a. Web サービスのプロパティを設定します。
 - b. Web サービスに操作を追加するには、**【次へ】** をクリックします。
デフォルトでは、Web サービスとしてデプロイするオブジェクトごとに操作が作成されます。
 - c. プロパティを表示および設定する各操作、操作入力、および操作出力を選択します。
 - d. **【完了】** をクリックします。
9. SQL データサービスにオブジェクトをデプロイするには、以下の手順を実行します。
 - a. SQL データサービスの名前を入力します。

- b. デフォルトの場所を受け入れるか、または **【参照】** をクリックし、SQL データサービスのモデルリポジトリおよびプロジェクトフォルダの場所を選択します。
 - c. **【次へ】** をクリックします。
【SQL データサービスに仮想テーブルを追加】 ダイアログボックスが表示されます。
 - d. **【追加】** ボタンをクリックします。
 - e. 仮想テーブルの名前を入力します。
 - f. **【データオブジェクト】** カラムの **【開く】** ボタンをクリックします。
【データオブジェクトの選択】 ダイアログボックスが表示されます。
 - g. 物理データオブジェクトを選択し、**【OK】** をクリックします。
 - h. **【仮想スキーマ】** カラムに仮想スキーマの名前を入力します。
 - i. **【データアクセス】** カラムで **【読み取り】** を選択し、仮想テーブルとデータオブジェクトをリンクします。仮想テーブルとデータオブジェクトをリンクしない場合は、**【なし】** を選択します。
10. オブジェクトを REST Web サービスにデプロイするには、以下の手順を実行します。
 - a. REST Web サービスのプロパティを設定します。
 - b. 定義に含めるカラムを選択します。
 - c. HTTP メソッドを選択します。
 11. **【完了】** をクリックします。

Developer tool によって、アプリケーションがデータ統合サービスにデプロイされます。データ統合サービスは、アプリケーションおよびアプリケーションに含まれるオブジェクトを実行します。

アプリケーションオブジェクトの出力をテストし、要件に対する出力を検証できます。必要に応じて、アプリケーションオブジェクトを編集し、デプロイされたアプリケーションを更新します。

アプリケーションアーカイブのインポート

以前にアーカイブファイルにデプロイされたアプリケーションまたはアプリケーションパッチをインポートし、モデルリポジトリでファイルコンテンツを使用できるようにします。

別のモデルリポジトリからアーカイブされたファイルやバージョン管理システムから取得されたファイルをインポートする場合、アーカイブファイルを作成するために使用したアプリケーションとアプリケーションオブジェクトをインポートできます。

1. **【ファイル】** > **【インポート】** をクリックします。
インポートウィザードが表示されます。
2. **【Informatica】** > **【アプリケーションアーカイブ】** を選択します。
3. **【次へ】** をクリックします。
4. **【参照】** をクリックし、アーカイブファイルを選択します。アプリケーションアーカイブファイルをインポートするには、.iar 拡張子を使用します。アプリケーションパッチアーカイブファイルをインポートするには、.piar 拡張子を使用します。
Developer tool でアーカイブファイルのコンテンツがリスト表示されます。
5. アプリケーションのインポート先のモデルリポジトリを選択します。
6. **【完了】** をクリックします。
アプリケーションがリポジトリにインポートされます。

ランタイムアプリケーションへの接続

ランタイムアプリケーションに接続し、[エディタ] を使用してアプリケーション内のランタイムオブジェクトを表示できるようにします。

1. [ファイル] > [ランタイムアプリケーションに接続] をクリックします。
2. ドメインを選択します。
3. データ統合サービスを選択します。
4. データ統合サービスでデプロイ済みアプリケーションを選択します。
5. [OK] をクリックします。

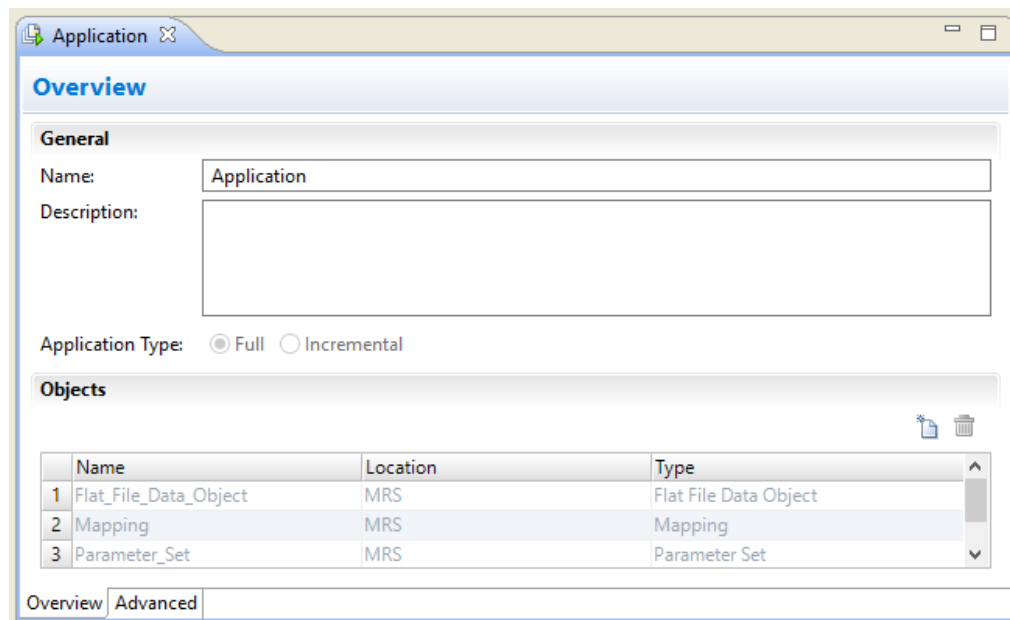
[Object Explorer] ビューにランタイムアプリケーションが表示されます。アプリケーションを展開すると、アプリケーション内のランタイムオブジェクトを表示できます。

アプリケーションの編集

アプリケーションオブジェクトを編集するか、オブジェクトを追加および削除するようにアプリケーションを編集し、アプリケーションのプロパティを編集します。

1. アプリケーションオブジェクトを編集するには、[オブジェクトエクスプローラ] ビューからオブジェクトを開きます。必要な変更を加えます。
2. アプリケーションを編集するには、[オブジェクトエクスプローラ] でアプリケーションを右クリックし、[開く] を選択します。

アプリケーションエディタが表示されます。



3. [追加] ボタンをクリックし、アプリケーションにオブジェクトを追加します。
[オブジェクトの追加] ダイアログボックスが表示されます。
4. 1つ以上のオブジェクトを選択し、[OK] をクリックします。
5. 削除するオブジェクトを1つ以上選択し、[削除] ボタンをクリックします。
6. アプリケーションを保存します。

アプリケーションを編集してから、アプリケーションを更新し、デプロイされたアプリケーションにそれらの変更がプロパゲートされるようにします。

データ統合サービスへのアプリケーションの再デプロイ

アプリケーションをデータ統合サービスに再デプロイして、データ統合サービス上のランタイムアプリケーションを設計時アプリケーションで置き換えます。

1. **【オブジェクトエクスプローラ】** ビューでアプリケーションを右クリックし、**【デプロイ】** をクリックします。

【デプロイメント】 ダイアログボックスが表示されます。

2. **【サービスにデプロイ】** を選択します。
3. デフォルトのデータ統合サービスが設定されていない場合、**【参照】** をクリックしてドメインを選択します。

【ドメインの選択】 ダイアログボックスが表示されます。

4. ドメインを選択して **【OK】** をクリックし、データ統合サービスを選択します。
【ターゲット名】 カラムには、同名のアプリケーションがデフォルトで表示されます。例えば、次の図に、アプリケーション MyApplication のデプロイメントを示します。

Deploy

Deployment Method
Choose how to deploy the application.

Application: MyApplication (Incremental)

☒ Deploy to Service

Domain: **Browse...**

Available Services:

<input checked="" type="checkbox"/>	Service	Target Name	Action
<input checked="" type="checkbox"/>	DIS	MyApplication (Running)	Full update
<input type="checkbox"/>			
<input type="checkbox"/>			

☒ Retain state information
Stop target applications before you deploy an application.

☐ Force the target application to stop, aborting running objects.

☐ Export as application archive file

Location: **Browse...**

Finish **Cancel**

5. **【ターゲット名】** カラムをクリックし、データ統合サービス上で異なるターゲットアプリケーションを選択します。
6. アプリケーションの一部である実行中オブジェクトの状態を保持するには、**【状態情報を保持する】** を選択します。

7. 必要に応じて、**[ターゲットアプリケーションを強制停止して実行中のオブジェクトを強制終了する]** を選択します。

アプリケーションの状態が「実行中」でない場合に限り、アプリケーションを停止せずに再デプロイできます。管理者がまだアプリケーションを停止していない場合は、管理者にアプリケーションを強制停止するかどうかを確認してください。

8. **[完了]** をクリックします。

アプリケーションを再デプロイしたら、組織またはプロジェクトの要件と比較して検証することができます。

第 10 章

アプリケーションパッチのデプロイメント

この章では、以下の項目について説明します。

- [アプリケーションパッチのデプロイメントの概要, 167 ページ](#)
- [アプリケーションオブジェクトインスタンス, 168 ページ](#)
- [アプリケーションオブジェクトの依存関係, 168 ページ](#)
- [差分デプロイメントウィザード, 172 ページ](#)
- [アプリケーションパッチのタイプの選択, 175 ページ](#)
- [アプリケーションオブジェクトの影響の決定, 181 ページ](#)
- [パッチデプロイメント戦略の指定, 185 ページ](#)
- [アプリケーションパッチのデプロイ, 187 ページ](#)
- [アプリケーションパッチのデプロイメントに関するルールとガイドライン, 189 ページ](#)

アプリケーションパッチのデプロイメントの概要

アプリケーションパッチは、デプロイされた差分アプリケーション内のオブジェクトを更新するために使用できるデプロイ可能なオブジェクトです。

アプリケーションパッチを設計する際には、アプリケーションオブジェクトの設計時インスタンスとランタイムインスタンス、アプリケーションオブジェクト間の依存関係を考慮し、作成するアプリケーションパッチのタイプを決定する必要があります。一部のアプリケーションオブジェクトが、更新する他のオブジェクトと関係がある場合があります。また、他のアプリケーションオブジェクトが、更新するオブジェクトによって影響を受ける場合があります。

パッチタイプを使用して、関係するオブジェクトをどのように継承するかを選択し、デプロイ済みのアプリケーションに対してパッチでどの程度の影響を与えるかを決定できます。パッチの継承とオブジェクトの影響を管理するために、差分デプロイメントウィザード内のツールを使用してパッチのコンテンツを把握し、データ統合サービスがそれらのコンテンツをデプロイされたアプリケーションに適用する方法を把握できます。

注: アプリケーションパッチを使用して次のタスクを実行することはできません。

- アプリケーションの説明、デフォルトの日時形式、ソート順など、アプリケーションおよびマッピングのデプロイメントのプロパティを更新する。
- Web サービス、データサービス、REST Web サービスをデプロイされたアプリケーションに追加する。

- デプロイされたアプリケーションからアプリケーションオブジェクトを削除する。

アプリケーションオブジェクトインスタンス

アプリケーション内のアプリケーションオブジェクトは、設計時インスタンスとランタイムインスタンスを持ちます。

アプリケーションオブジェクトの設計時インスタンスとは、開発者が Developer tool で作成し編集したアプリケーションオブジェクトを表します。このオブジェクトを含むアプリケーションをデータ統合サービスにデプロイするときに、このアプリケーションオブジェクトのランタイムインスタンスを生成します。オブジェクトのランタイムインスタンスには、設計時インスタンスのデータ構造の変更は反映されません。そのため、ランタイムオブジェクトを変更することなく、Developer tool で引き続き設計時オブジェクトを編集できます。

差分デプロイメントウィザードは、アプリケーションオブジェクトの設計時インスタンスとランタイムインスタンスの両方を使用し、デプロイされたアプリケーションにパッチコンテンツを適用する方法を決定します。例えば、このウィザードは選択したオブジェクトの設計時インスタンスとランタイムインスタンスの依存関係セットに基づいて、パッチが他のオブジェクトをどのように継承するかを決定し、アプリケーションオブジェクトのランタイムインスタンスを取得して、ランタイムインスタンスへの影響を決定します。

アプリケーションオブジェクトの依存関係

アプリケーションパッチに含めるよう選択したアプリケーションオブジェクトが、そのオブジェクトの依存関係セットに基づいて他のアプリケーションオブジェクトと関係がある場合があります。

アプリケーション内のオブジェクトには次の依存関係を含めることができます。

直接

直接依存関係とは、選択したオブジェクトが使用するアプリケーションオブジェクトであり、その直接依存関係に対するあらゆる直接依存関係も含まれます。

以下の表に、各オブジェクトタイプの直接依存関係を示します。

オブジェクト	可能な直接依存関係
ワークフロー	<ul style="list-style-type: none">- Mapping- マップレット- 再利用可能なトランスフォーメーション- 論理データオブジェクト- 物理データオブジェクト
Mapping	<ul style="list-style-type: none">- マップレット- 再利用可能なトランスフォーメーション- 論理データオブジェクト- 物理データオブジェクト
マップレット	<ul style="list-style-type: none">- 再利用可能なトランスフォーメーション- 論理データオブジェクト- 物理データオブジェクト

オブジェクト	可能な直接依存関係
再利用可能なトランスフォーメーション	<ul style="list-style-type: none"> - 論理データオブジェクト - 物理データオブジェクト
論理データオブジェクト	<ul style="list-style-type: none"> - 物理データオブジェクト
物理データオブジェクト	<ul style="list-style-type: none"> - 物理データオブジェクトは直接依存関係を持つことはできません。
パラメータセット	<ul style="list-style-type: none"> - パラメータセットは直接依存関係を持ったり、直接依存関係になることはできません。

間接

間接依存関係とは、選択したオブジェクトが使用するアプリケーションオブジェクトであるか、または選択したオブジェクトの直接依存関係を使用するアプリケーションオブジェクトです。間接依存関係には、その間接依存関係に対するあらゆる直接依存関係も含まれます。

リモート

リモート依存関係とは、選択したオブジェクトの間接依存関係の間接依存関係であるアプリケーションオブジェクトです。リモート依存関係には、そのリモート依存関係に対するあらゆる直接依存関係も含まれます。

アプリケーションパッチは、選択したオブジェクトの依存関係に基づいて、他のアプリケーションオブジェクトを継承します。例えば、選択したオブジェクトの直接依存関係と間接依存関係を継承するアプリケーションパッチを作成できます。

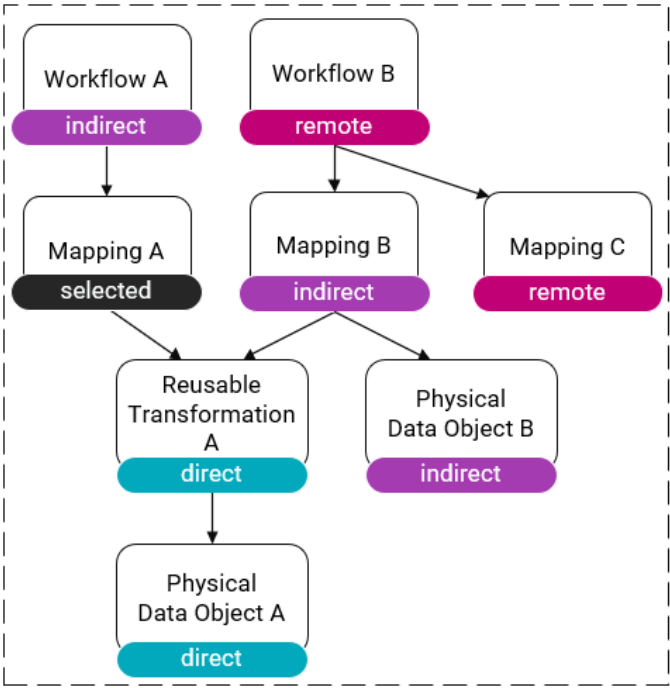
アプリケーションパッチでマップレット、再利用可能なトランスフォーメーション、論理データオブジェクトを選択することはできませんが、パッチはこれらのオブジェクトを継承できます。

間接およびリモート依存関係の特定方法については、次の Informatica How-To Library 記事を参照してください。

[Identifying Indirect and Remote Dependencies for an Application Patch in the Developer Tool](#)

例: 依存関係セット

次の図に、アプリケーションオブジェクト Mapping A の依存関係セットを示します。



以下の表に、マッピング Mapping A の依存関係セットを示します。

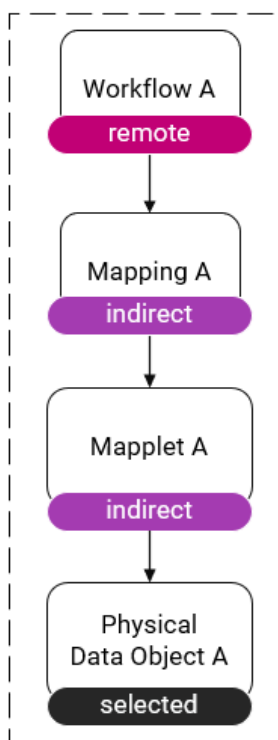
依存タイプ	オブジェクト	説明
直接	Reusable Transformation A	マッピング Mapping A は再利用可能なトランスフォーメーションを使用するため、この再利用可能なトランスフォーメーションは直接依存関係です。
直接	Physical Data Object A	マッピング Mapping A は再利用可能なトランスフォーメーション Reusable Transformation A を通して最終的にこのデータオブジェクトを使用するため、この物理データオブジェクトは直接依存関係です。
間接	Workflow A	このワークフローはマッピング Mapping A を使用するため、このワークフローは間接依存関係です。
間接	Mapping B	マッピング Mapping B はマッピング Mapping A と再利用可能なトランスフォーメーション Reusable Transformation A の直接依存関係を使用するため、マッピング Mapping B は間接依存関係です。
間接	Physical Data Object B	この物理データオブジェクトはマッピング Mapping B に対する直接依存関係であり、このマッピングはマッピング Mapping A に対する間接依存関係であるため、この物理データオブジェクトは間接依存関係です。

依存タイプ	オブジェクト	説明
リモート	Workflow B	このワークフローはマッピング Mapping B に対する間接依存関係であり、このマッピングはマッピング Mapping A に対する間接依存関係であるため、このワークフローはリモート依存関係です。
リモート	Mapping C	このマッピングはワークフロー Workflow B に対する直接依存関係であり、このワークフローはマッピング Mapping A に対するリモート依存関係であるため、マッピング Mapping C はリモート依存関係です。

関係の優先度

物理データオブジェクト、マッピング、ワークフロー間の関係は、他のタイプのオブジェクト間の関係よりも優先されます。

例えば、次の図はデータオブジェクト Physical Data Object A の依存関係セットを示します。



マッピング Mapping A とデータオブジェクト Physical Data Object A 間の関係は次の関係よりも優先されるため、マッピング Mapping A は間接依存関係です。

- データオブジェクト Physical Data Object A とマップレット Maplet A 間の関係。
- マップレット Maplet A とマッピング Mapping A 間の関係。

マッピング Mapping A は間接依存関係であり、その直接依存関係である Maplet A はデータオブジェクト Physical Data Object A に対する間接依存関係です。

差分デプロイメントウィザード

差分デプロイメントウィザードとは、アプリケーションパッチを作成するために使用するウィザードです。

ウィザードを実行するには、差分アプリケーションを右クリックし、**[アプリケーションオブジェクトの更新]**を選択します。

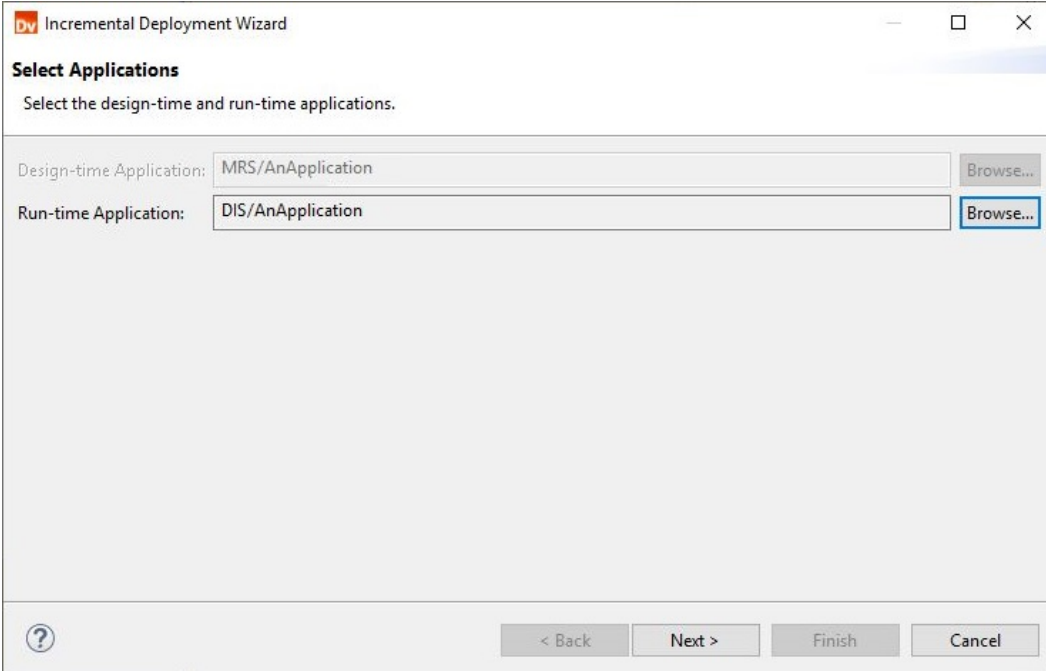
アプリケーションパッチを作成するには、ウィザードの以下のページを使用します。

1. **[アプリケーションの選択]** ページで、更新するランタイムアプリケーションを選択します。
2. **[ソースオブジェクトの選択]** ページで、パッチに含めるオブジェクトを選択します。ランタイムアプリケーションに対してパッチがどんな影響をどの程度及ぼすかをプレビューすることもできます。
3. **[パッチのデプロイメント戦略]** ページで、パッチに関する追加の詳細を定義して、パッチのデプロイ方法を決定します。

アプリケーションの選択

[アプリケーションの選択] ページで、更新するランタイムアプリケーションを選択します。

次の図は、**[アプリケーションの選択]** ページを示しています。



The image shows a screenshot of the 'Incremental Deployment Wizard' window, specifically the 'Select Applications' page. The window title is 'Incremental Deployment Wizard'. Below the title bar, the page is titled 'Select Applications' with the instruction 'Select the design-time and run-time applications.' There are two text input fields: 'Design-time Application:' with the value 'MRS/AnApplication' and a 'Browse...' button to its right; and 'Run-time Application:' with the value 'DIS/AnApplication' and a 'Browse...' button to its right. At the bottom of the window, there is a navigation bar with a help icon (question mark), and four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

ソースオブジェクトの選択

【ソースオブジェクトの選択】 ページで、作成するパッチのタイプとパッチに含めるオブジェクトを選択します。

次の図は、【ソースオブジェクトの選択】 ページを示しています。

Incremental Deployment Wizard

Source Object Selection

Select the objects in the design-time application that you want to add to the patch.

Include selected objects and.. Direct and indirect dependencies

Search by object name

<input checked="" type="checkbox"/> Object ^	Design-time Location	Related Objects
<input checked="" type="checkbox"/> MappingA	/MappingA	MappingB, WorkflowA
<input checked="" type="checkbox"/> MappingB	/MappingB	MappingA, WorkflowA
<input checked="" type="checkbox"/> MappingC	/MappingC	
<input checked="" type="checkbox"/> WorkflowA	/WorkflowA	MappingA, MappingB

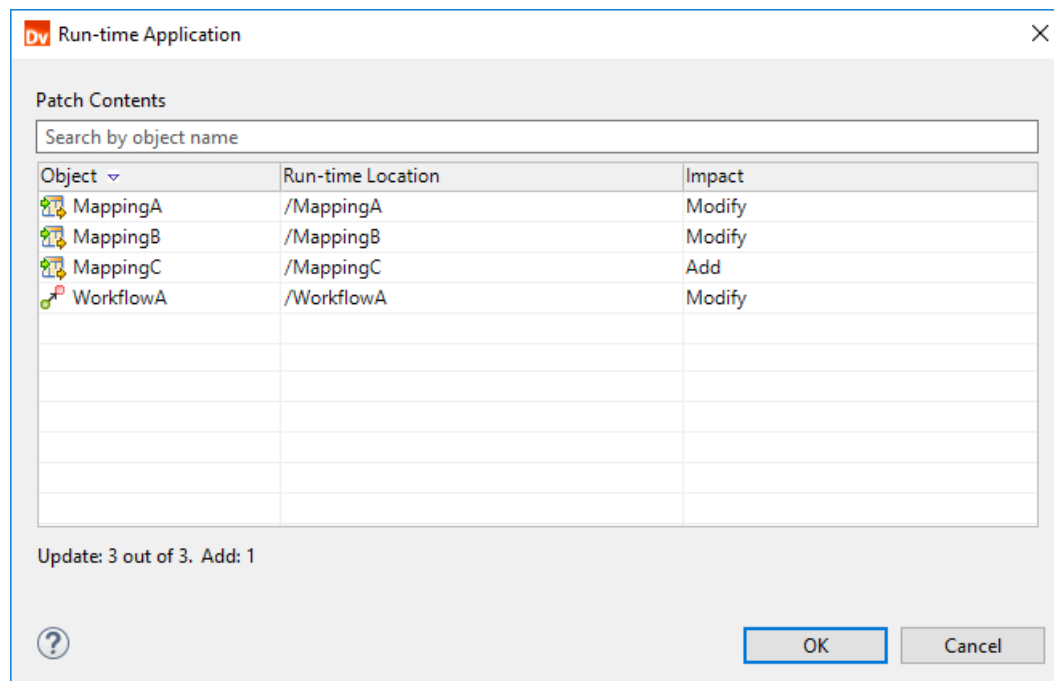
4 out of 4 selected. Update: 3 out of 3. Add: 1 Preview Application...

? < Back Next > Finish Cancel

各オブジェクトの【設計時の場所】 カラムを使用して、設計時アプリケーション内のオブジェクトのパスを確認できます。ウィザードでは、パッチタイプに応じて、【関連オブジェクト】 カラムにオブジェクトの依存関係がリスト表示されます。例えば、直接依存関係と間接依存関係を継承するパッチタイプの場合は、ウィザードでは各オブジェクトの直接依存関係と間接依存関係がリスト表示されます。

ページの右下では、ランタイムオブジェクトへの影響のサマ리를プレビューできます。【更新】の統計には、変更されるオブジェクトと影響を受けるオブジェクトの合計数が表示されます。【追加】の統計には、追加されるオブジェクトの合計数が表示されます。特定のランタイムオブジェクトの影響を確認するには、アプリケーションをプレビューできます。

次の図に、アプリケーションのプレビューを示します。

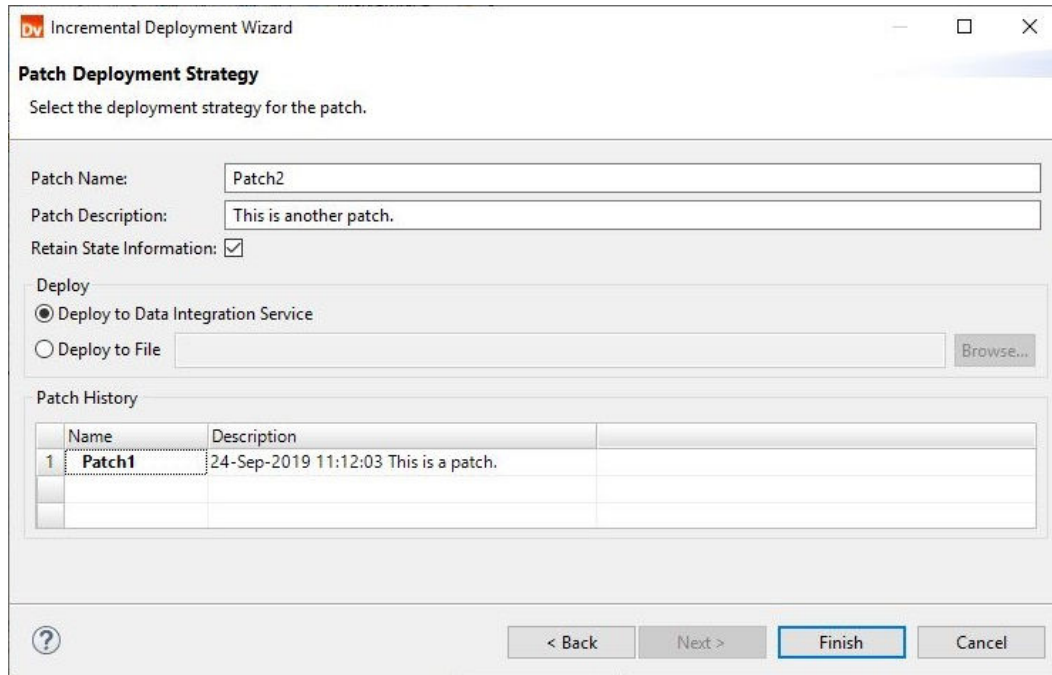


オブジェクト名を照合し、実行時の場所と設計時の場所を照合すると、プレビュー内のランタイムオブジェクトのリストと設計時オブジェクトを照合できます。各オブジェクトの「影響」には、そのオブジェクトが変更されるか、影響を受けるか、追加されるかどうか、または変更されないかがリスト表示されます。

パッチのデプロイメント戦略

【パッチのデプロイメント戦略】 ページで、パッチを説明し、そのデプロイ方法を定義して、パッチの状態情報を保持するかどうかを選択します。

次の図は、【パッチのデプロイメント戦略】 ページを示しています。



The dialog box is titled "Incremental Deployment Wizard" and "Patch Deployment Strategy". It contains the following fields and controls:

- Patch Name:** Text box containing "Patch2".
- Patch Description:** Text box containing "This is another patch."
- Retain State Information:** Check box, which is checked.
- Deploy:** Radio button group with two options:
 - ☒ Deploy to Data Integration Service
 - ☐ Deploy to File
- Patch History:** A table with two columns: "Name" and "Description". It contains one row with the following data:

	Name	Description
1	Patch1	24-Sep-2019 11:12:03 This is a patch.

At the bottom, there are navigation buttons: "< Back", "Next >", "Finish" (highlighted with a blue border), and "Cancel". A help icon (?) is located on the bottom left.

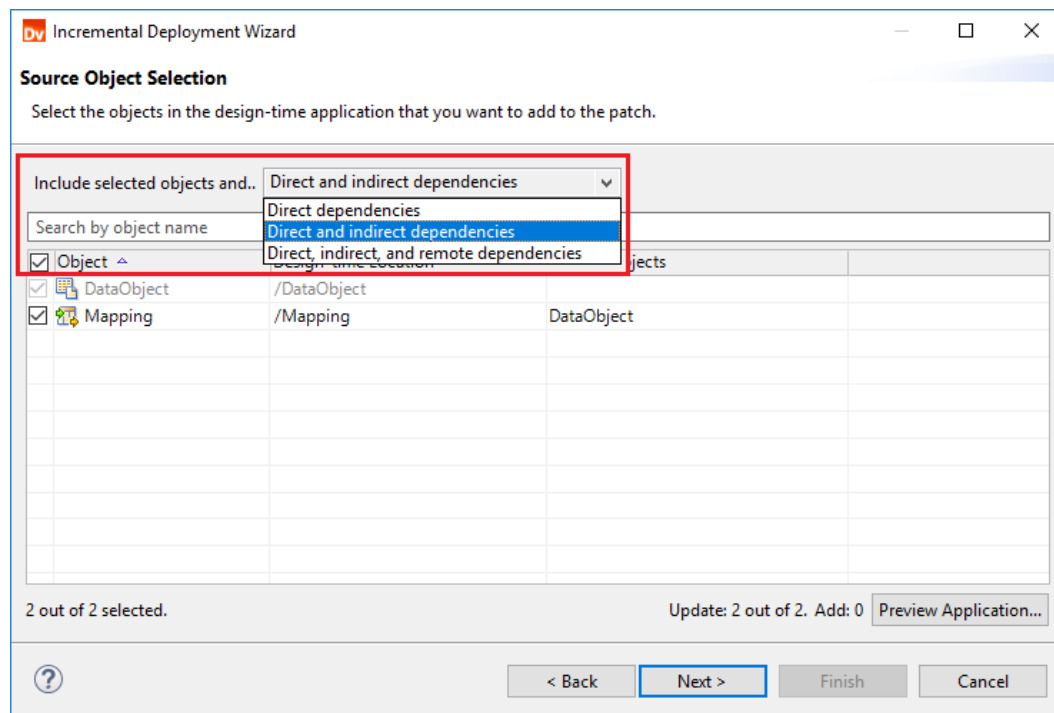
デプロイメント戦略を選択するときは、パッチ履歴を使用して、ランタイムアプリケーションを更新するためにデプロイされたパッチを確認することもできます。

アプリケーションパッチのタイプの選択

差分デプロイメントウィザードの使用時に、作成するアプリケーションパッチのタイプを選択できます。アプリケーションパッチのタイプによって、パッチが継承する依存関係のレベルが決まります。

パッチは直接依存関係、間接依存関係、リモート依存関係を継承できます。依存関係の各レベルで、パッチはより多くのオブジェクトを継承します。継承されたオブジェクトはパッチに含められ、パッチがランタイムアプリケーションに与える全体的な影響が大きくなります。アプリケーションの設計や更新するアプリケーションオブジェクトに応じて、継承する依存関係を選択できます。

次の図に、ウィザードのパッチタイプのリストを示します。

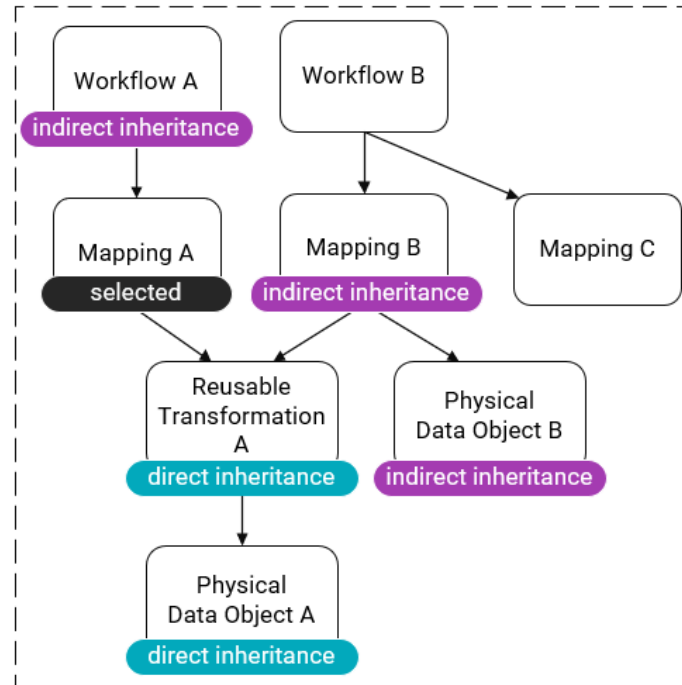


パッチのデプロイ時に、選択したオブジェクトのランタイムインスタンスと、ランタイムアプリケーション内の継承された依存関係を、データ統合サービスが置き換えます。これによりランタイムオブジェクトに影響が及ぶ可能性があります。データ統合サービスは影響を受けるオブジェクトを、置き換えられたランタイムオブジェクトを使用するように更新しますが、影響を受けるオブジェクトは設計時インスタンスには置き換えられません。

アプリケーションに高依存オブジェクトが含まれる場合、アプリケーションパッチが継承する依存関係には、ランタイムアプリケーション内の影響を受けるオブジェクトの数と逆の相関関係が生まれます。アプリケーションパッチが継承する依存関係が少ない場合に、ランタイムアプリケーションに影響を受けるオブジェクトが多く含まれる可能性が発生します。

例: 依存関係の継承

次の図に、選択したオブジェクトがマッピング Mapping A である場合にパッチが継承する直接依存関係と間接依存関係を示します。



パッチは以下の直接依存関係と間接依存関係を継承します。

- 直接依存関係。パッチは再利用可能なトランスフォーメーション Reusable Transformation A とデータオブジェクト Physical Data Object A を継承します。これらのオブジェクトは選択したマッピング Mapping A に対する直接依存関係であるためです。
- 間接依存関係。パッチはマッピング Mapping A とデータオブジェクト Physical Data Object B を継承します。これらのオブジェクトは選択したマッピング Mapping A に対する間接依存関係であるためです。

このパッチはワークフロー Workflow B とマッピング Mapping C は継承しません。これらのオブジェクトは選択したマッピング Mapping A に対するリモート依存関係であるためです。

直接依存関係のみの継承

設計時アプリケーションとランタイムアプリケーションについての完全な情報を持っていない場合は、直接依存関係のみを継承するパッチを作成すると、ランタイムアプリケーション内のオブジェクトへの影響を最小限に抑えることができます。

例

他の 30 人のユーザーとアプリケーションを共有しています。更新が必要なアプリケーションオブジェクトは、Kafka データオブジェクトからデータを読み取る 1 個のストリーミングマッピングです。ランタイムアプリケーション内のマッピングをテストした後、Developer tool でマッピングを編集し、パッチをデプロイしてランタイムアプリケーション内のマッピングを更新します。

アプリケーションパッチを作成するときに、マッピングを選択し、直接依存関係のみを継承するように選択します。マッピングを選択すると、パッチはこのマッピングが使用するデータオブジェクトを継承します。

次の図に、差分デプロイメントウィザードに継承がどのように表示されるかを示します。

Incremental Deployment Wizard

Source Object Selection

Select the objects in the design-time application that you want to add to the patch.

Include selected objects and..

Direct dependencies ▼

Search by object name

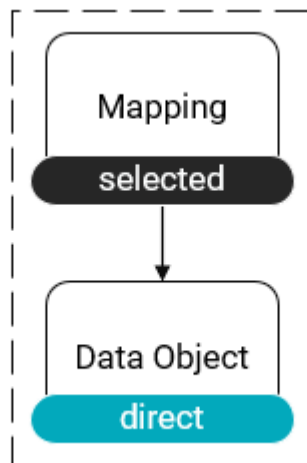
<input checked="" type="checkbox"/> Object ▲	Design-time Location	Related Objects	
<input checked="" type="checkbox"/> DataObject	/DataObject		
<input checked="" type="checkbox"/> Mapping	/Mapping	DataObject	

2 out of 2 selected.

Update: 2 out of 2. Add: 0 Preview Application...

?< BackNext >FinishCancel

次の図に、選択されたオブジェクトと継承された直接依存関係を示します。



直接および間接の依存関係の継承

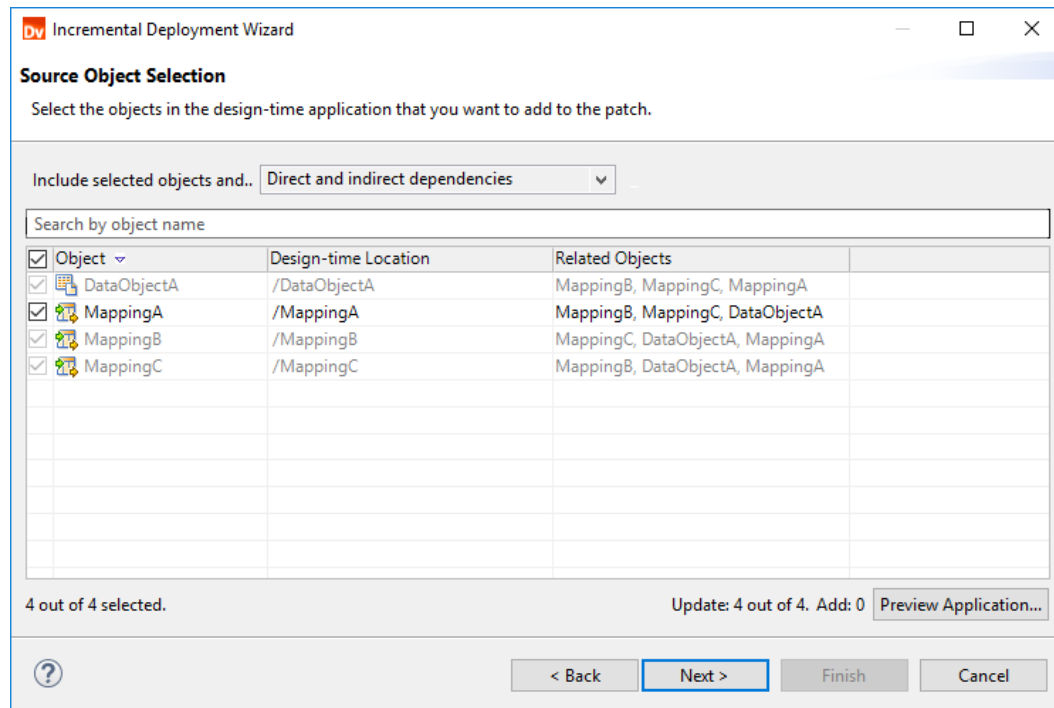
多数のその他のアプリケーションオブジェクトで使用されているアプリケーションオブジェクトを編集するときに、直接依存関係と間接依存関係を継承し、変更が親オブジェクトにプロパゲートされるように選択できます。

例

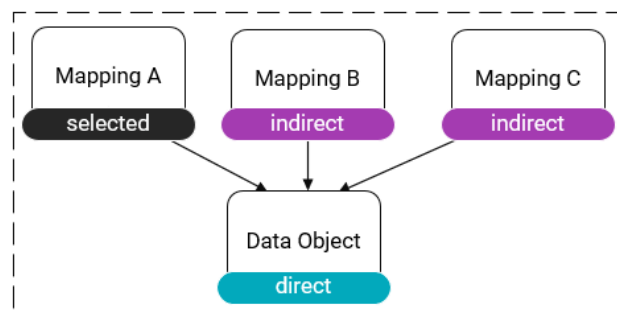
同じリレーショナルデータオブジェクトを使用する複数のマッピングを含むアプリケーションがあるとして、各マッピングは最終的に、別の目的でそのデータオブジェクト内のデータを変換します。マッピングの実行をスケジュールするには、アプリケーションをデータ統合サービスにデプロイします。

その後で、Developer tool を使用して、リレーショナルデータオブジェクトとリレーショナルデータベースを同期します。同期されたデータをランタイムアプリケーション内のマッピングにプロパゲートするために、直接依存関係と間接依存関係を継承するパッチを作成します。

パッチを作成するときに、そのデータオブジェクトを使用するマッピングのいずれか 1 つを選択します。パッチはそのデータオブジェクトを使用する他のすべてのマッピングを継承します。次の図に、差分デプロイメントウィザードに継承がどのように表示されるかを示します。



次の図に、選択されたオブジェクトと継承された間接依存関係を示します。



直接、間接、およびリモートの依存関係の継承

設計時アプリケーションとランタイムアプリケーションに関する完全な情報を持っている場合は、直接、間接、およびリモートの依存関係を継承できます。

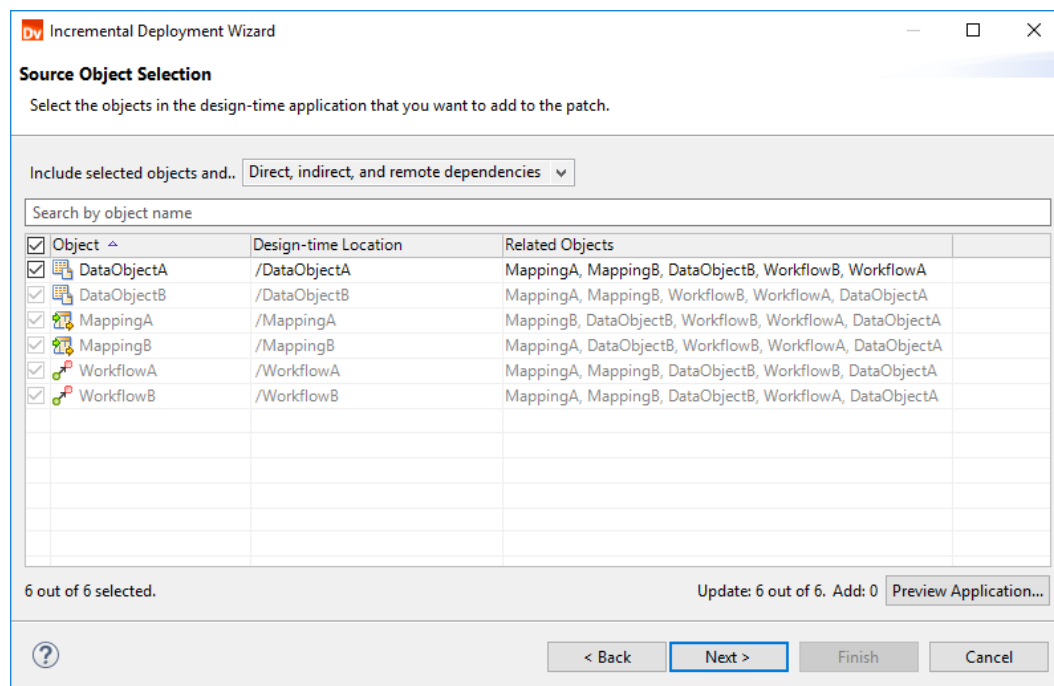
アプリケーションの唯一の開発者または担当部門の管理者がアプリケーションに関する完全な情報を持っていて、設計時アプリケーションを変更する際に、それらすべての変更をランタイムアプリケーションにプロパゲートし、ランタイムアプリケーションが設計時アプリケーションをミラーリングするようにしたい場合などがこれに該当します。

例

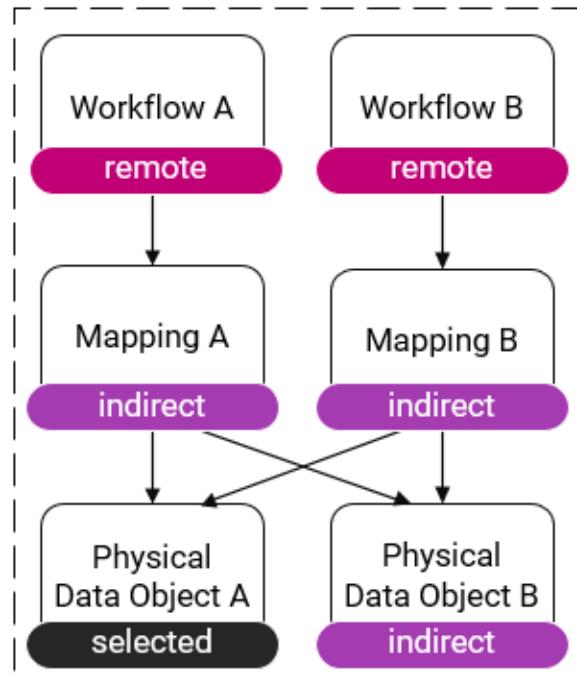
別のデータオブジェクトを共有するマッピングを実行するワークフローが含まれるアプリケーションを管理しています。アプリケーションをデプロイして出力をテストした後、データオブジェクトのいずれかを編集します。

データオブジェクトの変更を、そのデータオブジェクトを使用するその他すべてのアプリケーションオブジェクトにプロパゲートするには、直接、間接、およびリモートの依存関係を継承するパッチを作成します。そのデータオブジェクトを選択するときに、パッチがそのデータオブジェクトを使用するすべてのワークフローとマッピングを継承します。

次の図に、差分デプロイメントウィザードに継承がどのように表示されるかを示します。



次の図に、選択されたオブジェクトと継承された直接、間接、およびリモートの依存関係を示します。



注: データオブジェクト Physical Data Object A の変更を、ランタイムアプリケーション内のそのデータオブジェクトを使用するすべてのマッピングとワークフローにプロパゲートするために、直接依存関係のみを継承するパッチを使用することもできます。データオブジェクトを選択した場合、ウィザードによって、影響を受けるマッピングとワークフローが特定され、データ統合サービスがパッチのデプロイメント時にマッピングとワークフローを更新します。

ただし、直接、間接、およびリモートの依存関係を継承するパッチでは、オブジェクトがどのように更新されるかがより明確に示されます。また、ランタイムアプリケーション内の更新されるオブジェクトが、設計時アプリケーション内のオブジェクトと同じ方法でデータを変換することがより高い確率で保証されます。

アプリケーションオブジェクトの影響の決定

アプリケーションパッチを作成してデプロイするときに、パッチがランタイムアプリケーションに与える影響が、変更するオブジェクトや影響を受けるオブジェクトにまで拡大します。複数のユーザーでアプリケーションを共有している場合、パッチは他のユーザーがメンテナンスしているオブジェクトに影響を与える可能性があります。

パッチがランタイムアプリケーションにどの程度の影響を与えるのかを把握するために、次のタスクを実行できます。

- オブジェクトの影響をプレビューする。オブジェクトの影響は、各アプリケーションオブジェクトがどのように影響を受けるのかを示したサマリになります。
- 影響を受けるオブジェクトを特定する。影響を受けるオブジェクトは、他のユーザーに属しているオブジェクトである場合があります。影響を受けるオブジェクトを特定することで、他のユーザーのオブジェクトがどのように変更される可能性があるか把握できます。
- パッチの範囲を確認する。範囲は、パッチにより影響を受けるオブジェクトの数を示した数値のサマリになります。

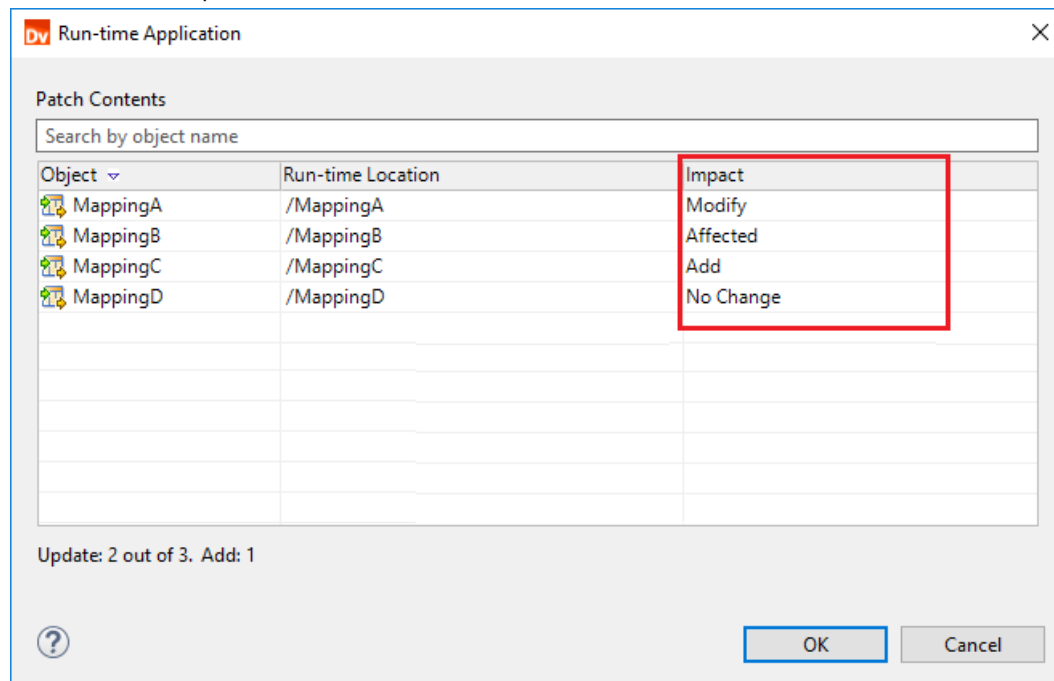
オブジェクトの影響のプレビュー

差分デプロイメントウィザードは、アプリケーションパッチにオブジェクトが追加され、パッチがそれらのオブジェクトの依存関係を継承している場合のランタイムアプリケーションへの影響を分析します。各アプリケーションオブジェクトへの影響をプレビューして、パッチのデプロイ後にオブジェクトのランタイムインスタンスがどのように変わるかを把握できます。

プレビューでは、各アプリケーションオブジェクトに対する次のタイプの影響がリスト表示されます。

- 変更。ランタイムアプリケーション内にオブジェクトのランタイムインスタンスが存在しています。データ統合サービスはランタイムインスタンスを設計時インスタンスで置き換えます。
- 影響。オブジェクトの依存関係が変更されるため、ウィザードはランタイムアプリケーションからオブジェクトを取得します。データ統合サービスは、ランタイムアプリケーション内の影響を受けるオブジェクトが、変更された依存関係を使用するように更新しますが、オブジェクトは置き換えません。
- 追加。ランタイムアプリケーション内にオブジェクトのランタイムインスタンスが存在しません。そのため、データ統合サービスはランタイムアプリケーション内にオブジェクトのランタイムインスタンスを作成します。
- 変更なし。オブジェクトは変更されたり、影響を受けたり、ランタイムアプリケーションに追加されたりしません。

次の図に、Developer tool でオブジェクトの影響をプレビューできる場所を示します。



影響を受けるオブジェクトの特定

影響を受けるオブジェクトとは、デプロイされたアプリケーションをメンテナンスするために差分デプロイメントウィザードがランタイムアプリケーションから取得するオブジェクトです。影響を受けるオブジェクトを所有するユーザーは、データ統合サービスがパッチを適用している間はそれらのオブジェクトを実行できません。

ウィザードは、ユーザーがアプリケーションパッチを設計するときに、影響を受けるオブジェクトを取得します。オブジェクトの影響をプレビューすると、影響を受けるオブジェクトを確認できます。

ウィザードは、次の理由により、影響を受けるオブジェクトを取得します。

変更のプロパゲート

アプリケーションパッチ内のオブジェクトを選択し、そのパッチが他のアプリケーションオブジェクトを継承している場合、ウィザードはそれらのオブジェクトの変更がランタイムアプリケーション内の親オブジェクトにどのようにプロパゲートされるかを決定する必要があります。ウィザードは、変更の影響を受ける親オブジェクトを特定します。

パッチをデプロイするとき、データ統合サービスは影響を受けるオブジェクトのランタイムインスタンスを置き換えません。しかし、影響を受けるオブジェクトを、それらの実行時の直接依存関係に対する変更を使用するように更新します。

注: ワークフローが影響を受けるオブジェクトである場合、ワークロードは次の実行時に更新されます。例えば、マッピングを変更し、マッピングタスクでこのマッピングを使用するワークフローを変更した場合、このワークフローは以降の実行でこの変更されたマッピングを使用します。

有効性

ウィザードは、データ統合サービスがパッチを適用した後もランタイムアプリケーション内のアプリケーションオブジェクトが有効であるか確認する必要があります。ウィザードは、影響を受けるオブジェクトに変更がどのようにプロパゲートされるかを決定した後、影響を受けるオブジェクトを検証します。影響を受けるオブジェクトが変更されたオブジェクトと互換性がない場合、影響を受けるオブジェクトが有効でなくなる可能性があります。

どのアプリケーションパッチのタイプでも、影響を受けるオブジェクトになる可能性があります。例えば、直接、間接、リモートの依存関係を継承するパッチを作成する場合、リモート依存関係がパッチが継承する直接依存関係を持っている可能性があります。この直接依存関係がランタイムアプリケーション内の別のオブジェクトで使用されている場合、親オブジェクトが影響を受けるオブジェクトになります。

データ統合サービスが影響を受けるオブジェクトを更新した後で、影響を受けるオブジェクトのデータがアプリケーションの設計と一貫性がない可能性があります。マップレットを使用するマッピングを含むランタイムアプリケーションがあるとして。変更されたマップレットが別のトランスフォーメーションを使用する場合、影響を受けるマッピングは有効である可能性があります。予測できない方法でデータを変換する可能性もあります。

可能な限り、更新するオブジェクトを選択するか、または影響を受けるオブジェクトの数が最小限になり、ランタイムアプリケーション内のオブジェクトが設計時アプリケーション内のオブジェクトと同じ方法でデータを変換することが保証されるようにオブジェクトを継承するアプリケーションパッチを設計するようにしてください。

例: 変更のプロパゲートと有効性

データオブジェクトを変更するパッチをデプロイします。ウィザードはデータオブジェクトの変更を、そのデータオブジェクトを使用するその他のランタイムオブジェクトにプロパゲートします。その結果、そのデータオブジェクトを使用する、影響を受けるランタイムマッピングは、データ統合サービスがパッチを適用した後は、変更されたデータオブジェクトを使用します。

マッピングがこの変更されたデータオブジェクトを使用し、データオブジェクトに含まれる行数がパッチ適用前と比べて少ない場合、マッピングが有効でなくなる可能性があります。アップストリームトランスフォーメーションとダウンストリームトランスフォーメーション間のリンクの数が少なくなるため、データオブジェクトはマッピング内のトランスフォーメーションと互換性がない可能性があります。影響を受けるマッピングが有効でない場合、パッチをデプロイすることはできません。

削除したオブジェクト

設計時アプリケーションからアプリケーションオブジェクトを削除してパッチをデプロイしても、データ統合サービスはランタイムアプリケーションからそのオブジェクトを削除しません。場合によっては、影響を受けるオブジェクトを削除している可能性があります。

このウィザードが、削除したオブジェクトに影響を受けるオブジェクトとして特定した場合、ウィザードは削除したオブジェクトの子オブジェクトの変更を、削除したオブジェクトにプロパゲートするよう試みます。ま

た、ウィザードは削除したオブジェクトの検証も試みます。削除したオブジェクトが有効でない場合、パッチをデプロイすることはできません。

削除したオブジェクトが実行中でない場合は、アプリケーションを再デプロイしてランタイムアプリケーションからオブジェクトを削除してください。

例.影響を受けるオブジェクトの特定

次の例は、差分デプロイメントウィザードがランタイムアプリケーションから取得した、影響を受けるオブジェクトを特定する方法を示しています。

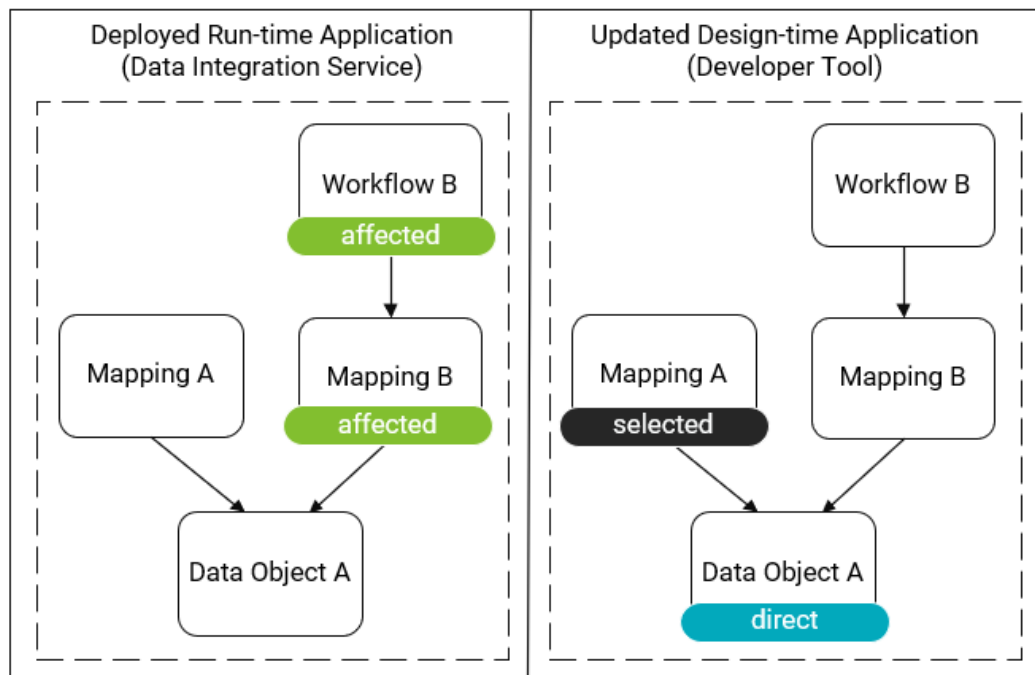
アプリケーションに2つのマッピング Mapping A と Mapping B が含まれていて、それらのマッピングが物理データオブジェクト Data Object A を共有しているとします。また、アプリケーションにはマッピング Mapping B を使用するワークフロー Workflow B も含まれています。このアプリケーションをデプロイした後で、マッピング Mapping A を編集し、ランタイムアプリケーション内のこのマッピングを更新することにしました。

直接依存関係のみを継承するパッチを作成し、マッピング Mapping A を選択します。マッピングを選択しているため、パッチはこのマッピングが使用するデータオブジェクト Data Object A を継承します。

ウィザードはデータ統合サービスがこのマッピングとこのデータオブジェクトのランタイムインスタンスを置き換えることを想定しているため、ウィザードはランタイムアプリケーション内でこのマッピングまたはこのデータオブジェクトを使用するすべての親オブジェクトを特定します。マッピング Mapping B はこのデータオブジェクトを使用します。そのため、ウィザードはこのマッピングに影響を受けるオブジェクトとしてマークします。

マッピング Mapping B はランタイムアプリケーション内で更新され、ワークフロー Workflow B はマッピング Mapping B を使用するため、ワークフロー Workflow B も影響を受けるオブジェクトです。

次の図に、設計時と実行時の状況、選択したマッピング、パッチが継承する依存関係、影響を受けるオブジェクトを示します。



注: オブジェクト Mapping B と Workflow B のランタイムインスタンスは、それぞれの設計時インスタンスには置き換えられません。設計時アプリケーション内のマッピング Mapping B を編集した場合、Mapping B のランタイムインスタンスは設計時インスタンスとは異なります。

オブジェクトの影響の範囲の特定

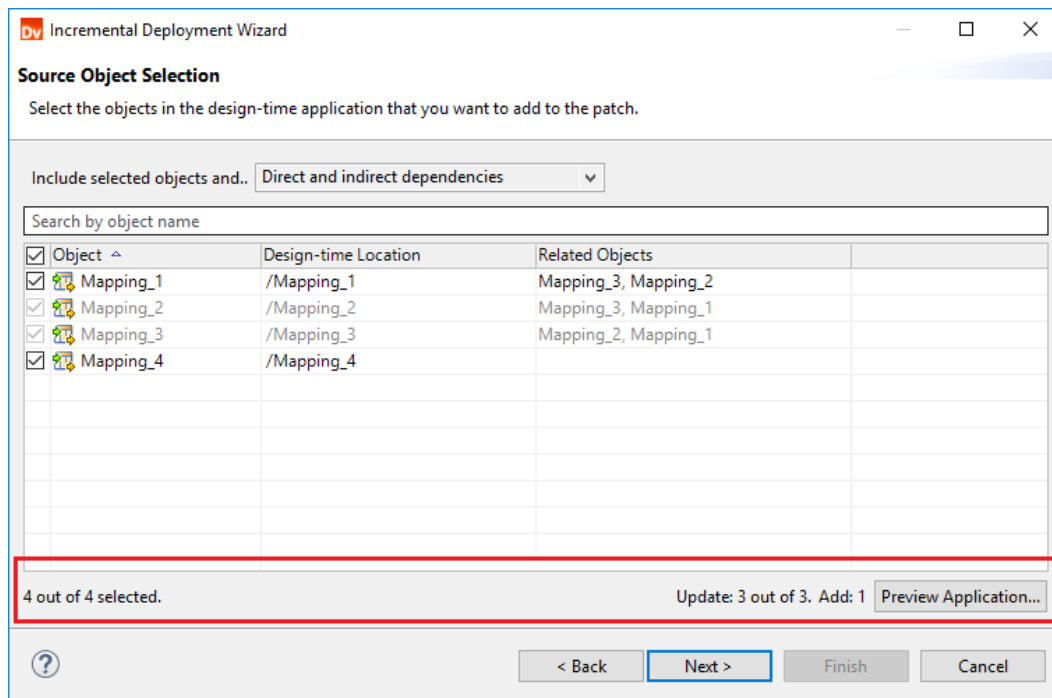
デプロイされたアプリケーションにアプリケーションパッチが与える影響の範囲を特定し、パッチのデプロイメントの間にダウンタイムが発生するオブジェクトの数を把握できます。アプリケーションに複数のユーザーがいる場合、範囲が大きくなると、ユーザーが使用できるオブジェクトの数が少なくなります。

影響を特定するには、次のオブジェクトタイプに基づいて、影響のサマリにある値を使用します。

- 選択したオブジェクト。パッチ内に登場する選択したオブジェクトとパッチが継承するオブジェクトの合計数。最大値は、設計時アプリケーション内のアプリケーションオブジェクトの数です。
- 更新されるオブジェクト。変更される、または影響を受けるオブジェクトの合計数。最大値は、ランタイムアプリケーション内のアプリケーションオブジェクトの数です。
- 追加されるオブジェクト。実行にアプリケーションに追加されるオブジェクトの数。

注: 設計時アプリケーションにランタイムアプリケーションとは異なる依存関係セットが含まれている場合、選択したオブジェクトの数が、更新されるオブジェクトの数と異なる場合があります。

次の図に、差分デプロイメントウィザードがサマリを表示する場所を示します。



パッチデプロイメント戦略の指定

組織のポリシーに応じて、パッチデプロイメント戦略を指定し、アプリケーションパッチのコンテンツをエンドユーザーに提供する方法を決定します。

アプリケーションパッチは次の方法でデプロイできます。

- データ統合サービスにデプロイする。パッチをデータ統合サービスにデプロイすると、データ統合サービスがパッチのコンテンツに基づいて、対応する差分アプリケーションを自動的に更新します。影響を受けないオブジェクトは実行を続け、影響を受けるオブジェクトはパッチのコンテンツに応じて更新されます。

- アーカイブファイルにデプロイする。パッチをアーカイブファイルにデプロイすると、パッチをローカルマシンにファイルとして保存して、そのファイルをバージョン管理システムに格納できます。そのパッチは後でデータ統合サービスにデプロイできます。

パッチをデプロイしてもアプリケーションの状態は変更されません。例えば、停止中のアプリケーションが実行を始めたりはしません。

デプロイメントをキャンセルした場合、データ統合サービスがパッチのコンテンツをデプロイされたアプリケーションに部分的に適用する可能性があります。デプロイメントをキャンセルしても、差分アプリケーションはロールバックされません。

パッチのデプロイメント戦略のプロパティ

[パッチのデプロイメント戦略] ページで、全般プロパティおよびデプロイメントオプションを設定できます。ランタイムアプリケーションのパッチ履歴も確認できます。

次の表に、設定可能な全般プロパティを示します。

全般プロパティ	説明
パッチ名	パッチの名前。
パッチの説明	オプション。パッチの説明。パッチの説明は 270 文字以下にする必要があります。
状態情報の保持	<p>オプション。状態情報が保持されるか破棄されるかを示します。状態情報は、マッピングプロパティと、マッピング出力やシーケンスジェネレータトランスフォーメーションなどのランタイムオブジェクトのプロパティを参照しています。</p> <p>デフォルトでは、状態情報は保持されます。</p> <p>注: アプリケーションの状態を保持または破棄しても、シーケンスジェネレータトランスフォーメーションの「リセット」プロパティには影響しません。</p> <p>状態情報の詳細については、「アプリケーションの状態情報」(ページ 157)を参照してください。</p>

以下の表に、パッチデプロイメントの選択オプションを示します。

デプロイメントオプション	説明
データ統合サービスにデプロイ	パッチをデータ統合サービスにデプロイするには、このオプションを選択します。
ファイルにデプロイ	パッチをアプリケーションパッチアーカイブファイルにデプロイするには、このオプションを選択します。

以下の表に、パッチ履歴内の各パッチに関して確認できるプロパティを示します。

パッチプロパティ	説明
名前	デプロイされたパッチの名前。
説明	デプロイされたパッチの説明。パッチの作成時間がパッチの説明の冒頭に付加されます。

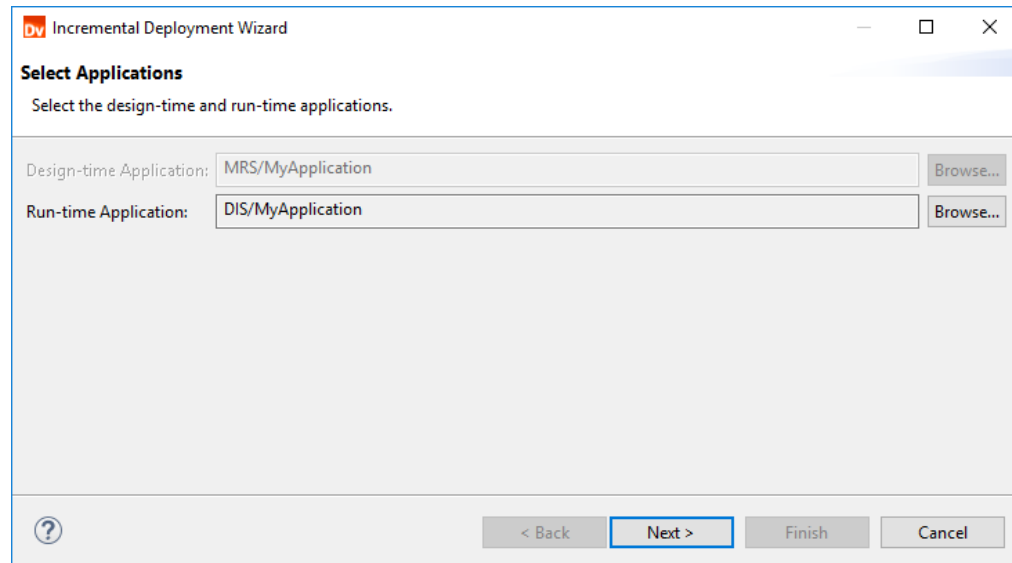
注: デフォルトで、パッチ履歴には、パッチの一覧が作成時間別に表示されます。

アプリケーションパッチのデプロイ

差分アプリケーション内のオブジェクトを編集する場合、アプリケーションパッチを作成してデプロイすることで、デプロイしたアプリケーションのオブジェクトを更新できます。アプリケーションパッチをデータ統合サービスまたはアプリケーションパッチアーカイブファイルにデプロイします。

1. **【オブジェクトエクスプローラ】** ビューで、差分アプリケーションを右クリックし、**【アプリケーションオブジェクトの更新】** を選択します。

差分デプロイメントウィザードの**【アプリケーションの選択】** ページが表示されます。



The image shows a screenshot of the 'Incremental Deployment Wizard' window, specifically the 'Select Applications' page. The window has a title bar with the text 'Incremental Deployment Wizard' and standard Windows window controls (minimize, maximize, close). Below the title bar, the page is titled 'Select Applications' with a subtitle 'Select the design-time and run-time applications.' There are two input fields: 'Design-time Application:' with the text 'MRS/MyApplication' and a 'Browse...' button to its right; and 'Run-time Application:' with the text 'DIS/MyApplication' and a 'Browse...' button to its right. At the bottom of the window, there is a navigation bar with a help icon (question mark in a circle) on the left, and four buttons: '< Back', 'Next >' (which is highlighted with a blue border), 'Finish', and 'Cancel'.

2. **【アプリケーションの選択】** ページで、**【参照】** をクリックし、データ統合サービス上の更新するランタイムアプリケーションを選択します。
3. **【次へ】** をクリックします。

ウィザードがランタイムアプリケーションを取得します。その後、**【ソースオブジェクトの選択】** ページが表示されます。

Incremental Deployment Wizard

Source Object Selection

Select the objects in the design-time application that you want to add to the patch.

Include selected objects and.. **Direct and indirect dependencies**

Search by object name

<input type="checkbox"/> Object ^	Design-time Location	Related Objects
<input type="checkbox"/> MappingA	/MappingA	MappingB
<input type="checkbox"/> MappingB	/MappingB	MappingA

0 out of 2 selected. Update: 0 out of 2. Add: 0 **Preview Application...**

< Back **Next >** **Finish** **Cancel**

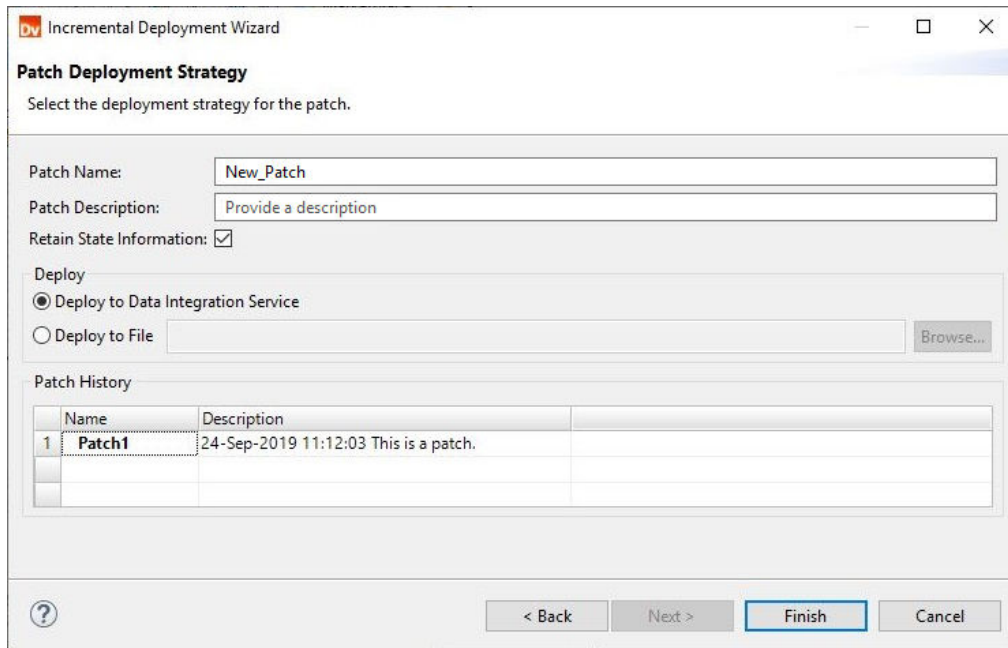
4. **【ソースオブジェクトの選択】** ページで、データ統合サービスでランタイムアプリケーションで置き換えたい設計時アプリケーション内のオブジェクトを選択します。
5. 次のいずれかのアプリケーションパッチタイプを選択します。
 - 直接依存関係。パッチは選択したオブジェクトの直接依存関係を継承します。
 - 直接および間接の依存関係。パッチは選択したオブジェクトの直接依存関係と間接依存関係を継承します。
 - 直接、間接、およびリモートの依存関係。パッチは選択したオブジェクトの直接依存関係、間接依存関係、およびリモート依存関係を継承します。

デフォルトは**【直接および間接の依存関係】**です。

依存関係が**【関連オブジェクト】**カラムに表示されます。例えば、直接依存関係と間接依存関係を継承するアプリケーションパッチの場合は、このカラムには選択したオブジェクトの直接依存関係と間接依存関係がリスト表示されます。

6. ランタイムアプリケーション内の各オブジェクトの影響を確認するには、**【アプリケーションのプレビュー】**を選択します。
7. **【次へ】**をクリックします。

[パッチのデプロイメント戦略] ページが表示されます。



The dialog box is titled "Incremental Deployment Wizard" and "Patch Deployment Strategy". It contains the following fields and controls:

- Patch Name:** A text box containing "New_Patch".
- Patch Description:** A text box containing "Provide a description".
- Retain State Information:** A checkbox that is checked.
- Deploy:** A group box containing two radio buttons:
 - ☒ Deploy to Data Integration Service
 - ☐ Deploy to File
- Patch History:** A table with the following data:

	Name	Description
1	Patch1	24-Sep-2019 11:12:03 This is a patch.
- Buttons:** "< Back", "Next >", "Finish" (highlighted), and "Cancel".

8. パッチの名前を入力します。
9. 必要に応じて、パッチの説明を入力します。
10. 必要に応じて、状態情報を保持することを選択します。
状態情報を保持しないことを選択すると、状態情報は破棄されます。
11. 次のいずれかのデプロイメントオプションを選択します。
 - データ統合サービスにデプロイアプリケーションパッチをデータ統合サービスにデプロイします。データ統合サービスはアプリケーションパッチをランタイムアプリケーションに適用します。
 - ファイルにデプロイアプリケーションパッチをアーカイブファイルにデプロイします。パッチをアーカイブファイルにデプロイする場合、ファイルを保存するローカルマシンのディレクトリを選択します。
12. パッチ履歴を確認します。
パッチ履歴には、差分アプリケーションを更新するためにデプロイされたアプリケーションパッチのリストが表示されます。

アプリケーションパッチのデプロイメントに関するルールとガイドライン

アプリケーションパッチをデプロイして差分アプリケーションを更新する場合は次のツールとガイドラインを参考にしてください。

- [オブジェクトエクスプローラ] ビューからアプリケーションオブジェクトを更新するように選択した場合は、差分デプロイメントウィザードがアプリケーションの依存関係の構築を開始します。[進捗状況] ビューで進捗状況を確認できますが、操作をキャンセルすることはできません。
- パッチを作成する際に、差分デプロイメントウィザードではアプリケーション内のオブジェクトではないオブジェクトは一切リストに表示されません。

例えば、アプリケーションにマッピングが含まれていて、そのマッピングがデータの変換に使用するデータオブジェクトは除外されている可能性があります。この場合、データオブジェクトはアプリケーションオブジェクトではありません。パッチの作成時は、データオブジェクトを確認することはできませんが、ウィザードで、アプリケーション内オブジェクト間の依存関係が認識されればデータオブジェクトか処理されます。

- 設計時アプリケーション内のアプリケーションオブジェクトの名前またはプロジェクトを変更し、そのオブジェクトをアプリケーションパッチを使用して更新した場合、ランタイムアプリケーション内でそのオブジェクトが別のパスに複製されます。
- ユーザーがパッチを初めてデプロイしたときにランタイムアプリケーションのアプリケーションの説明が空である場合、差分デプロイメントウィザードがアプリケーションのパッチ履歴を使用して、説明を更新します。追加のパッチをデプロイすると、ウィザードはその説明にパッチ名を追加します。ユーザーがいずれかのクライアントを使用してアプリケーションの説明を編集しない限り、ウィザードはパッチ名を追加し続けます。

第 11 章

アプリケーションパッチのデプロイメント例

この章では、以下の項目について説明します。

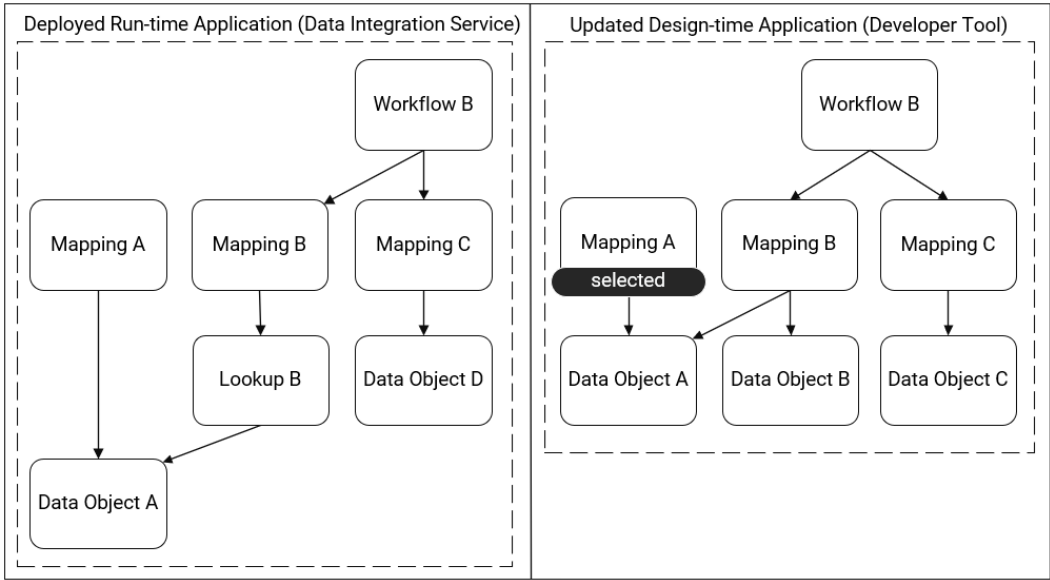
- [アプリケーションパッチのデプロイメント例の概要, 191](#) ページ
- [パッチタイプ: 直接, 192](#) ページ
- [パッチタイプ: 直接および間接, 194](#) ページ
- [パッチタイプ: 直接、間接、リモート, 196](#) ページ
- [まとめ, 198](#) ページ

アプリケーションパッチのデプロイメント例の概要

あなたは DevOps エンジニアです。設計時オブジェクトの変更点を監視して、データ統合サービスにデプロイされたアプリケーション内の対応するランタイムオブジェクトに変更点をマージしています。

最近、ある開発者が差分アプリケーションでマッピング Mapping A を編集しました。そのため、ランタイムアプリケーション内の同じマッピングを更新するパッチを作成する必要があります。

次の図は、データ統合サービス上のランタイムアプリケーションと、Developer tool での設計時アプリケーションを示しており、後者ではマッピング Mapping A がオブジェクトとして選択されています。



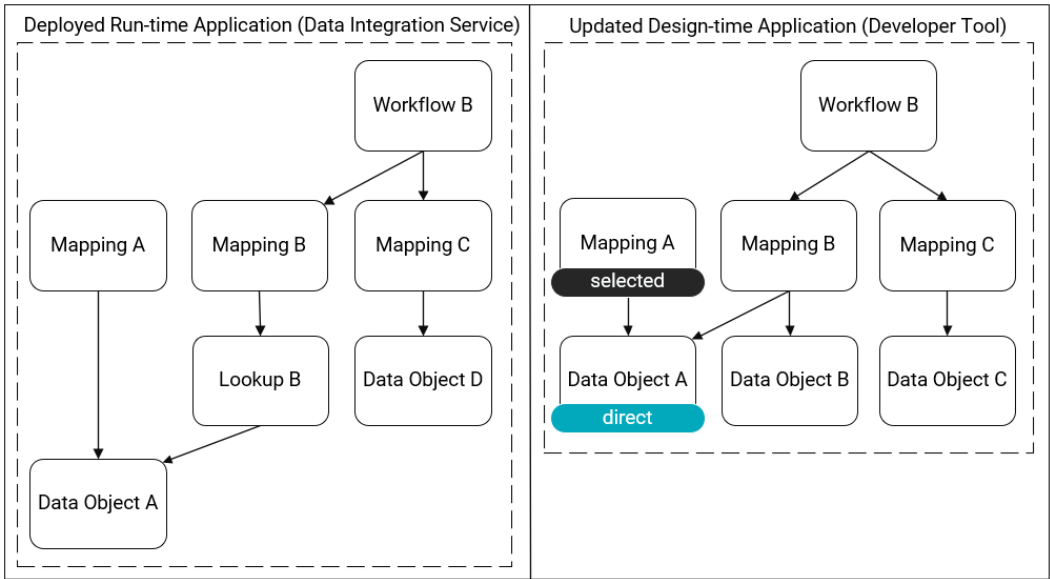
アプリケーションパッチを作成するには、パッチで継承する依存関係を選択する必要があります。継承されるオブジェクトを特定し、オブジェクトの影響を評価して、デPLOYされたアプリケーションへの変更範囲を評価します。

パッチタイプ: 直接

このセクションでは、直接の依存関係のみを継承するアプリケーションパッチについて説明します。

継承されたオブジェクトの特定

次の図は、直接の依存関係のみを継承したパッチに含まれるオブジェクトを示しています。

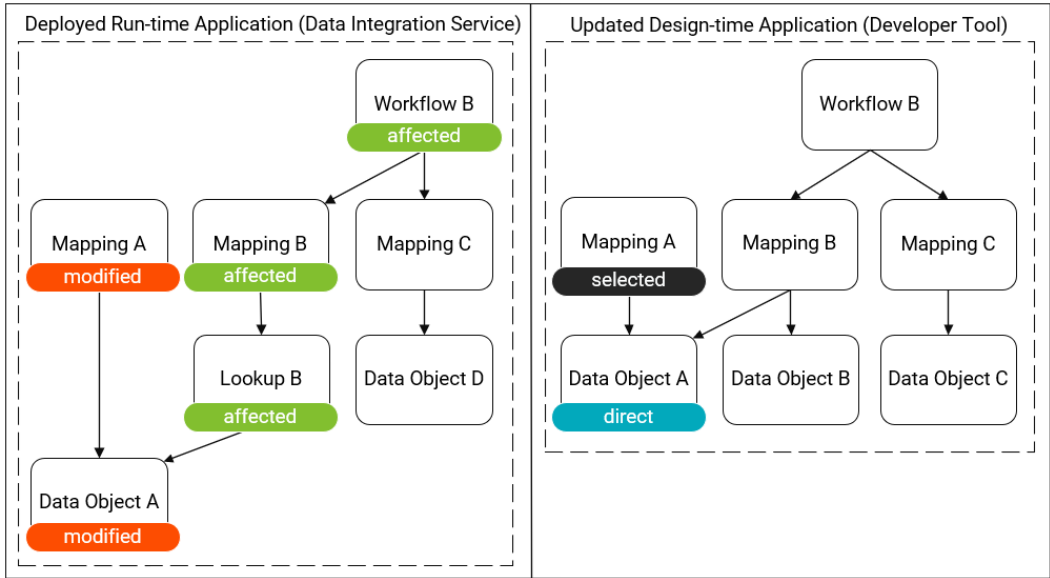


次の表に、マッピング Mapping A がオブジェクトとして選択された場合にパッチが継承する依存関係を示します。

依存関係のタイプ	依存関係
Direct	- Data Object A

オブジェクトへの影響の評価

次の図は、パッチをデプロイした後のランタイムオブジェクトへの影響を示しています。



変更範囲の評価

次の表に、パッチのデプロイメント時に影響を受けるオブジェクトの数を示します。

スコープ項目	オブジェクト数
選択	2
更新	変更: 2 影響: 3* 合計: 5*
追加	0

* 再利用可能なトランスフォーメーション *Lookup B* は、ランタイムでは影響を受けたオブジェクトとして動作しますが、差分デプロイメントウィザードでは影響を受けたオブジェクトとして表示されません。差分デプロイメントウィザードでは、影響を受けたオブジェクトの数は 2 です。

結論

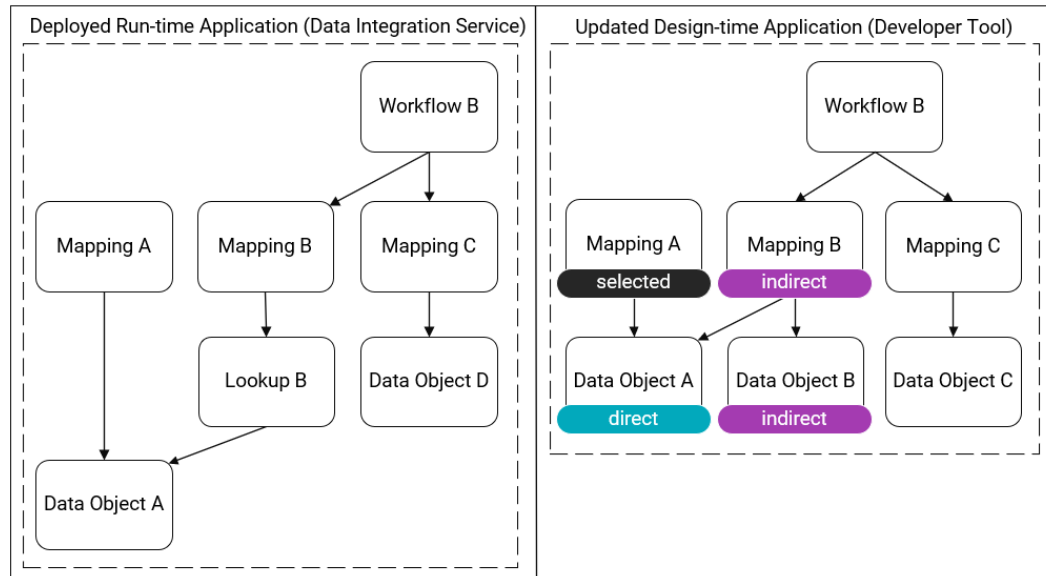
直接の依存関係のみを継承するパッチのランタイムアプリケーションへの影響は小さいですが、ランタイムアプリケーションには影響を受けたオブジェクトが含まれています。影響を受けたオブジェクトは変更されたオブジェクトと違って予測不可能な場合があるため、パッチの適用後にアプリケーションがデータをどのように変換するかについての透過性は低下します。

パッチタイプ: 直接および間接

このセクションでは、直接および間接の依存関係を継承するアプリケーションパッチについて説明します。

継承されたオブジェクトの特定

次の図は、直接および間接の依存関係を継承したパッチに含まれるオブジェクトを示しています。

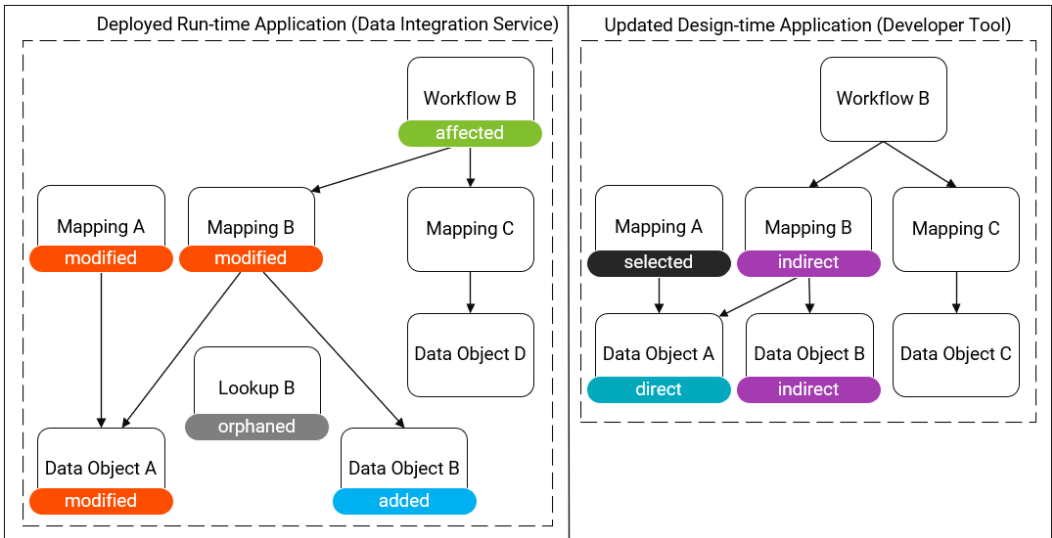


次の表に、マッピング Mapping A がオブジェクトとして選択された場合にパッチが継承する依存関係を示します。

依存関係のタイプ	依存関係
直接	- Data Object A
間接	- Mapping B - Data Object B

オブジェクトへの影響の評価

次の図は、パッチをデプロイした後のランタイムオブジェクトへの影響を示しています。



パッチではマッピング Mapping B が間接依存関係として継承されるため、このマッピングは設計時依存関係を使用し、以前のランタイム依存関係を破棄します。また、ランタイムアプリケーションでは再利用可能なトランスフォーメーション Lookup B を使用しません。代わりに、データオブジェクト Data Object A と Data Object B を使用します。トランスフォーメーション Lookup B は、ランタイムアプリケーションでは孤立した状態となります。

変更範囲の評価

次の表に、パッチのデプロイメント時に影響を受けるオブジェクトの数を示します。

スコープ項目	オブジェクト数
選択	4
更新	変更: 3 影響: 1 合計: 4
追加	1

結論

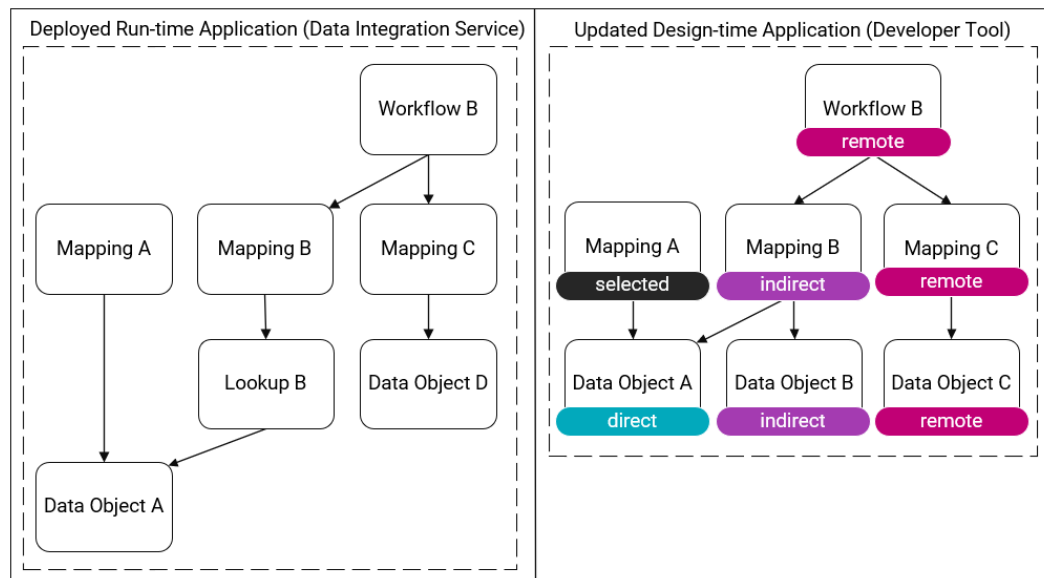
直接および間接の依存関係を継承する場合、直接の依存関係のみを継承する場合と同様の影響がありますが、影響を受けるオブジェクトは少なくなります。マッピング Mapping B は影響を受けるのではなく、変更されるため、より予測可能な形で更新されます。

パッチタイプ: 直接、間接、リモート

このセクションでは、直接、間接、およびリモートの依存関係を継承するアプリケーションパッチについて説明します。

継承されたオブジェクトの特定

次の図は直接、間接、およびリモートの依存関係を継承したパッチに含まれるオブジェクトを示しています。

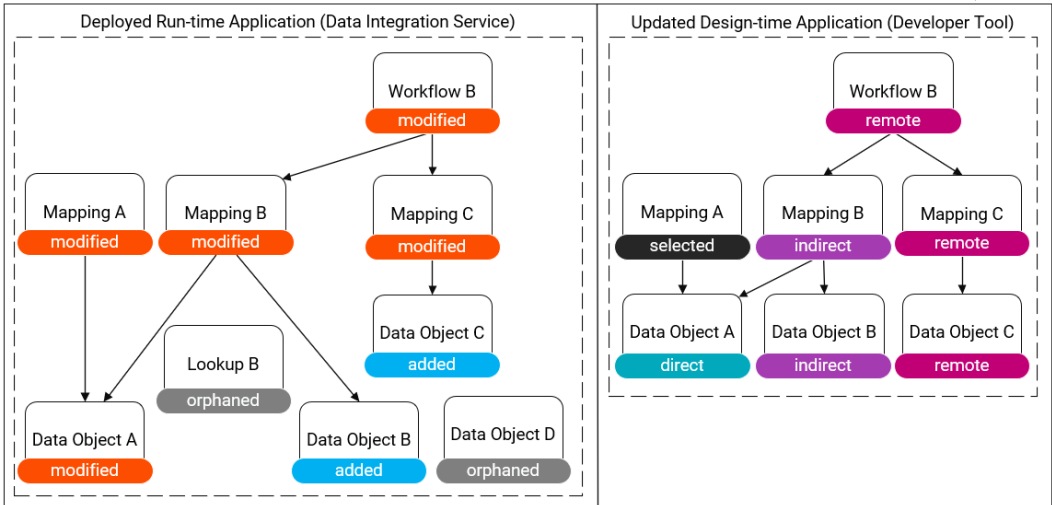


次の表に、マッピング Mapping A がオブジェクトとして選択された場合にパッチが継承する依存関係を示します。

依存関係のタイプ	依存関係
Direct	- Data Object A
Indirect	- Mapping B - Data Object B
Remote	- Workflow B - Mapping C - Data Object C

オブジェクトへの影響の評価

次の図は、パッチをデプロイした後のランタイムオブジェクトへの影響を示しています。



パッチではマッピング Mapping B と Mapping C が継承されるため、これらのマッピングはランタイムアプリケーションで設計時の依存関係を使用します。再利用可能なトランスフォーメーション Lookup B とデータオブジェクト Data Object B は、実行時アプリケーションでは孤立状態となります。

変更範囲の評価

次の表に、パッチのデプロイメント時に影響を受けるオブジェクトの数を示します。

スコープ項目	オブジェクト数
選択	7
更新	変更: 5 影響: 0 合計: 5
追加	2

結論

直接、間接、およびリモートの依存関係を継承するパッチを作成する場合、ランタイムアプリケーションへの変更は透過的です。ランタイムアプリケーションで更新および追加されたオブジェクトは設計時アプリケーションで選択されたオブジェクトと一致するため、ランタイムアプリケーションが設計時アプリケーションのオブジェクトと同じ方法でデータを変換すると想定できます。

ただし代わりにすべてのオブジェクトでダウンタイムが発生するため、アプリケーションを再デプロイしたほうが早いこともあります。アプリケーションを再デプロイした場合、同じオブジェクトが更新され、孤立したオブジェクト（再利用可能なトランスフォーメーション Lookup B とデータオブジェクト Data Object D）はランタイムアプリケーションから削除されます。

まとめ

この例では、直接および間接の依存関係を継承するパッチを作成することでランタイムアプリケーションに対する影響が最小限に抑えられ、オブジェクトのアップタイムが最大化されます。また、差分アプリケーションの機能を最大限に活用できます。

開発環境または QA 環境にアプリケーションをデプロイする場合も、直接および間接の依存関係を継承するパッチを作成するという選択が考えられます。例えば、エンタープライズデータの処理を開始する前に、サンプルデータを使用して選択したマッピング Mapping A が正常に実行するかどうかをテストする場合などです。

ユーザー受け入れテスト（UAT）環境または本番環境にアプリケーションがデプロイされていて、アプリケーションを停止する余裕がある場合は、アプリケーションを再デプロイするという選択肢もあります。アプリケーションを再デプロイした場合、間違いなくランタイムアプリケーションに変更が反映されるため、データは確実に要件に従って変換されます。

第 12 章

継続的統合と継続的配信 (CI/CD)

この章では、以下の項目について説明します。

- [CI/CD の概要, 199 ページ](#)
- [REST API および infacmd クイックリファレンス, 200 ページ](#)
- [継続的統合, 203 ページ](#)
- [継続的配信, 205 ページ](#)
- [自動化された統合プロセスのサンプル, 205 ページ](#)
- [自動化された配信プロセスのサンプル, 208 ページ](#)

CI/CD の概要

CI/CD、つまり継続的統合（continuous integration）と継続的配信（continuous delivery）は、CI/CD パイプラインでの統合と配信の操作を自動化することです。統合と配信の操作はそれぞれ、データ統合サービスの REST API または infacmd コマンドラインプログラムを使用して自動化できます。

CI/CD パイプラインには、オブジェクト設計の際に開発者が使用する統合操作と、オブジェクトをプロダクション環境に配信する配信操作が含まれます。REST API または infacmd を使用すると、以下の方法で統合と配信を自動化できます。

- 開発者がオブジェクトに対して行うすべての変更をデプロイしてテストします。開発者は、オブジェクトがテストに合格したかどうかと、オブジェクトに必要な変更の種類に関するフィードバックを即座に受け取ります。
- テストに合格したオブジェクトをプロダクション環境に配信します。組織の要件に基づいて、プロダクション環境にオブジェクトを配信する前に、QA や UAT などの追加要件にオブジェクトを配信することができます。

継続的統合と継続的配信は完全にログに記録され、チーム全体が確認できるため、チームのメンバは手動による作業を避けて時間を配分できます。

REST API および infacmd クイックリファレンス

継続的統合と継続的配信の設定に使用できる一般的な統合および配信の操作を参照してください。

REST API 要求の詳細については、次のいずれかの場所を参照してください。

- データ統合サービスプロセスのプロパティまたは REST Operations Hub サービスプロセスのプロパティ経由でアクセスする REST API のドキュメント。
- 『*Informatica アプリケーションサービスガイド*』。

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

オブジェクトに関する情報の取得

次の表に、オブジェクトに関する情報を取得するために自動化できる操作を示します。

操作	REST API	infacmd
設計時オブジェクトのクエリ	設計時オブジェクトの取得	dis queryDesignTimeObjects
ランタイムオブジェクトのクエリ	ランタイムオブジェクトの取得	dis queryRunTimeObjects
アプリケーションの一覧表示	ランタイムオブジェクトの取得	dis listApplications
アプリケーションオブジェクトの一覧表示	ランタイムオブジェクトの取得	dis listApplicationObjects
パラメータセットの一覧表示	ランタイムオブジェクトの取得	dis listParameterSets
アプリケーションパッチの一覧表示	該当なし	dis listPatchNames
マッピングの比較	クエリされた 2 つのマッピングの比較	dis compareMapping

オブジェクトのタグ付け

次の表に、オブジェクトにタグ付けするために自動化できる操作を示します。

操作	REST API	infacmd
オブジェクトのタグ付け	設計時オブジェクトへのタグの割り当て	dis tag
オブジェクトタグの置き換え	リポジトリオブジェクトタグの置き換え	dis replaceAllTag
オブジェクトのタグ解除	設計時オブジェクトからタグを削除	dis untag

競合の検出と解決

次の表に、競合を検出および解決するために自動化できる操作を示します。

操作	REST API	infacmd
オブジェクトのロック解除	適用できません	mrs unlockObject
ロックされたオブジェクトの一覧表示	適用できません	mrs listLockedObjects

バージョン管理操作の実行

次の表に、バージョン管理操作を実行するために自動化できる操作を示します。

操作	REST API	infacmd
オブジェクトのチェックイン	バージョン管理操作の実行	mrs checkInObject
オブジェクトのチェックアウト	バージョン管理操作の実行	適用できません
チェックアウトの取り消し	バージョン管理操作の実行	mrs undoCheckout
チェックアウトされたオブジェクトの再割り当て	バージョン管理操作の実行	mrs reassignCheckedOutObject
チェックアウトされたオブジェクトの一覧表示	設計時オブジェクトの取得	mrs listCheckedOutObjects

注: バージョン管理操作は、モデルリポジトリがバージョン管理システムと統合されている場合にのみ実行できます。

オブジェクトのデプロイ

次の表に、オブジェクトをデプロイするために自動化できる操作を示します。

操作	REST API	infacmd
アプリケーションアーカイブファイルへのオブジェクトのデプロイ	該当なし*	tools deployApplication
アプリケーションパッチアーカイブファイルへのオブジェクトのデプロイ	アーカイブファイルへの設計時オブジェクトのデプロイ	dis deployObjectsToFile
データ統合サービスへのアプリケーションアーカイブファイルのデプロイ	アプリケーションのデプロイ	dis deployApplication
データ統合サービスへのアプリケーションパッチアーカイブファイルのデプロイ	アプリケーションの更新	tools patchApplication
データ統合サービスへのアプリケーションアーカイブファイルの再デプロイ	アプリケーションの更新	dis updateApplication
* REST API を使用してアプリケーションアーカイブファイルにオブジェクトをデプロイすることはできません。データ統合サービスに初めてアプリケーションをデプロイする場合、またはアプリケーションを再デプロイする場合は、アプリケーションパッチアーカイブファイルを使用してください。		

デプロイ済みアプリケーションの管理

次の表に、デプロイ済みアプリケーションを管理するために自動化できる操作を示します。

操作	REST API	infacmd
アプリケーションの起動	アプリケーションの起動または停止	dis startApplication
アプリケーションの停止	アプリケーションの起動または停止	dis stopApplication
アプリケーションのバックアップ	適用できません	dis backupApplication
アプリケーションの復元	適用できません	dis restoreApplication
アプリケーションの名前変更	適用できません	dis renameApplication
アプリケーションのデプロイ解除	アプリケーションのデプロイ解除	dis undeployApplication
デプロイ済みマッピングの実行	デプロイ済みマッピングの実行	ms runMapping

REST API または infacmd の使用

データ統合サービスの REST API または infacmd コマンドラインプログラムのどちらを使用して統合と配信の操作を自動化するかを決定するには、組織の要件を査定します。

REST API と infacmd には以下の違いがあります。

- infacmd を使用したほうが、REST API よりも多くのユースケースを実現できます。例えば、infacmd コマンドを使用してアプリケーションをバックアップすることができます。
- infacmd を使用するには、Informatica サービスまたは infacmd ユーティリティをインストールする必要があります。これに対して、REST API にはセットアップ要件がないため、クライアントマシンに Informatica サービスがインストールされていない環境でも REST API を実行できます。

データ統合サービスの REST API の詳細については、『*Informatica アプリケーションサービスガイド*』を参照してください。

infacmd コマンドラインプログラムの詳細については、『*Informatica コマンドリファレンス*』を参照してください。

REST API を使用したアプリケーションのデプロイと再デプロイ

データ統合サービス REST API は、アプリケーションパッチアーカイブファイルを使用してオブジェクトをデプロイします。データ統合サービスに初めてオブジェクトをデプロイする場合、およびアプリケーションを再デプロイする場合は、アプリケーションアーカイブファイルではなくアプリケーションパッチアーカイブファイルを使用します。

次のタスクを実行するには、アプリケーションパッチアーカイブファイルを使用します。

アプリケーションを初めてデプロイする

オブジェクトをアプリケーションとしてデータ統合サービスに初めてデプロイするには、次のタスクを実行します。

1. REST API Deploy design-time objects to an archive file を使用して、アプリケーションパッチアーカイブファイルを作成します。

2. REST API Deploy an application を使用して、パッチをデータ統合サービスにデプロイします。
データ統合サービスがパッチ内のオブジェクトを使用して差分アプリケーションを作成します。

アプリケーションを再デプロイする

アプリケーションを再デプロイするには、REST API Update an application を使用して、次のコンポーネントを設定します。

1. アプリケーションパッチアーカイブファイルを指定します。
2. 要求ペイロードパラメータ "redployApplication":true. を設定します。
データ統合サービスが、デプロイされた差分アプリケーション内のオブジェクトをパッチ内のオブジェクトで置き換えます。

継続的統合

継続的統合を使用すると、開発者がオブジェクトに対して行ったすべての変更が開発環境でデプロイされテストされてから、オブジェクトの追加環境への配信が承認されます。

継続的統合を設定するために、以下のタスクを自動化できます。

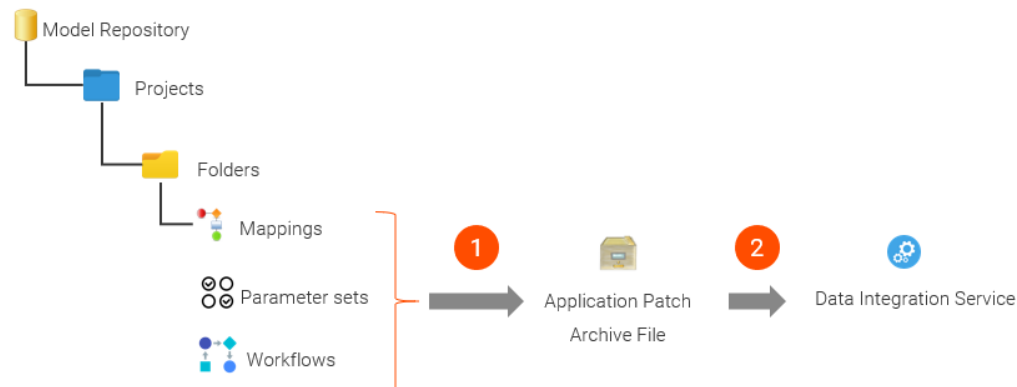
- オブジェクトをデプロイして正常性テストを実行する。
- バージョン管理操作を実行してオブジェクトを開発者に割り当てる。
- オブジェクトのロックによる競合を検出して解決する。

オブジェクトのデプロイ

開発者がオブジェクトに対して行ったすべての変更がデプロイされテストされるように継続的統合を設定します。

テスト用にオブジェクトを分離するには、オブジェクトをアプリケーションアーカイブファイルに、さらにデータ統合サービスにデプロイします。オブジェクトがデータ統合サービスにデプロイされると、開発者は、テスト結果に干渉することなく、Developer tool でオブジェクトへのアクセスおよび編集を続けることができます。

以下の図は、オブジェクトのデプロイ方法を示しています。



1. 自動化されたプロセスにより、オブジェクトがアプリケーションパッチアーカイブファイルにデプロイされます。
2. 自動化されたプロセスにより、このアーカイブファイルがデータ統合サービスにデプロイされます。

オブジェクトがデプロイされたら、正常性テストを実行してマッピングが正常に実行されることを確認できます。さらにテストを行うために、サードパーティのツールを使用してマッピング出力が正しいことを確認できます。テスト結果に基づいて、追加環境へのオブジェクトの配信を承認することや、オブジェクトを編集のために開発者に割り当てることができます。

オブジェクトの割り当て

チームベース開発環境で、継続的統合を設定してオブジェクトを開発者に割り当てることができます。

例えば、マッピングの正常性テストが失敗した場合に、最後にマッピングを変更した開発者に編集用のマッピングを割り当てることができます。マッピングを割り当てるには、REST API [バージョン管理操作の実行] またはコマンド `infacmd mrs reassignCheckedOutObject` を使用できます。

オブジェクトを開発者に割り当てたり、オブジェクトをある開発者から別の開発者に再割り当てするには、バージョン管理操作を実行する必要があります。モデルリポジトリが、GIT、SVN、Perforce などのバージョン管理システムと統合されていることを確認してください。

モデルリポジトリをバージョン管理システムと統合する方法の詳細については、『*Informatica アプリケーションサービスガイド*』を参照してください。

オブジェクト競合の解決

オブジェクトに変更が必要な場合に、インテントベースのロックを使用するように継続的統合を設定し、オブジェクト間の競合を検出して解決し、開発者のコラボレーションを促進することができます。

インテントベースのロックを使用すると、他の開発者はオブジェクトに変更を保存できなくなります。ロックにより、同じオブジェクトへのアクセスが必要な開発者間で競合が発生する場合があります。

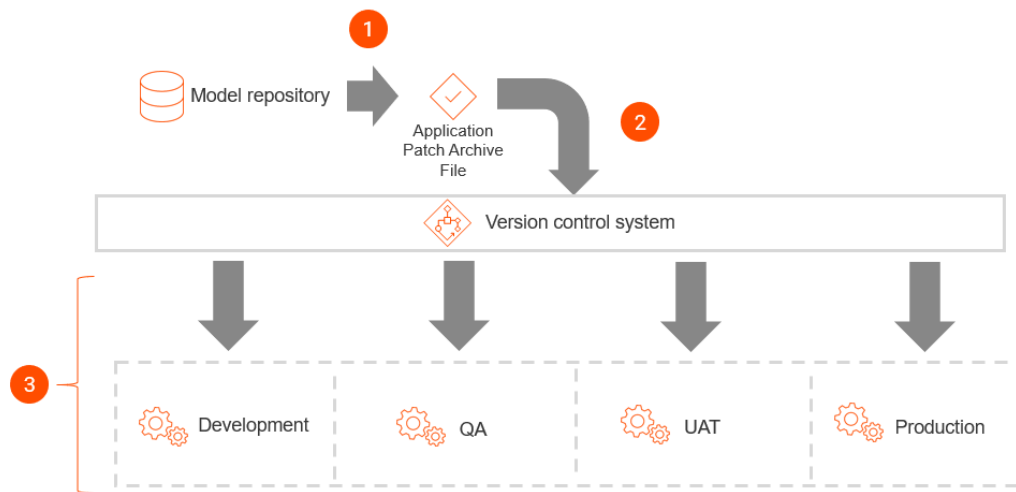
例えば、ある開発者が、あるオブジェクトを休暇前にロックしたとします。このロックにより、そのオブジェクトへのアクセスを必要とする他の開発者に競合が生じます。このような競合を解決するために、サードパーティのツールを使用して従業員の作業スケジュールを読み込み、競合の可能性を検出し、`infacmd` コマンドにロジックを渡してロックされたオブジェクトを一覧表示し、必要に応じてオブジェクトのロックを解除することができます。

継続的配信

オブジェクトが開発環境でのテストに合格すると、そのオブジェクトは継続的にプロダクション環境に配信されます。組織の要件に基づいて、プロダクション環境にオブジェクトを配信する前に、QA や UAT など追加の環境への継続的配信を設定できます。

継続的配信を簡素化するために、アーカイブファイルを使用して、開発環境でテストおよび承認されたオブジェクトと追加の環境に配信されるオブジェクトの間に一貫性を作成することができます。一貫性を作成するには、継続的配信を設定してオブジェクトを 1 つのアーカイブファイルにデプロイし、そのアーカイブファイルをバージョン管理システムに保存して、アーカイブファイルを各環境にデプロイします。

以下の図は、開発環境、QA 環境、UAT 環境およびプロダクション環境へのアプリケーションパッチアーカイブファイルのデプロイ方法を示しています。



1. オブジェクトがアプリケーションパッチアーカイブファイルにデプロイされます。
2. アーカイブファイルはバージョン管理システムに保存されています。
3. 同じアーカイブファイルが、開発環境、QA 環境、UAT 環境およびプロダクション環境にデプロイされます。

自動化された統合プロセスのサンプル

この節では、開発環境で開発者がマッピングに対して行ったすべての変更がデプロイされテストされるように継続的統合を設定する方法を示します。

以下のタスクを行うように継続的統合を設定します。

1. デプロイメント用にマッピングを準備する。
2. マッピングをデータ統合サービスにデプロイする。
3. 正常性テストを実行する。
4. 編集用にマッピングを準備する。

手順 1. デプロイメント用にマッピングを準備

テスト用にマッピングをデプロイする前に、変更されたマッピングを特定し、デプロイメント用にマッピングをマークします。

以下の統合操作を行うように継続的統合を設定します。

1. 変更されたマッピングを特定する。

変更されたマッピングを特定するには、各設計時マッピングを、対応するランタイムマッピング（データ統合サービスにデプロイされたもの）と、継続的に比較します。

マッピングを比較するには、クエリを REST API Compare two queried mappings またはコマンド `infacmd dis compareMapping` に渡します。

2. タグを割り当ててデプロイメント用にマッピングをマークする。

設計時マッピングと対応するランタイムマッピングの間の違いが比較レポートで特定された場合は、デプロイメント用にマッピングをマークするタグを設計時マッピングに割り当てます。

例えば、タグ `DeploymentInProgress` をマッピングに割り当てます。このタグを、REST API Assign tags to design-time objects またはコマンド `infacmd dis tag` に渡すことができます。

手順 2. データ統合サービスへのマッピングのデプロイ

デプロイメント用にマッピングを準備したら、変更されたマッピングをアプリケーションパッチアーカイブファイルに、さらにデータ統合サービスにデプロイします。

以下の統合操作を行うように継続的統合を設定します。

1. クエリを実行してデプロイメント用にマークされたマッピングを特定する。

デプロイメントが必要なマッピングを特定するには、タグ `DeploymentInProgress` が付いた設計時オブジェクトをクエリします。クエリを、REST API [設計時オブジェクトの取得] またはコマンド `infacmd dis queryDesignTimeObjects` に渡します。

2. マッピングをアプリケーションパッチアーカイブファイルにデプロイする。

クエリで少なくとも 1 つのマッピングが返された場合は、そのマッピングをアプリケーションパッチアーカイブファイルにデプロイします。

クエリを、REST API [設計時オブジェクトをアーカイブファイルにデプロイ] またはコマンド `infacmd dis deployObjectsToFile` に渡します。この REST API 要求または `infacmd` コマンドにより、アーカイブファイルがデータ統合サービスマシンに保存されます。

3. アーカイブファイルをデータ統合サービスにデプロイする。

アーカイブファイルを、開発環境のデータ統合サービスにデプロイします。アーカイブファイルの場所を、REST API [アプリケーションの更新] またはコマンド `infacmd tools patchApplication` に渡します。

手順 3. 正常性テストの実行

データ統合サービスでマッピングが更新されたら、正常性テストを実行してマッピングが正常に動作することを確認します。

以下の統合操作を行うように継続的統合を設定します。

1. デプロイ済みのマッピングが正常に動作することを確認する。

デプロイ済みのマッピングが正常に動作することを確認するには、以下の作業を実行します。

1. デプロイ済みのマッピングを実行するには、REST API `Run a deployed mapping` またはコマンド `infacmd ms runMapping` を使用します。
2. 組織の要件に基づいて正常性テストを実行します。例えば、サードパーティツールのスクリプトを使用して統合タスクを自動化する場合は、マッピングターゲット内のデータを読み込むテストケースを作成し、マッピング出力が正しいことを確認できます。

2. 追加の変更が必要なマッピングをマークするタグを割り当てる。

追加の変更が必要なマッピングをマークするには、タグ `RequiresChanges` を、REST API `Assign tags to design-time objects` またはコマンド `infacmd dis tag` に渡します。

変更が必要なマッピングからタグ `DeploymentInProgress` を削除することもできます。タグ `DeploymentInProgress` を、REST API `Remove tags from design-time objects` またはコマンド `infacmd dis untag` に渡します。

3. 正常に実行されたマッピングを承認するタグを割り当てる。

正常に実行されたマッピングを承認するには、タグ `ReadyForDelivery` を、REST API `Replace repository object tags` またはコマンド `infacmd dis replaceAllTag` に渡します。この REST API 要求または `infacmd` コマンドにより、マッピングに割り当てられているすべてのタグが、タグ `ReadyForDelivery` に置き換えられます。

手順 4. 編集用にマッピングを準備

更新されたマッピングがテストに失敗した場合は、マッピングを編集用に準備します。

以下の統合操作を行うように継続的統合を設定します。

1. 編集用にマッピングを適切な開発者に割り当てる。

マッピングを適切な開発者に割り当てるには、以下の作業を実行します。

1. REST API `Get design-time objects` またはコマンド `infacmd dis queryDesignTimeObjects` を使用して、タグ `RequiresChanges` が付いている設計時マッピングをクエリします。
2. 使用するサードパーティツールによっては、クエリ結果を解析して、各マッピングを最後に変更した開発者のユーザー名を取得できます。
3. 編集用にマッピングを開発者に割り当てるには、REST API `Perform version control operations` またはコマンド `infacmd mrs reassignCheckedOutObject` を使用します。

注: バージョン管理操作は、モデルリポジトリがバージョン管理システムと統合されている場合にのみ実行できます。

2. テスト結果について開発者に通知する。

必要に応じて、サードパーティツールを使用して、変更が必要なマッピングについて開発者に通知することができます。

例えば、REST API `Get design-time objects` またはコマンド `infacmd dis queryDesignTimeObjects` を使用して、タグ `RequiresChanges` が付いている設計時マッピングをクエリできます。

次に、サードパーティツールでスクリプトを実行して、以下の作業を実行できます。

1. クエリ結果を読み取って応答を解析し、マッピングを最後に変更した開発者のユーザー名を取得する。
2. そのユーザー名に関連付けられている電子メールアドレスを従業員データベースで検索する。

3. テストに失敗したマッピングに関する情報とマッピングが失敗した理由を伝える電子メールメッセージを生成する。

自動化された配信プロセスのサンプル

この節では、開発環境でマッピングが承認されたときにマッピングをプロダクション環境にデプロイするように継続的配信を設定する方法を示します。

以下の配信操作を行うように継続的配信を設定します。

1. 配信が承認されたマッピングを特定する。

配信が承認されたマッピングを特定するには、以下の作業を実行します。

1. 開発環境内のマッピングにタグ `ReadyForDelivery` が付いているものがあるかどうかを継続的にチェックします。
2. 設計時マッピングをクエリして、タグ `ReadyForDelivery` の有無を確認します。
3. クエリを、REST API `Get design-time objects` またはコマンド `infacmd dis queryDesignTimeObjects` に渡します。

2. マッピングをアプリケーションパッチアーカイブファイルにデプロイする。

クエリで少なくとも 1 つのマッピングが返された場合は、そのマッピングをアプリケーションパッチアーカイブファイルにデプロイします。

クエリを、REST API `Deploy design-time objects to an archive file` またはコマンド `infacmd dis deployObjectsToFile` に渡します。この REST API 要求または `infacmd` コマンドにより、アーカイブファイルがデータ統合サービスマシンに保存されます。

3. アーカイブファイルをデータ統合サービスにデプロイする。

アーカイブファイルを、プロダクション環境のデータ統合サービスにデプロイします。アーカイブファイルの場所を、REST API `Update an application` またはコマンド `infacmd tools patchApplication` に渡します。

第 13 章

オブジェクトのインポートとエクスポート

この章では、以下の項目について説明します。

- [オブジェクトのインポートとエクスポートの概要, 209 ページ](#)
- [オブジェクトのインポートとエクスポート, 210 ページ](#)
- [オブジェクトエクスポート, 211 ページ](#)
- [オブジェクトインポート, 212 ページ](#)

オブジェクトのインポートとエクスポートの概要

プロジェクトの複数のオブジェクトを.xml ファイルにエクスポートできます。オブジェクトをインポートするときに、.xml ファイルで個々のオブジェクトまたはすべてのオブジェクトを選択できます。

オブジェクトを.xml ファイルにエクスポートしたり、その.xml ファイルからオブジェクトをインポートしたりできます。オブジェクトをエクスポートすると、そのオブジェクトのメタデータが格納された.xml ファイルが作成されます。この.xml ファイルを使用して、オブジェクトをプロジェクトまたはフォルダにインポートします。オブジェクトのインポートとエクスポートは、infacmd コマンドを使用して行うこともできます。

オブジェクトのエクスポートとインポートで、次の作業を行えます。

メタデータのプロダクション環境へのデプロイメント

開発環境のリポジトリでマッピングのテストを終えた後で、そのマッピングを.xml ファイルにエクスポートし、そのファイルからプロダクション環境のリポジトリへマッピングをインポートできます。

メタデータのアーカイブ

不要になったオブジェクトをリポジトリから削除する前に、.xml ファイルへエクスポートできます。

メタデータの共有

第三者とメタデータを共有できます。例えば、テストまたは分析のためにマッピングを他の人に送信できます。

リポジトリ間でのメタデータのコピー

同じクライアントからでは接続できないリポジトリの間でオブジェクトをコピーできます。このような場合には、オブジェクトを.xml ファイルにエクスポートし、その.xml ファイルをコピー先のマシンに転送します。そして、.xml ファイルからコピー先のリポジトリにオブジェクトをインポートします。同じバージョンのリポジトリ間でオブジェクトをエクスポートおよびインポートできます。オブジェクトにタグが含まれている場合は、Developer ツールによって自動的にリポジトリにインポートされます。

infacmd を使用して、読み取り可能な.xml ファイルをエクスポートファイルから生成できます。オブジェクトをリポジトリにインポートする前に、読み取り可能な.xml でオブジェクト名を編集したりエクスポート.xml を更新したりすることもできます。

以前の Informatica リリースから.xml ファイルにエクスポートしたオブジェクトを現在のメタデータ形式にアップグレードしてから、アップグレードしたオブジェクトを現在のリリースにインポートできます。オブジェクトをインポートする前に、mrs UpgradeExportedObjects コマンドを使用して、エクスポートしたオブジェクトを現在のメタデータ形式にアップグレードします。

オブジェクトのインポートとエクスポート

プロジェクトやプロジェクト内のオブジェクトをインポートおよびエクスポートできます。リポジトリ内のアプリケーションアーカイブファイルをインポートおよびエクスポートすることもできます。

オブジェクトをエクスポートすると、依存オブジェクトもエクスポートされます。依存オブジェクトは、別のオブジェクトによって使用されるオブジェクトです。例えば、マッピング入力として使用される物理データオブジェクトは、そのマッピングの依存オブジェクトです。オブジェクトをインポートすると、すべての依存オブジェクトがインポートされます。

プロジェクトまたはフォルダー内のオブジェクトをエクスポートまたはインポートするときは、モデルリポジトリサービスによってオブジェクト階層が保持されます。

以下の表に、エクスポートできるオブジェクトと依存オブジェクトを示します。

オブジェクト	依存関係
アプリケーション	- SQL データサービス、マッピング、またはワークフローおよびその依存オブジェクト
プロジェクト	- プロジェクトには他のオブジェクトは含まれますが、依存オブジェクトは含まれません
フォルダー	- フォルダーには他のオブジェクトは含まれますが、依存オブジェクトは含まれません
参照テーブル	- 参照テーブルには依存オブジェクトは含まれません
コンテンツセット	- コンテンツセットには依存オブジェクトは含まれません
物理データオブジェクト (カスタマイズデータオブジェクト以外)	- 物理データオブジェクトには依存オブジェクトは含まれません
カスタマイズデータオブジェクト	- 物理データオブジェクト
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 - 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とその依存オブジェクト - マップレットとその依存オブジェクト

オブジェクト	依存関係
トランスフォーメーション	<ul style="list-style-type: none"> - 物理データオブジェクト - 参照テーブル - コンテンツセット
マップレット	<ul style="list-style-type: none"> - 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とその依存オブジェクト - マップレットとその依存オブジェクト
マッピング	<ul style="list-style-type: none"> - 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とその依存オブジェクト - マップレットとその依存オブジェクト
SQL データサービス	<ul style="list-style-type: none"> - 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とその依存オブジェクト - マップレットとその依存オブジェクト
プロファイル	<ul style="list-style-type: none"> - 論理データオブジェクト - 物理データオブジェクト
スコアカード	<ul style="list-style-type: none"> - プロファイルとその依存オブジェクト
Web サービス	<ul style="list-style-type: none"> - 操作マッピング
ワークフロー	<ul style="list-style-type: none"> - マッピングとその依存オブジェクト

オブジェクトエクスポート

オブジェクトをエクスポートすると、そのオブジェクトのメタデータが格納された XML ファイルが作成されます。

エクスポートするオブジェクトを選択できます。依存オブジェクトもすべてエクスポートするように選択する必要があります。Developer ツールによってオブジェクトと依存オブジェクトがエクスポートされます。エクスポートされるのは最後に保存されたバージョンのオブジェクトです。Developer ツールは、XML ファイルの要素に Cyclic Redundancy Checking Value (CRCVALUE) コードを付加します。CRCVALUE コードが含まれる要素の属性を変更すると、オブジェクトをインポートできなくなります。属性を変更する場合は、`infacmd xrf` コマンドを使用します。

オブジェクトのエクスポートは、`infacmd tools ExportObjects` コマンドを使用して行うこともできます。

オブジェクトのエクスポート

別のリポジトリのモデルリポジトリオブジェクトを使用するために、オブジェクトを XML メタデータファイルとしてエクスポートできます。

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】 ウィザードが開きます。
2. **【Informatica】** > **【オブジェクトメタデータファイルのエクスポート】** を選択します。
【次へ】 をクリックします。
3. **【参照】** をクリックします。エクスポートするオブジェクトを含むリポジトリプロジェクトを選択します。
【次へ】 をクリックします。
4. エクスポートするオブジェクトを 1 つ以上選択してください。エクスポートプロセスを開始する前にリポジトリオブジェクトをハイライトした場合は、ウィザードがオブジェクトを選択します。
5. ファイル名と XML メタデータファイルの場所を入力します。Developer ツールは、選択したオブジェクトをすべて 1 つのファイルにエクスポートします。
【次へ】 をクリックします。
6. ウィザードはメタデータオブジェクトが使用する依存オブジェクトを表示します。
【次へ】 をクリックし、依存オブジェクトを受け入れます。
7. 選択したオブジェクトに参照データオブジェクトが含まれる場合は、**【コンテンツのエクスポート】** オプションを選択し、エクスポートの設定を確認します。
 - エクスポートする参照データファイルの名前と場所を確認します。Developer ツールサービスは参照データファイルを 1 つの ZIP ファイルにエクスポートします。デフォルトの場合、ウィザードは ZIP ファイルと XML メタデータファイルを同じディレクトリにエクスポートします。
 - 参照データが使用するコードページを確認します。デフォルトのコードページは UTF-8 です。参照テーブルデータをエクスポートする場合は、デフォルトのコードページを受け入れます。
 - エクスポートする確率的なモデルデータを確認します。デフォルトの場合、ウィザードはすべてのモデルデータをエクスポートします。選択したオブジェクトが確率的なモデルを含まない場合、エクスポートプロセスはオプションを無視します。
8. **【完了】** をクリックし、選択したオブジェクトをエクスポートします。
Developer ツールはオブジェクトメタデータを XML ファイルにエクスポートし、任意の依存参照データファイルを ZIP ファイルにエクスポートします。

オブジェクトインポート

プロジェクトまたはプロジェクト内のオブジェクトをエクスポートファイルからインポートできます。オブジェクトと依存オブジェクトをプロジェクトまたはフォルダにインポートできます。

プロジェクトまたは個別のオブジェクトをインポートできます。プロジェクト内のすべてのオブジェクトを再利用する場合は、プロジェクトをインポートします。複数のプロジェクトにまたがってオブジェクトを再利用する場合は、個々のオブジェクトをインポートします。

オブジェクトをインポートするときに、すべての依存オブジェクトが一覧表示されます。オブジェクトをインポートする前に、各依存オブジェクトをターゲットに追加する必要があります。

オブジェクトをインポートするときに、エクスポートファイル内のオブジェクトの名前がターゲットプロジェクトまたはフォルダ内のオブジェクトと同じである場合があります。名前の競合を解決する方法を選択できます。

オブジェクトのインポートは、`infacmd tools ImportObjects` コマンドを使用して行うこともできます。

以前のバージョンで作成したエクスポートファイルからオブジェクトをインポートすることはできません。

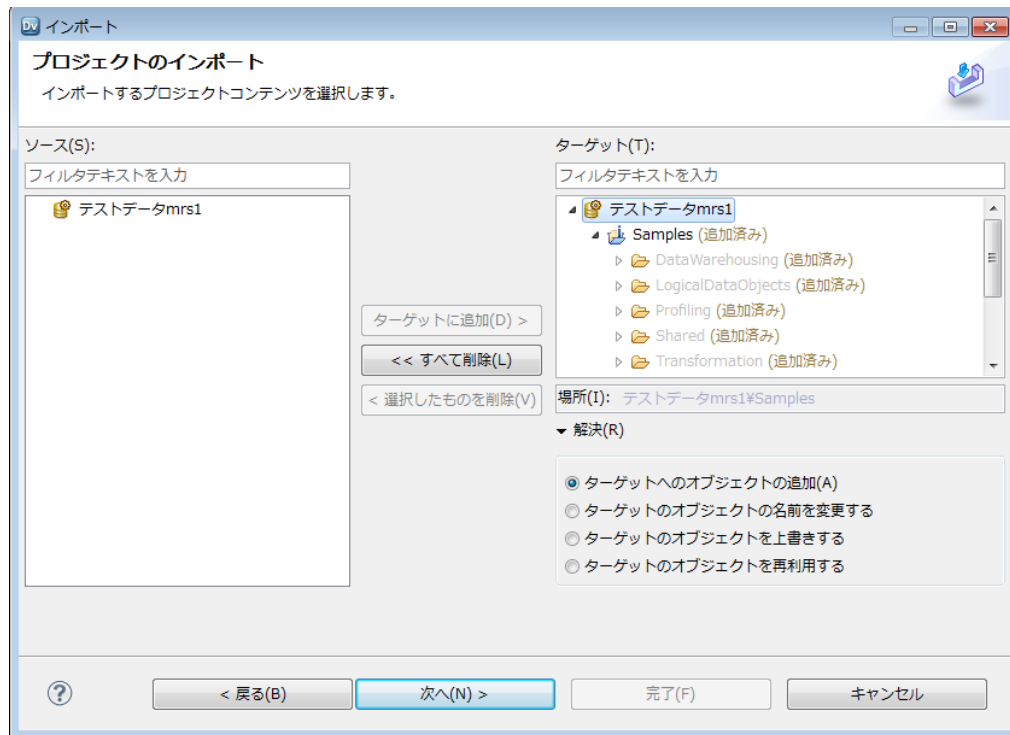
プロジェクトのインポート

XML ファイルからターゲットリポジトリにプロジェクトをインポートできます。また、プロジェクトのコンテンツをターゲットリポジトリ内のプロジェクトにインポートすることもできます。

1. **【ファイル】** > **【インポート】** をクリックします。
2. **【Informatica】** > **【オブジェクトメタデータファイルのインポート（基本）】** を選択します。
3. **【次へ】** をクリックします。
4. **【参照】** をクリックして、インポートするエクスポートファイルを選択します。
5. **【次へ】** をクリックします。
6. **【ソース】** ペインでプロジェクトまたは `[<project name>プロジェクトコンテンツ]` を選択します。
 - **【ソース】** ペインでプロジェクトを選択した場合は、**【ターゲット】** ペインでプロジェクトをインポートするモデルリポジトリサービスを選択します。
 - **【ソース】** ペインでプロジェクトコンテンツを選択した場合は、**【ターゲット】** ペインでプロジェクトコンテンツをインポートするプロジェクトを選択します。
7. **【ターゲットに追加】** をクリックして、プロジェクトをターゲットに追加します。

ヒント: プロジェクトを **【ソース】** ペインから **【ターゲット】** ペインのリポジトリにドラッグすることもできます。また、**【ソース】** ペインのプロジェクトコンテンツを **【ターゲット】** ペインのプロジェクトにドラッグすることもできます。
8. **【解決】** をクリックして、重複するオブジェクトの処理方法を指定します。

インポートするオブジェクトの名前を変更するか、インポートするオブジェクトで既存のオブジェクトを上書きするか、既存のオブジェクトを再利用することができます。デフォルトでは、重複するすべてのオブジェクトの名前が変更されます。



9. [次へ] をクリックします。

インポート中の参照テーブルデータが一覧表示されます。追加の参照テーブル設定を指定します。

10. [次へ] をクリックします。

インポート対象オブジェクトの概要が表示されます。[ソースとターゲットのオブジェクトのリンク] をクリックし、ソースとターゲットのどちらかのオブジェクトを選択すると、[ソース] および [ターゲット] の表示ペインにあるオブジェクトがリンクされます。例えば、このオプションを選択してから [ソース] ペインでオブジェクトを選択すると、[ターゲット] ペインにある同じオブジェクトが選択されます。

11. [追加のインポート設定] ペインで、インポートファイルからターゲットドメインへの接続をマッピングします。オブジェクトの既存のタグを上書きするかどうかを選択することもできます。
12. [完了] をクリックします。

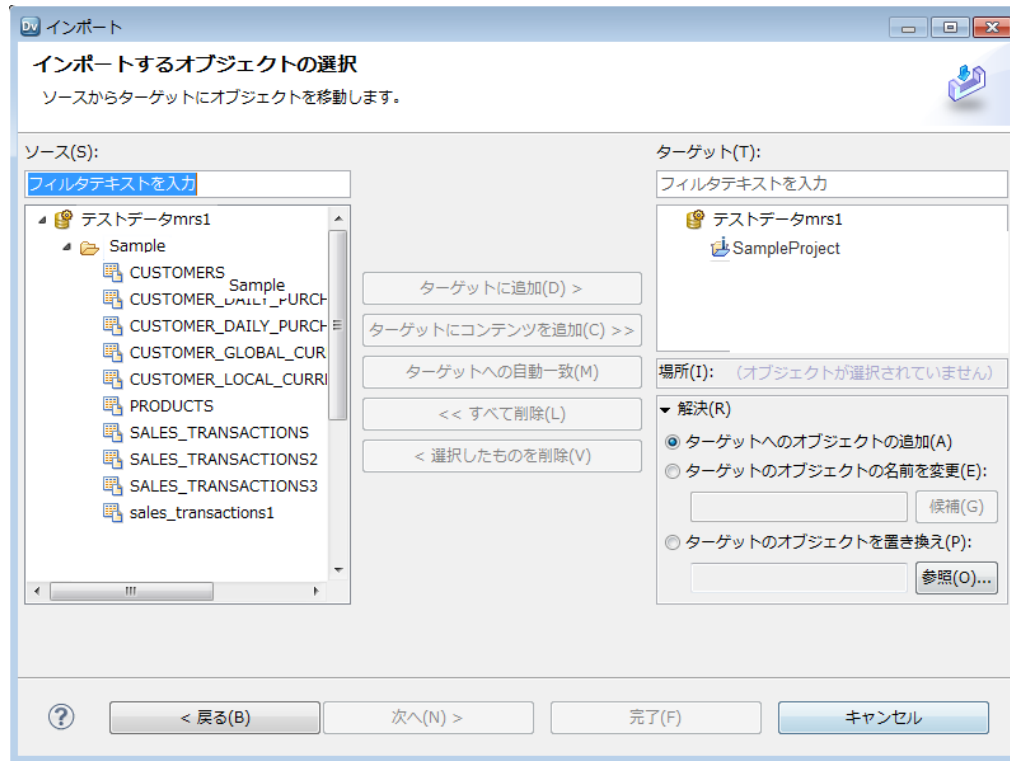
重複するプロジェクトの名前を変更した場合は、モデルリポジトリサービスによってオブジェクト名に番号が付加されます。プロジェクトの名前はインポート後に変更できます。

オブジェクトのインポート

XML ファイルまたはアプリケーションアーカイブファイルからオブジェクトをインポートできます。オブジェクトと依存オブジェクトをプロジェクトにインポートします。

1. [ファイル] > [インポート] をクリックします。
2. [Informatica] > [オブジェクトメタデータファイルのインポート (詳細)] を選択します。
3. [次へ] をクリックします。
4. [参照] をクリックして、インポートするエクスポートファイルを選択します。

5. **【次へ】** をクリックします。
6. **【ソース】** ペインで、インポートするオブジェクトを選択します。
7. **【ターゲット】** ペインで、オブジェクトのインポート先のプロジェクトを選択します。
8. **【ターゲットに追加】** をクリックして、オブジェクトをターゲットに追加します。



【ターゲットへの自動一致】 をクリックすると、現在のソース選択の下位オブジェクトがターゲット選択の名前、タイプ、親階層を基準にして個別に照合され、一致するオブジェクトが追加されます。

フォルダーまたはプロジェクトのすべてのオブジェクトをインポートする場合は、インポート先のフォルダーまたはプロジェクトを選択して **【ターゲットにコンテンツを追加】** をクリックします。

ヒント: オブジェクトを **【ソース】** ペインから **【ターゲット】** ペインの必要なプロジェクトにドラッグすることもできます。ソースとターゲットでオブジェクト階層を維持するには、Ctrl キーを押しながらドラッグします。

9. クリックして、重複するオブジェクトの処理方法を指定します。
インポートするオブジェクトの名前を変更するか、インポートするオブジェクトで既存のオブジェクトを上書きするか、既存のオブジェクトを再利用することができます。デフォルトでは、重複するすべてのオブジェクトの名前が変更されます。
10. **【次へ】** をクリックします。
インポートファイルの依存オブジェクトが一覧表示されます。
11. 依存オブジェクトをターゲットフォルダーまたはプロジェクトに追加します。
12. **【次へ】** をクリックします。
インポート中の参照テーブルデータが一覧表示されます。追加の参照テーブル設定を指定します。
13. **【次へ】** をクリックします。

インポート対象オブジェクトの概要が表示されます。[ソースとターゲットのオブジェクトのリンク] をクリックし、ソースおよびターゲットのどちらかのオブジェクトを選択すると、[ソース] および [ターゲット] の表示ペインにあるオブジェクトがリンクされます。例えば、このオプションを選択してから [ソース] ペインでオブジェクトを選択すると、[ターゲット] ペインにある同じオブジェクトが選択されます。

14. [追加のインポート設定] ペインで、インポートファイルからターゲットドメインへの接続をマッピングします。オブジェクトの既存のタグを上書きするかどうかを選択することもできます。

15. [完了] をクリックします。

重複するプロジェクトの名前を変更するように選択した場合は、インポートウィザードによって、インポートするプロジェクトに「<Original Name>_<number of the copy>」という名前が付けられます。プロジェクトの名前はインポート後に変更できます。

以前の Informatica リリースからのオブジェクトのインポート

以前の Informatica リリースからエクスポートしたオブジェクトを現在のメタデータ形式にアップグレードしてから、アップグレードしたオブジェクトを現在の Informatica リリースにインポートできます。現在のリリースにオブジェクトをインポートする前に、エクスポートしたオブジェクトを現在のメタデータ形式にアップグレードする必要があります。

複数の Informatica リリースがインストールされている企業では、以前の Informatica リリースから現在の Informatica リリースにオブジェクトをインポートする場合があります。例えば、以前の Informatica リリースの開発ドメインからオブジェクトをエクスポートし、そのオブジェクトを現在の Informatica リリースのテストドメインにインポートする場合があります。

Informatica 10.1 または 10.1.1 のモデルリポジトリから、オブジェクトメタデータを含む.xml ファイルにオブジェクトをエクスポートします。エクスポートしたオブジェクトを現在のメタデータ形式にアップグレードするには、mrs UpgradeExportedObjects コマンドを使用します。このコマンドを実行するときに、エクスポートしたオブジェクトを含む.xml ファイルのパスとファイル名を指定する必要があります。

mrs UpgradeExportedObjects コマンドを実行する前に、.xml ファイルを変更しないでください。このコマンドを実行する前にファイルを変更すると、ファイル内のオブジェクトが正常にアップグレードされないことがあります。

mrs UpgradeExportedObjects コマンドでは、オブジェクトをアップグレードした後に、アップグレードしたオブジェクトを含む.xml ファイルを生成します。.xml ファイルのファイル名とパスを指定する必要があります。.xml ファイルを現在の Informatica リリースのリポジトリにインポートできます。

mrs UpgradeExportedObjects コマンドを実行するには、Informatica 10.2 のモデルリポジトリサービスにアクセスする必要があります。このコマンドを実行するときに、ドメイン内で実行されている Informatica 10.2 のモデルリポジトリサービスのサービス名を指定する必要があります。

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

付録 A

データ型リファレンス

この付録では、以下の項目について説明します。

- [データ型リファレンスの概要, 217 ページ](#)
- [トランスフォーメーションデータ型, 218 ページ](#)
- [複合ファイルデータ型とトランスフォーメーションデータ型, 229 ページ](#)
- [フラットファイルとトランスフォーメーションデータ型, 237 ページ](#)
- [DB2 for LUW とトランスフォーメーションデータ型, 238 ページ](#)
- [DB2 for i5/OS および DB2 for z/OS のデータ型とトランスフォーメーションデータ型, 239 ページ](#)
- [JDBC のデータ型とトランスフォーメーションのデータ型, 241 ページ](#)
- [Microsoft SQL Server のデータ型とトランスフォーメーションデータ型, 242 ページ](#)
- [非リレーショナルのデータ型とトランスフォーメーションのデータ型, 245 ページ](#)
- [ODBC データ型とトランスフォーメーションデータ型, 247 ページ](#)
- [Oracle のデータ型およびトランスフォーメーションデータ型, 248 ページ](#)
- [SAP HANA およびトランスフォーメーションデータ型, 251 ページ](#)
- [XML データ型とトランスフォーメーションデータ型, 253 ページ](#)
- [データの変換, 255 ページ](#)

データ型リファレンスの概要

マッピングを作成する場合は、ソースからのデータ読み込み、変換、ターゲットへの書き込みを Data Integration Service に実行させるための指示セットを作成します。データ統合サービスは、マッピング内の最初のトランスフォーメーションから始まるマッピング内のデータフロー、およびマッピング内の各ポートに割り当てられているデータ型に基づいて、データを変換します。

Developer tool には、次の 2 種類のデータ型が表示されます。

ネイティブデータ型

ネイティブデータ型は、物理データオブジェクトとして使用されるリレーショナルテーブルまたはフラットファイルに固有です。ネイティブデータ型は、物理データオブジェクトカラムのプロパティに表示されます。

トランスフォーメーションデータ型

トランスフォーメーションデータ型は、トランスフォーメーションに表示されるデータ型のセットです。ANSI SQL-92 汎用データ型に基づく内部データ型で、プラットフォーム間でデータを移動するときデータ

統合サービスによって使用されます。トランスフォーメーションデータ型は、マッピング内のすべてのトランスフォーメーションに表示されます。

トランスフォーメーションデータ型には、次のデータ型があります。

- プリミティブデータ型。単一のカラム位置にある 1 つのデータ値を表します。
- 複合データ型。単一のカラム位置にある複数のデータ値を表します。Spark エンジンで実行されるマッピングで複合データ型を使用して、複合ファイルの階層データを処理します。

データ統合サービスは、ソースデータを読み取るときに、ネイティブデータ型に対応するトランスフォーメーションデータ型に変換してから、データのトランスフォームを実行します。データ統合サービスは、ターゲットに書き込むときに、トランスフォーメーションデータ型に対応するネイティブデータ型に変換します。

マルチバイト文字セットを指定した場合、データ型は 3 バイトまで文字を格納する追加スペースをデータベース内に割り当てます。

トランスフォーメーションデータ型

以下の表に、トランスフォーメーションのデータ型を示します。

データ型	サイズ (バイト数)	説明
Array	文字数に制限はありません。	複合データ型。 複雑なソースとターゲットを持つ配列を使用できます。
Bigint	8 バイト	-9,223,372,036,854,775,808～ 9,223,372,036,854,775,807 精度 19、位取り 0。 整数値。
Binary	精度	1～104,857,600 バイト フラットファイルソースでバイナリデータは使用できません。
Date/Time	16 バイト	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 精度 29、位取り 9 (精度はナノ秒まで) 日付値と時刻値が結合されたデータ型。
Decimal	8 バイト (高精度がオフまたは 38 より大きい場合) 16 バイト (精度が 18 以下で高精度がオンの場合) 20 バイト (精度が 18 より大きく、28 以下の場合) 24 バイト (精度が 28 より大きく、38 以下の場合)	宣言された精度と位取りを持つ 10 進型の値。位取りは、精度以下にする必要があります。 最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。

データ型	サイズ (バイト数)	説明
Double	8 バイト	倍精度の浮動小数点数値。 精度と位取りは編集できます。スケールは精度以下にする必要があります。
Integer	4 バイト	-2,147,483,648～2,147,483,647 精度 10、位取り 0 整数値。
Map	文字数に制限はありません。	複合データ型。 複雑なソースとターゲットを持つマップを使用できます。
String	Unicode モード: (精度+1) * 2 ASCII モード: 精度+1	1～104,857,600 文字 固定長または可変長文字列。
構造	文字数に制限はありません。	複合データ型 複雑なソースとターゲットを持つ構造を使用できます。
Text	Unicode モード: (精度+1) * 2 ASCII モード: 精度+1	1～104,857,600 文字 固定長または可変長文字列。
timestampWithTZ	40 バイト	西暦 1947 年 8 月 1 日～西暦 2040 年 12 月 31 日 -12:00～+14:00 精度 36、スケール 9。 (精度はナノ秒まで) Timestamp with Time Zone データ型では、次のタイムゾーン地域はサポートされません。 - AFRICA_CAIRO - AFRICA_MONROVIA - エジプト - AMERICA_MONTREAL

Integer データ型

ソースからターゲットに整数データを渡して、整数データでトランスフォーメーションを実行できます。トランスフォーメーション言語は、Bigint および Integer のデータ型をサポートします。

トランスフォーメーションの integer データ型は正確な値を表します。

計算における整数値

計算に整数値を使用する場合、データ統合サービスでは、計算の前に整数値が浮動小数点数に変換されることがあります。例えば、MOD(12.00, 5)を評価する場合、データ統合サービスは、除算を実行する前に整数値「5」を浮動小数点数に変換します。データ統合サービスでは、高精度が有効にされているかどうかに応じて、整数値が Double 値または Decimal 値に変換されます。

データ統合サービスは、次の算術演算の整数値を変換します。

算術演算	高精度の無効	高精度の有効
小数点を使用できない関数と計算。 たとえば、整数の加算、減算、乗算、および CUME、MOVINGSUM、SUM などの関数。	変換なし ¹	Decimal
小数点を使用できる非科学関数と計算。 たとえば、整数の除算、および AVG、MEDIAN、 PERCENTILE などの関数。	Double	Decimal
すべての科学関数、EXP 関数、LN 関数、LOG 関数、POWER 関数、および SQRT 関数。	Double	ダブル

¹ 計算が範囲外の結果を算出した場合、統合サービスは行エラーを書き込みます。

トランスフォーメーションの Double データ型は最大 15 桁の精度をサポートし、Bigint データ型は最大 19 桁の精度をサポートします。このため、精度が 15 桁より大きい Bigint 値となる計算では、精度の損失が発生することがあります。

例えば、式トランスフォーメーションには以下の計算が含まれています。

POWER(BIGINTVAL, EXPVAL)

データ統合サービスでは、計算を開始する前に、POWER 関数への入力を Double 値に変換します。BIGINTVAL ポートに Bigint 値 9223372036854775807 が含まれている場合、データ統合サービスはこの値を 9.22337203685478e+18 に変換し、最後の 4 桁の精度が失われます。EXPVAL ポートに値 1.0 が含まれており、結果ポートが Bigint の場合、計算結果の 9223372036854780000 が最大 Bigint 値を超えているため、行エラーが作成されます。

結果が Decimal 値となる計算で Integer データ型を使用し、高精度を有効にする場合、データ統合サービスでは、整数値が Decimal 値に変換されます。

最大 28 桁精度の Decimal データ型をサポートするトランスフォーメーションの場合、結果で高精度モードの 28 桁を超える値が出ない限り計算の精度は失われません。この場合、データ統合サービスは結果を Double 型として保存します。ポート精度が 28 桁以下で、結果が高精度モードの 28 桁より大きい場合、データ統合サービスは行を拒否します。

最大 38 桁精度の Decimal データ型をサポートするトランスフォーメーションの場合、結果で高精度モードの 38 桁を超える値が出ない限り計算の精度は失われません。この場合、データ統合サービスは結果を Double 型として保存します。ポート精度が 38 桁以下で、結果が高精度モードの 38 桁より大きい場合、データ統合サービスは行を拒否します。

式の整数定数

Integration Service では、計算結果が整数の場合でも、式の定数を浮動小数点数として解釈します。例えば、式 INTVALUE+1000 では、高精度が有効でない場合、Integration Service は整数値「1000」を Double 値に変換します。高精度が有効な場合は、値「1000」を 10 進数値に変換します。値 1000 を整数値として処理するには、Integer データ型で変数ポートを作成して定数を保持し、式を変更して 2 つのポートを追加します。

NaN 値

NaN（非数）とは、通常、特に浮動小数点数の計算で、無効な入力オペランドでの演算の結果として返される値のことです。例えば、ゼロをゼロで除算する演算の場合、結果として NaN が返されます。

NaN の表示は、オペレーティングシステムおよびプログラミング言語によって異なります。例えば、以下のリストのような NaN の表示があります。

```
nan
NaN
NaN%
NAN
NaNQ
NaNS
qNaN
sNaN
1.#SNAN
1.#QNAN
```

Integration Service は、Win64EMT プラットフォームでは QNAN 値を 1.#QNAN に変換します。1.#QNAN は、NaN の有効な表示です。

フラットファイルへの整数値の書き込み

整数値を固定長のフラットファイルに書き込む場合、ファイルライターではデータが範囲内であることが確認されません。例えば、ターゲットカラムのフィールド長が少なくとも 13 の場合、ファイルライターによりターゲットの Integer カラムに結果 3,000,000,000 が書き込まれます。ファイル writer では、結果が整数値の有効範囲外であるため、行は拒否されません。

Binary データ型

マッピングにバイナリデータが含まれている場合、統合サービスがデータをソースからターゲットに移動するために必要なメモリを割り当てることができるように、トランスフォーメーションの Binary データ型の精度を設定します。

フラットファイルソースで Binary データ型は使用できません。

Date/Time データ型

Date/Time データ型は、グレゴリオ暦の西暦 1 年から西暦 9999 年までの年を扱います。西暦 9999 より先の年はエラーになります。

Date/Time データ型は、ナノ秒単位の精度で日付をサポートします。データ型の精度は 29 で、位取りは 9 です。ネイティブデータ型の中には、精度が低いものがあります。日付の値を含むソースをインポートする際、インポートプロセスはソースカラムから正しい精度をインポートします。例えば、Microsoft SQL Server の Datetime データ型の精度は 23 で、位取りは 3 です。日時の値を含む Microsoft SQL Server ソースをインポートする場合、マッピングソースの日付カラムは精度が 23 で、位取りが 3 になります。

Integration Service は、マッピングソースに指定されている精度で、ソースから日時の値を読み込みます。Integration Service は、日時の値を変換する際、29 桁までの精度をサポートします。例えば、ミリ秒の精度で日時の値をインポートする場合、式トランスフォーメーションの ADD_TO_DATE 関数を使用して日付にナノ秒を追加することができます。

より短い精度をサポートするターゲットカラムに日付/時刻値を書き込むと、Integration Service はターゲットカラムの精度まで値を切り詰めます。より長い精度をサポートするターゲットカラムに日付/時刻値を書き込むと、Integration Service は日時の値でサポートされていない部分にゼロを挿入します。

Timestamp with Time Zone

Timestamp with Time Zone は、タイムゾーンのオフセット (TIME_ZONE_HOUR: TIME_ZONE_MINUTE) が含まれる Timestamp データ型のバリエーションです。夏時間やタイムゾーンの地域名がこれに含まれる場合があ

ります。タイムゾーンのオフセットは、ローカルタイムゾーンと UTC（Coordinated Universal Time: 協定世界時）の差（時分単位）です。

例えば、「16-JUN-08 07.01.25.376000 PM -06:00」のようなタイムスタンプがあります。この例では、「-06:00」がタイムゾーンのオフセットです。

また、「05-JUN-2008 07:01:25.376000 PM America/Los_Angeles」のようなタイムスタンプがあります。この例では、「America/Los_Angeles」がタイムゾーンの地域名です。

Timestamp with Time Zone データ型を Developer tool にインポートすると、関連付けられたトランスフォーメーションデータ型は timestampWithTZ になります。

timestampWithTZ の精度は 36 で、スケールは 9 です。Timestamp with Time Zone の置換値の範囲は、-12:00 < UTC < +14:00 です。

データ統合サービスは、Timestamp with Time Zone データ型を Oracle およびフラットファイルデータオブジェクト内で処理できます。Timestamp with Time Zone データ型は、1947 年 8 月～2040 年 12 月の日付にのみ適用されます。

Timestamp with Time Zone データ型は、Developer tool 内では、以下のタイムゾーン地域をサポートしません。

- AFRICA_CAIRO
- AFRICA_MONROVIA
- エジプト
- AMERICA_MONTREAL

Timestamp with Time Zone データ型は、タイムゾーンデータベースバージョン 2015b の IANA 標準を使用します。Timestamp with Time Zone データ型を使用する場合は、データの破損を防ぐため、Oracle サーバーとクライアントに Oracle DST パッチ 24 が適用されていることを確認します。データオブジェクトのキャッシュを有効にする場合、データオブジェクトのキャッシュで使用するデータベースサーバーに Oracle DST パッチ 24 が適用されていることを確認します。

パッチが適用されていない場合は、Oracle Web サイトから Oracle DST パッチ 24 をダウンロードしてください。Oracle サーバーで Oracle DST パッチ 24 が使用されていることを確認するには、Oracle サーバーのコマンドプロンプトで以下のコマンドを実行します。

```
SELECT VERSION FROM v$timezone_file;
```

バージョンが 24 として表示されることを確認できます。Oracle サーバーで以下のコマンドを実行してもバージョンを確認できます。

```
SELECT TZ_VERSION FROM registry$database;
```

TZ_VERSION が 24 として表示されることを確認できます。

Timestamp with Local Time Zone

Timestamp with Local Time Zone は、タイムスタンプデータがデータベースタイムゾーンに正規化される場合の Timestamp データ型のバリエーションです。タイムゾーンの置換はカラムデータの一部ではありません。データ統合サービスがデータを読み取る場合、Oracle はデータ統合サービスのタイムゾーンのデータを返しません。

Timestamp with Local Time Zone データ型を Developer tool にインポートする場合、関連付けられるトランスフォーメーションのデータ型は日付/時刻です。

例えば、「04-APR-10 10.27.451 AM」などです。

Timestamp (9) with Local Time Zone の精度は 29、スケールは 9 です。これは、日付/時刻 (29,9) トランスフォーメーションデータ型にマッピングされます。

データ統合サービスが Timestamp with Local Time Zone データを読み取る、または書き込む場合にデフォルトのセッションタイムゾーンを設定するには、ORA_SDTZ 環境変数を指定します。ORA_SDTZ 環境変数を次の値に設定できます。

- オペレーティングシステムローカルタイムゾーン (「OS_TZ」)
- データベースタイムゾーン (「DB_TZ」)
- UTC からの絶対オフセット (例: 「-05:00」)
- タイムゾーン地域名 (例: 「America/Los_Angeles」)

Oracle でサポートされているタイムゾーン

次のインストール場所で、タイムゾーンファイルに Oracle でサポートされているすべてのタイムゾーン (AFRICA_CAIRO、AFRICA_MONROVIA、EGYPT、および AMERICA_MONTREAL の 4 つのタイムゾーンを除く) のリストが格納されていることを確認できます。

<Informatica インストールディレクトリ>/services/shared/timezones/timezones.txt

後に Oracle でタイムゾーンが追加サポートされたときに、それに準じてタイムゾーンを追加する場合は、新しいタイムゾーンを含む新しいファイルを同じインストール場所に配置します。

Decimal および Double データ型

10 進型データおよび倍精度浮動小数点数型データをソースからターゲットに渡して、10 進型データおよび倍精度浮動小数点数型データに対するトランスフォーメーションを実行することができます。

トランスフォーメーション言語では以下のデータ型をサポートしています。

Decimal

最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1~38 桁、スケールが 0~38 になります。最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1~28 桁、スケールが 0~28 になります。精度より大きなスケールや負の精度を持つ 10 進型の値は使用できません。トランスフォーメーションには 10 進型のデータ型に割り当てた任意の範囲が表示されますが、データ統合サービスでは、トランスフォーメーションに応じて、最大 38 桁または 28 桁の精度のみサポートします。

高精度を有効にし、ポート精度が 38 桁または 28 桁 (トランスフォーメーションに依存) より高い場合、データ統合サービスは結果を Double 型として保存します。

Double

倍精度の浮動小数点数値。

精度と位取りは編集できます。スケールは精度以下にする必要があります。

計算における Decimal 値と Double 値

以下の表に、データ統合サービスで高精度設定に基づいて 10 進値がどのように処理されるかを示します。

ポートのデータ型	精度	高精度オフ	高精度オン
Decimal	0～15	Decimal	Decimal
Decimal	最大 38 桁の精度を持つ Decimal データ型をサポートするトランスフォーメーションの場合、15～38。 最大 28 桁の精度を持つ Decimal データ型をサポートするトランスフォーメーションの場合、15～28。	Double	Decimal
Decimal	最大 38 桁の精度を持つ Decimal データ型をサポートするトランスフォーメーションの場合、38 より大きい。 最大 28 桁の精度を持つ Decimal データ型をサポートするトランスフォーメーションの場合、28 より大きい。	Double	Double

高精度を有効にすると、データ統合サービスは式関数の数値定数を Decimal 型に変換します。高精度を有効にしないと、データ統合サービスは数値定数を Double 型に変換します。

数値の最大精度を、トランスフォーメーションに応じて、28 または 38 桁よりも大きくすることができます。トランスフォーメーション関数を使用して計算またはトランスフォーメーションを実行する前に、大きい数字を切り詰めるか、丸めます。

最大 38 桁の精度の Decimal データ型をサポートするトランスフォーメーションでは、Decimal データ型を使用し、高精度を有効にして、最大 38 桁の精度を保証します。

ルックアップ条件や結合条件など、等価条件で使用するデータに倍精度浮動小数点数型のデータ型を使用しないでください。

Decimal 値と Double 値の精度の損失

計算結果が最大許容桁数を超える精度の値になる場合、Decimal と Double のデータ型を計算に使用すると精度の損失が発生する可能性があります。

精度が最大許容桁数を超える場合、データ統合サービスは結果を Double として保存します。

精度の損失は、中間結果の精度または位取りに変更があった場合は中間計算中にも発生する可能性があります。中間結果の精度と位取りは、最終結果の精度または位取りに影響する可能性があります。

高精度が無効になっている場合の精度の損失

高精度を無効にすると、データ統合サービスは 10 進値を倍精度浮動小数点数に変換します。10 進値が 15 桁を超える精度を持つ場合には、精度の損失が発生します。例えば、Decimal (20,0) を使って数値 40012030304957666903 を渡すマッピングがあるとします。高精度を無効にすると、データ統合サービスは 10 進値を倍精度浮動小数点数に変換し、 $4.00120303049577 \times 10^{19}$ を渡します。

中間の計算を含むトランスフォーメーションで式を使用すると、追加の精度の損失が発生する可能性があります。高精度を無効にすると、データ統合サービスは中間結果を double に変換し、低精度の中間結果を使用して式で次の計算を実行します。その結果、データ統合サービスは、引数の順序に基づいて、式に対して異なる出力を生成する可能性があります。例えば、高精度を無効にすると、 $\tan(90 \times \pi / 180)$ の出力は $\tan(90 / 180 \times \pi)$ の出力と異なります。データ統合サービスは、これらの式で次のように中間計算を実行します。

```
tan(90*$pi/180) = tan(1.5707963267950003) = -9.649380295141232E12
```

```
tan(90/180*$pi) = tan(1.570796326795) = -9.670099380792184E12
```

Double 値の丸め処理方式

システムの実行時ライブラリと、データベースが double データタイプ計算を行うコンピュータシステムの違いにより、予想外の結果になる場合があります。double データタイプは IEEE 794 標準に準拠します。データベースクライアントライブラリへの変更、データベースの異なるバージョン、システム実行時ライブラリへの変更は、算術的に同等な値のバイナリ表現に影響します。また、多くのシステム実行時ライブラリは偶数丸め方法や対称算術法を実装しています。偶数丸め方法は、数字が次の上下の数の中間にある場合、偶数の最小ビットの最も近い値に丸められます。例えば、偶数丸め方法では、0.125 は 0.12 に丸められます。対称算術法では末尾桁が 5 以上の場合は次の桁に丸められます。例えば、対称算術法では 0.125 は 0.13 に丸められ、0.124 は 0.12 に丸められます。

プラットフォームの違いによる計算結果への影響を少なくするため、Integration Service は有効桁数 15 桁の Double データタイプの値を格納しています。例えば、Windows で計算すると、数値 1234567890.1234567890 が返され、同じ計算を UNIX で実行すると、1234567890.1234569999 が返される場合、Integration Service はこの数値を 1234567890.1234600000 に変換します。

String データ型

トランスフォーメーションデータ型には、次の string データ型が含まれます。

- String
- Text

String および Text データ型は 104,857,600 文字までの同精度をサポートしますが、統合サービスは、ソースからターゲットに文字列データを移動する場合には String を、テキストデータを移動する場合には Text を使用します。一部のデータベースではテキストデータと文字列データの格納方法が異なるため、Integration Service ではこの 2 つの文字データタイプを区別する必要があります。通常、Char や Varchar などのサイズの小さい string データ型は、トランスフォーメーション内で String として表示されますが、Text、Long、および Long Varchar などのサイズの大きな文字列データ型は Text として表示されます。

トランスフォーメーション内では、String および Text を相互に入れ替えて使用できます。ただし、ルックアップトランスフォーメーションでは、対象のデータ型が一致している必要があります。データベースドライバは、データが正確に渡されるように、string データ型とトランスフォーメーションデータ型を一致させる必要があります。例えば、ルックアップテーブルの Varchar は、ルックアップトランスフォーメーションの String と一致しなければなりません。

複合データ型

複合データ型を使用して、トランスフォーメーション内の 1 つの行またはカラムの位置に複数のデータ値を表します。複合データ型を使用すると、Spark エンジンで実されるマッピングで、階層データを直接読み取り、処理し、書き込むことができます。

階層データを処理するには複合データ型をマッピング内の複合ポートに割り当てます。階層データの複合ポートを処理するポートを呼び出します。Developer tool のポートタイプ設定で複合ポートからのデータを処理するために使用する複合データ型を指定します。

Spark エンジンで実行されるマッピングのトランスフォーメーションで次の複合データ型を使用できます。

array

配列は、要素の順序付けられたコレクションです。要素には、integer や string などのプリミティブデータ型、または array、struct、map などの複合データ型を指定できます。配列内のすべての要素は、同じデータ型である必要があります。

map

マップには、キーと値のペアの順序付けされないコレクションが含まれています。値部分にはプリミティブデータ型、または複合データ型を指定できます。値には 1 つのデータ型のみ指定できます。

struct

構造体は、さまざまなデータ型の要素のコレクションです。struct データ型は、概念上はテーブルの行と似ています。このデータ型には、それぞれ定義済みのデータ型を持つ、固定された数の名前付きフィールドが含まれます。

複合データ型には、プリミティブデータ型または複合データ型を含めることができます。複合データ型内に含まれる文字数に制限はありません。

複合ポートではネストされたデータ型を使用できます。ネストされたデータ型は、他の複合データ型を含む複合データ型です。例えば、構造体の配列、または他の構造体の配列を含む構造体を作成できます。ネストされたデータ型には、最大 10 レベルのネストを含めることができます。

array データ型

array データ型は、要素の順序付けられたコレクションを表します。配列データを渡す、生成する、または処理するには、array データ型をポートに割り当てます。

配列は、0 から始まるインデックス付きリストです。配列インデックスは、配列要素の位置を示します。例えば、配列インデックス 0 は、配列の最初の要素を示します。トランスフォーメーション言語には、配列データを生成して処理する配列要素と関数にアクセスするための演算子があります。

配列は、1 次元または多次元にすることができます。1 次元配列は線形配列です。多次元配列は配列の配列です。array トランスフォーメーションデータ型は、最大 5 つの次元を持つことができます。

形式

array <data_type> []

このデータ型の引数を次の表に示します。

引数	説明
array	配列のカラムまたはポートの名前。
data_type	配列内の要素のデータ型。 要素は、プリミティブデータ型または複合データ型にすることができます。配列内のすべての要素は、同じデータ型である必要があります。
[]	添字として表される配列の次元。 単一の添字[] は、1 次元配列を表します。2 つの添字[][] は、2 次元配列を表します。 各次元の要素は、同じデータ型です。

配列の要素には名前がありません。配列内の要素の数は、各行によって異なる場合があります。

配列の例

1 次元配列

次の配列カラムは、顧客の電話番号を含む文字列要素の 1 次元配列を表します。

```
custphone string[]
```

次の例は、custphone 列のデータ値を示しています。

custphone

```
[205-128-6478,722-515-2889]
```

```
[107-081-0961,718-051-8116]
```

```
[107-031-0961,NULL]
```

2 次元配列

次の配列カラムは、顧客の作業と個人の電子メールアドレスを含む文字列要素の 2 次元配列を表します。

```
email_work_pers string[][]
```

次の例は、email_work_pers 列のデータ値を示しています。

email_work_pers

```
[john.baer@xyz.com,jbaer@xyz.com][john.baer@fgh.com,jbaer@ijk.com]
```

```
[bobbi_apperley@xyz.com,bapperl@xyz.com][apperlbob@fgh.com,bobbi@ijk.com]
```

```
[linda_bender@xyz.com,lbender@xyz.com][l.bender@fgh.com,NULL]
```

Map データ型

map データ型は、キーと値のペア要素の順序が指定されていないコレクションを表します。マップ要素は、1 つのものを別のものにマップするキーと値のペアです。マップデータを渡す、生成する、または処理するには、ポートに map データ型を割り当てます。

キーはプリミティブデータ型である必要があります。この値は、プリミティブデータ型または複合データ型にすることができます。複合データ型の値を持つ map データ型は、ネストされたマップになります。ネストされたマップには、最大で 3 つのレベルの map データ型ネストを含めることができます。

注: Spark エンジンを使用すると、マップデータ内でキーを重複させることができます。

トランスフォーメーション言語には、マップ要素にアクセスするための添字演算子が含まれます。また、マップデータを生成して処理するための関数も含まれます。

形式

```
map <primitive_type -> data_type>
```

このデータ型の引数を次の表に示します。

引数	説明
map	マップカラムまたはポートの名前。
primitive_type	マップ要素内のキーのデータ型。 キーはプリミティブデータ型である必要があります。
data_type	マップ要素内の値のデータ型。 この値は、プリミティブデータ型または複合データ型にすることができます。

マップの例

次のマップカラムは、顧客 ID と顧客名をマッピングするための整数キーと文字列値を持つマップデータを表します。

```
custid_name <integer -> string>
```

次の例は、custid_name カラムのデータ値を示しています。

custid_name

```
<26745 -> 'John Baer'>
```

```
<56743 -> 'Bobbi Apperley'>
```

```
<32879 -> 'Linda Bender'>
```

Struct データ型

struct データ型は、異なるデータ型の要素のコレクションを表します。struct データ型には、データの構造を定義するスキーマが関連付けられています。構造体データを渡す、生成する、または処理するには、struct データ型をポートに割り当てます。

struct データ型のスキーマによって、構造データの要素名と要素数が決まります。スキーマは、構造データ内の要素の順序も決定します。Informatica は、構造データのスキーマを表す複合データ型定義を使用します。

トランスフォーメーション言語には、構造要素にアクセスするための演算子があります。また、構造体データを生成、処理、解析したり、データのスキーマを変更したりするための関数もあります。

形式

```
struct {element_name1:value1 [, element_name2:value2, ...]}
```

構造のスキーマは次の形式で表されます。

```
schema {element_name1:data_type1 [, element_name2:data_type2, ...]}
```

このデータ型の引数を次の表に示します。

引数	説明
struct	構造のカラムまたはポートの名前。
schema	データの構造の定義、 schema は、構造要素の名前とデータ型を決定する名前と型のペアです。

引数	説明
element_name	構造要素の名前。
value	構造要素の値。
data_type	要素値のデータ型。 要素の値は、プリミティブデータ型または複合データ型にすることができます。構造の各要素は、異なるデータ型を持つことができます。

構造の例

次のスキーマは、顧客の住所を格納する構造データです。

```
address
{st_number:integer,st_name:string,city:string,state:string,zip:string}
```

次の例は、cust_address カラムの構造データ値を示しています。

cust_address

```
{st_number:154,st_name:Addison Ave,city:Redwood City,state:CA,zip:94065}
```

```
{st_number:204,st_name:Ellis St,city:Mountain View,state:CA,zip:94043}
```

```
{st_number:357,st_name:First St,city:Sunnyvale,state:CA,zip:94085}
```

複合ファイルデータ型とトランスフォーメーションデータ型

マッピングで複合データ型を使用して複合ファイル内の階層データを処理できます。

Spark および Databricks Spark エンジンで実行されるマッピングでは次の複合ファイルで複合データ型を使用できます。

- Avro
- JavaScript Object Notation (JSON)
- Optimized Row Columnar (ORC)
- Parquet

Avro のデータ型とトランスフォーメーションデータ型

Avro データ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

次の表で、データ統合サービスでサポートされる Avro のデータ型と、それに対応するトランスフォーメーションのデータ型を比較します。

Avro データ型	トランスフォーメーションデータ型	範囲
Array	Array	文字数に制限はありません。
Boolean	Integer	TRUE (1) または FALSE (0)
Byte	Binary	精度 4000。
Date	Date/Time	1 月 1 日、0001～12 月 31 日、9999。
Decimal	Decimal	宣言された精度と位取りを持つ 10 進型の値。位取りは、精度以下にする必要があります。 最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
Double	ダブル	精度 15。
固定長	Binary	1～104,857,600 バイト。
Float	Double	精度 15。
Int	Integer	-2,147,483,648～2,147,483,647 精度 10、スケール 0。
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807。 精度 19、スケール 0。
マップ	Map	文字数に制限はありません。
Record	Struct	文字数に制限はありません。
String	String	1～104,857,600 文字。
Time	Date/Time	1 日のうちの時間。精度はマイクロ秒まで。

Avro データ型	トランスフォーメーションデータ型	範囲
タイムスタンプ	Date/Time	1 月 1 日、0001 00:00:00～12 月 31 日、9999 23:59:59.997。 精度はマイクロ秒まで。
Union	["primitive_type complex_type", "null"]または["null", "primitive_type complex_type"]の共有体で表される対応するデータ型。	プリミティブデータ型または複合データ型に依存します。

Avro Union データ型

共有体は、フィールドが複数のデータ型を持つ可能性があることを示します。例えば、共有体はフィールドが string または null であることを示す場合があります。共有体は、データ型を含む JSON 配列として表されます。

Developer tool は、["primitive_type|complex_type", "null"]または["null", "primitive_type|complex_type"]の共有体のみを解釈します。Avro データ型は、対応するトランスフォーメーションデータ型に変換されます。

Avro のタイムスタンプデータ型のサポート

次の表に、Avro ファイル形式のタイムスタンプデータ型のサポート状況を示します。

タイムスタンプデータ型	Native	Spark
Timestamp_micros	○	はい
Timestamp_millis	○	×
Time_millis	○	×
Time_micros	○	×

サポートされていない Avro データ型

Developer tool でサポートされていない Avro のデータ型は以下のとおりです。

- Enum
- Null
- Timestamp_tz

JSON のデータ型とトランスフォーメーションデータ型

JSON データ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

次の表で、データ統合サービスでサポートされる JSON のデータ型と、それに対応するトランスフォーメーションのデータ型を比較します。

JSON	トランスフォーメーション	範囲
Array	Array	文字数に制限はありません。
Double	Double	精度 15。
Integer	Integer	-2,147,483,648～2,147,483,647。 精度 10、位取り 0 を選択します。
Object	Struct	文字数に制限はありません。
String	String	1～104,857,600 文字。

サポートされていない JSON のデータ型

Developer tool でサポートされていない JSON のデータ型は以下のとおりです。

- Date
- Decimal
- タイムスタンプ
- Enum
- 共有体

ORC のデータ型とトランスフォーメーションデータ型

ORC ファイルのデータ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

以下の表は、データ統合サービスでサポートされる ORC ファイルのデータ型と、対応するトランスフォーメーションデータ型を示しています。

ORC ファイルのデータ型	トランスフォーメーションデータ型	範囲と内容
BigInt	BigInt	-9223372036854775808～9,223,372,036,854,775,807。
Boolean	Integer	TRUE (1) または FALSE (0)
Char	String	1～104,857,600 文字。
Date	Date/Time	1 月 1 日、0001～12 月 31 日、9999。
Double	ダブル	精度 15 桁。

ORC ファイルのデータ型	トランスフォーメーションデータ型	範囲と内容
Float	Double	精度 15 桁。
Integer	Integer	-2,147,483,648～2,147,483,647。
SmallInt	Integer	-32,768～32,767。
String	String	1～104,857,600 文字。
タイムスタンプ	Date/Time	1 月 1 日、0001 00:00:00～12 月 31 日、9999 23:59:59.997。 精度はマイクロ秒まで。
TinyInt	Integer	-128～127。
Varchar	String	1～104,857,600 文字。

Spark エンジンまたは Databricks Spark エンジンでマッピングを実行して ORC ファイルをターゲットに書き込むと、データ統合サービスは、Char および Varchar データ型のデータを String として書き込みます。

注: ORC データ型を使用すると、Spark エンジンのみで実行されるマッピング内で複合ファイルオブジェクトの読み取りと書き込みが可能になります。

サポートされていない ORC のデータ型

Developer tool でサポートされていない ORC のデータ型は以下のとおりです。

- マップ
- リスト
- 構造
- 共有体

Parquet のデータ型とトランスフォーメーションデータ型

Parquet のデータ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

次の表で、データ統合サービスでサポートされる Parquet のデータ型と、それに対応するトランスフォーメーションのデータ型を比較します。

Parquet	トランスフォーメーション	範囲
Binary	Binary	1～104,857,600 バイト
Binary (UTF8)	String	1～104,857,600 文字
Boolean	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Date	Date/Time	1 月 1 日、0001～12 月 31 日、9999。

Parquet	トランスフォーメーション	範囲
Decimal	Decimal	<p>宣言された精度と位取りを持つ 10 進型の値。位取りは、精度以下にする必要があります。</p> <p>最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。</p> <p>最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。</p> <p>この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。</p>
Double	Double	精度 15。
Float	Double	精度 15。
Int32	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Int64	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0。
マップ	マップ	文字数に制限はありません。
Struct	Struct	文字数に制限はありません。
Time	Date/Time	1 日のうちの時間。精度はマイクロ秒まで。
タイムスタンプ	Date/Time	1 月 1 日、0001 00:00:00～12 月 31 日、9999 23:59:59.997。 精度はマイクロ秒まで。
group (LIST)	Array	文字数に制限はありません。

Parquet ファイルの読み取りまたは書き込みを行うために指定する Parquet スキーマは小文字である必要があります。Parquet は大文字と小文字を区別するスキーマではありません。

Parquet のタイムスタンプデータ型のサポート

次の表に、Parquet ファイル形式のタイムスタンプデータ型のサポート状況を示します。

タイムスタンプデータ型	Native	Spark
Timestamp_micros	○	×
Timestamp_millis	○	×
Time_millis	○	×

タイムスタンプデータ型	Native	Spark
Time_micros	○	いいえ
int96	○	はい

サポートされていない Parquet のデータ型

Developer tool でサポートされていない Parquet のデータ型は以下のとおりです。

- Timestamp_nanos
- Time_nanos
- Timestamp_tz

データ型のルールとガイドライン

データ型については、次のルールおよびガイドラインを考慮します。

- Avro データ型のサポート:
 - Date、Decimal、Timestamp データ型は、Cloudera CDH 6.3 ディストリビューションのネイティブ環境または Spark エンジンでマッピングを実行する場合に使用できます。
 - Time データ型は、Cloudera CDH 6.3 ディストリビューションのネイティブ環境でマッピングを実行する場合に使用できます。
- JSON データ型のサポート:
 - PowerExchange for Microsoft Azure Data Lake Storage Gen2 の場合、ネイティブ環境、Spark エンジン、および Databricks Spark エンジンで実行されるマッピングで JSON 形式の複合ファイルオブジェクトを読み取るおよび書き込むことができます。
 - その他のファイルベースのアダプタについては、Spark エンジンで実行されるマッピングでのみ JSON 形式の複合ファイルオブジェクトを読み取るおよび書き込むことができます。
- Parquet データ型のサポート:
 - Databricks エンジンで新しいマッピングを作成して実行し、文字列データ型または階層データ型の Parquet ファイルを読み取る前に、developerCore.ini ファイルで `-DINFA_HADOOP_DIST_DIR=hadoop\` Databricks_7.2 オプションを設定する必要があります。
 - developerCore.ini ファイルで `-DINFA_HADOOP_DIST_DIR=hadoop\<Distro>` オプションを設定して Parquet ファイルをインポートした場合、インポートされるメタデータの形式はディストリビューションに応じて異なります。Cloudera CDP 7.1 の場合は、メタデータが文字列としてインポートされ、サポートされている他のディストリビューションの場合は、メタデータが UTF8 としてインポートされます。
 - ミリ秒までの Date、Time、Timestamp データ型は、Hortonworks HDP 3.1、Azure HDInsight HDI 4.0、Cloudera CDP 7.1 ディストリビューションのネイティブ環境、Blaze、および Spark エンジンでマッピングを実行する場合に使用できます。
 - Date、Time_Millis、および Timestamp_Millis データ型は、ネイティブ環境または Map R6.1 の Spark エンジンでマッピングを実行する場合に使用できます。
 - Decimal データ型は、Cloudera CDH 6.3、Hortonworks HDP 3.1、5.20、Amazon EMR、MapR 6.1、Azure HDInsight HDI 4.0 ディストリビューションのネイティブ環境および Spark エンジンでマッピングを実行する場合に使用できます。
 - Date、Time、Timestamp、Decimal データ型は、Databricks Spark エンジンでマッピングを実行する場合に使用できます。

- 時間の値のない Date データ型を使用してマッピングを実行すると、データ統合サービスはタイムゾーンに基づいてターゲットの日付に時間を追加します。

例えば、次の Date データ型がソースで使用されている場合、

1980-01-09

ターゲットには次の値が生成されます。

1980-01-09 00:00:00

- ネイティブ環境でマッピングを実行し、ソースで Time データ型を使用すると、データ統合サービスは誤った日付の値をターゲットに書き込みます。

例えば、次の Time データ型がソースで使用されている場合、

1980-01-09 06:56:01.365235000

ターゲットでは次の誤った日付の値が生成されます。

1899-12-31 06:56:01.365235000

- ネイティブ環境でマッピングを実行し、ソースで Date データ型を使用すると、データ統合サービスは誤った時間の値をターゲットに書き込みます。

例えば、次の Date データ型がソースで使用されている場合、

1980-01-09 00:00:00

ターゲットでは次の誤った時間の値が生成されます。

1980-01-09 05:30:00

- Date、Time、Timestamp、Decimal データ型の読み取りと書き込みを行うマッピングを実行するには、developerCore.ini ファイルで -DINFA_HADOOP_DIST_DIR オプションを更新します。developerCore.ini ファイルは、次のディレクトリにあります。

<クライアントインストールディレクトリ>\clients\DeveloperClient\

以下のパスを developerCore.ini ファイルに追加します。

-DINFA_HADOOP_DIST_DIR=hadoop\<Hadoop distribution>_<version>

例: -DINFA_HADOOP_DIST_DIR=hadoop\CDH_6.3

- ネイティブ環境で Decimal データ型に最大 38 桁の精度を使用するには、データ統合サービスの EnableSDKDecimal38 カスタムプロパティを true に設定します。EnableSDKDecimal38 カスタムプロパティは、PowerExchange for HDFS を除くすべてのファイルベースの PowerExchange アダプタに該当します。

フラットファイルとトランスフォーメーションデータ型

フラットファイルのデータ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

次の表で、フラットファイルのデータ型とトランスフォーメーションのデータ型を比較します。

フラットファイルのデータ型	トランスフォーメーションデータ型	範囲
Bigint	Bigint	精度 19、位取り 0
Datetime	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日（精度はナノ秒まで）
Double	Double	精度 15
Int	Integer	-2,147,483,648～2,147,483,647
Nstring	String	1～104,857,600 文字
Number	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。 精度が 15 より大きい場合、データ統合サービスは低精度モードで Decimal 値を Double に変換します。
String	String	1～104,857,600 文字
TimestampWithTZ	TimestampWithTZ	西暦 1947 年 8 月 1 日～西暦 2040 年 12 月 31 日 -12:00～+14:00 精度 36、スケール 9。 (精度はナノ秒まで) Timestamp with Time Zone データ型では、次のタイムゾーン地域はサポートされません。 - AFRICA_CAIRO - AFRICA_MONROVIA - エジプト - AMERICA_MONTREAL 注: TimestampWithTZ は、ネイティブ環境のみで使用できます。

フラットファイルの数値カラムから数値以外のデータを読み込むと、データ統合サービスはその行を削除し、ログにメッセージを書き込みます。また、フラットファイルの日時カラムから日時以外のデータを読み込むと、データ統合サービスはその行を削除し、ログにメッセージを書き込みます。

DB2 for LUW とトランスフォーメーションデータ型

DB2 for LUW のデータ型は、データ統合サービスがプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマップされます。

次の表に、DB2 for LUW のデータ型とトランスフォーメーションデータ型の比較を示します。

データ型	範囲	トランスフォーメーション	範囲
Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0
Blob	1～2,147,483,647 バイト	Binary	1～104,857,600 バイト
Char	1～254 文字	String	1～104,857,600 文字
ビットデータの Char	1～254 バイト	Binary	1～104,857,600 バイト
Clob	1～2,447,483,647 バイト	Text	1～104,857,600 文字
Date	西暦 0001 年～西暦 9999 年 精度 19、位取り 0（精度は日付まで）	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 （精度はナノ秒まで）
Dbclob	1～1073741823 バイト	Dbclob	1～104,857,600 バイト
Decimal	精度 1～31、位取り 0～31	Decimal	精度 1～31、位取り 0～31
Float	精度 1～15	Double	精度 15
Graphic	1～127 バイト	Graphic	1～127 バイト
Integer	-2,147,483,648～2,147,483,647	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Long Varchar	32700 バイト	Long Varchar	32700 バイト
Long Vargraphic	163350 バイト	Long Vargraphic	163350 バイト
数値	精度 1～31、位取り 0～31	Decimal	精度 1～28、位取り 0～28
Smallint	-32,768～32,767	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Time ¹	24 時間制の時刻 精度 19、位取り 0 （精度は秒まで）	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 （精度はナノ秒まで）
タイムスタンプ	26 バイト 精度 26、位取り 6 （精度はマイクロ秒まで）	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 （精度はナノ秒まで）

データ型	範囲	トランスフォーメーション	範囲
Varchar	4,000 文字まで	String	1～104,857,600 文字
ビットデータの Varchar	4,000 バイトまで	Binary	1～104,857,600 バイト
Vargraphic	1～16336 バイト	Vargraphic	1～16336 バイト
1. データ統合サービスで DB2 の Time データ型をトランスフォーメーションの Date/Time データ型に変換すると、日付値が時刻値に追加されます。			

DB2 for i5/OS および DB2 for z/OS のデータ型とトランスフォーメーションデータ型

DB2 for i5/OS および DB2 for z/OS のデータ型は、IBM DB2 のデータ型と同じ方法で、トランスフォーメーションのデータ型に対応します。データ統合サービスは、トランスフォーメーションのデータ型を使用して、プラットフォーム間のデータ移動を行います。

以下の表に、DB2 for i5/OS および DB2 for z/OS のデータ型と、トランスフォーメーションのデータ型の比較を示します。

データ型	範囲	トランスフォーメーション	範囲
Bigint	-9,223,372,036,854,775,808～ 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808～ 9,223,372,036,854,775,807 精度 19、位取り 0
Char	1～254 文字	String	1～104,857,600 文字
ビットデータの Char	1～254 バイト	Binary	1～104,857,600 バイト
Date	西暦 0001 年～西暦 9999 年 精度 19、位取り 0（精度は日付 まで）	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 （精度はナノ秒まで）
Decimal	精度 1～31、位取り 0～31	Decimal	精度 1～31、位取り 0～31
Float	精度 1～15	Double	精度 15
Integer	-2,147,483,648～2,147,483,647	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Smallint	-32,768～32,767	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0

データ型	範囲	トランスフォーメーション	範囲
Time ¹	24 時間制の時刻 精度 19、位取り 0 (精度は秒まで)	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Timestamp ²	26 バイト 精度 26、位取り 6 (精度はマイクロ秒まで)	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Varchar	4,000 文字まで	String	1～104,857,600 文字
ビットデータの Varchar	4,000 バイトまで	Binary	1～104,857,600 バイト
<p>1. データ統合サービスで DB2 の Time データ型をトランスフォーメーションの Date/Time データ型に変換すると、日付値が時刻値に追加されます。</p> <p>2. DB2 for z/OS バージョン 10 の拡張精度タイムスタンプはトランスフォーメーションのデータ型に以下のように対応します。</p> <ul style="list-style-type: none"> - 位取りが 6 の場合、精度は 26 でトランスフォーメーションのデータ型は date/time です。 - 位取りが 0 の場合、精度は 19 でトランスフォーメーションのデータ型は string です。 - 位取りが 1～5 または 7～12 の場合、精度は 20 以上でトランスフォーメーションのデータ型は string です。 			

DB2 for i5/OS および DB2 for z/OS のサポートされていないデータ型

Developer ツールは DB2 for i5/OS および DB2 for z/OS の一部のデータ型をサポートしていません。

Developer ツールは DB2 for i5/OS および DB2 for z/OS の large object (LOB) データ型はサポートしていません。LOB カラムはリレーショナルテーブルオブジェクト内ではサポートされないものとして、ネイティブタイプである varchar、および精度と位取りが 0 で表示されます。カラムは、カスタマイズされたデータオブジェクトまたは出力にマッピングには投影されません。

JDBC のデータ型とトランスフォーメーションのデータ型

データ統合サービスでデータを JDBC ソースから読み取るときは、ネイティブデータタイプが対応する JDBC データ型に変換され、その後、トランスフォーメーションデータ型に変換されます。トランスフォーメーションデータ型は、プラットフォーム間でのデータの移動に使用されます。

以下の表は、JDBC データ型とトランスフォーメーションデータ型を比較します。

JDBC データ型	トランスフォーメーション	範囲
Bigint	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0
Binary*	バイナリ	1～104,857,600 バイト
Bit	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Blob*	バイナリ	1～104,857,600 バイト
ブール	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Char*	String	1～104,857,600 文字
Clob*	テキスト	1～104,857,600 文字
日付	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Decimal	Decimal	精度 1～28、位取り 0～28
ダブル	ダブル	精度 15
Float	Double	精度 15
Integer	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Long VarBinary*	バイナリ	1～104,857,600 バイト
Long Varchar*	テキスト	1～104,857,600 文字
Numeric	Decimal	精度 1～28、位取り 0～28
Real	ダブル	精度 15
Smallint	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0

JDBC データ型	トランスフォーメーション	範囲
Time	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
タイムスタンプ	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Tinyint	Integer	-2,147,483,648~2,147,483,647 精度 10、位取り 0
Varchar*	String	1~104,857,600 文字
Varbinary*	バイナリ	1~104,857,600 バイト
*ポートのデータサイズが 100 MB を超えている場合、デフォルトで Developer ツールはポート精度を 4000 に設定します。より大きいサイズのデータを処理するには、ポート精度を大きくします。		

Microsoft SQL Server のデータ型とトランスフォーメーションデータ型

Microsoft SQL Server のデータ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型に対応します。

次の表で、Microsoft SQL Server のデータ型とトランスフォーメーションデータ型を比較します。

Microsoft SQL Server	範囲	トランスフォーメーション	範囲
Binary	1~8,000 バイト	Binary	1~104,857,600 バイト
bigint	-9,223,372,036,854,775,808 - 9,223,372,036,854,775,807	bigint	-9,223,372,036,854,775,808~9,223,372,036,854,775,807 精度 19、位取り 0
Bit	1 ビット	String	1~104,857,600 文字
Char	1~8,000 文字	String	1~104,857,600 文字
Date	西暦 0001 年 1 月 1 日~西暦 9999 年 12 月 31 日	Date/Time	西暦 0001 年 1 月 1 日~西暦 9999 年 12 月 31 日
Datetime	西暦 1753 年 1 月 1 日~西暦 9999 年 12 月 31 日 精度 23、位取り 3 (精度は 3.33 ミリ秒まで)	Date/Time	西暦 0001 年 1 月 1 日~西暦 9999 年 12 月 31 日 (精度はナノ秒まで)

Microsoft SQL Server	範囲	トランスフォーメーション	範囲
日時 2	西暦 0001 年 1 月 1 日 00:00:00 ～ 西暦 9999 年 12 月 31 日 23:59:59.9999999	タイムスタンプ	精度 22～27
Decimal	精度 1～38、位取り 0～38	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
Float	-1.79E+308～1.79E+308	Double	精度 15
Image	1～2,147,483,647 バイト	Binary	1～104,857,600 バイト
Int	-2,147,483,648～2,147,483,647	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
通貨	-922,337,203,685,477.5807～ 922,337,203,685,477.5807	Decimal	精度 1～28、位取り 0～28
nchar	1 - 4000 文字	String	1～104,857,600 文字
ntext	1 - 1,073,741,823 バイト	Text	1～104,857,600 文字
Numeric	精度 1～38、位取り 0～38	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
nvarchar	1 - 4000 文字	String	1～104,857,600 文字
Real	-3.40E+38～3.40E+38	Double	精度 15
Smalldatetime	西暦 1900 年 1 月 1 日～西暦 2079 年 6 月 6 日 精度 19、位取り 0 (精度は分まで)	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)

Microsoft SQL Server	範囲	トランスフォーメーション	範囲
Smallint	-32,768～32,768	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Smallmoney	-214,748.3648～214,748.3647	Decimal	精度 1～28、位取り 0～28
Sysname	1～128 文字	String	1～104,857,600 文字
Text	1～2,147,483,647 文字	Text	1～104,857,600 文字
タイムスタンプ	8 バイト	Binary	1～104,857,600 バイト
Tinyint	0～255	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Uniqueidentifier	精度 38、位取り 0	String	Uniqueidentifier データを正常に移動または変更するために、データが次の形式であることを確認します。 XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX x は、0～9 の範囲の 16 進数または a～f です。
Varbinary	1～8,000 バイト	Binary	1～104,857,600 バイト
Varchar	1～8,000 文字	String	1～104,857,600 文字

Uniqueidentifier データタイプ

Uniqueidentifier は、グローバル一意識別子（GUID）を格納するための Microsoft SQL Server のデータタイプです。16 バイトのデータを格納できます。

Developer ツールは Uniqueidentifier データタイプを文字列として扱います。Uniqueidentifier データを移動または変更するには、Uniqueidentifier カラムを文字列カラムに接続します。Uniqueidentifier データを正常に移動または変更するために、データが次の形式であることを確認します。

XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

x は、0～9 の範囲の 16 進数または a～f です。

例えば、6F9619FF-8B86-D011-B42D-00C04FC964FF は有効な Uniqueidentifier 値です。

Developer ツールでは、36 文字に 16 バイトの Uniqueidentifier データを格納できます。ただし、Uniqueidentifier データは 2 つの中括弧内に表示できるため、Developer ツールでは 2 つの追加文字を精度に割り当てて、中括弧に対応しています。Uniqueidentifier カラムを文字列カラムに接続する際、Uniqueidentifier データを正常に移動または変更するために、文字列カラムの精度を 38 に設定します。

非リレーショナルのデータ型とトランスフォーメーションのデータ型

非リレーショナルのデータ型は、Data Integration Service がプラットフォーム間のデータ移動に使用するトランスフォーメーションのデータ型に対応します。

非リレーショナルのデータ型は、以下のタイプのデータ型に適用されます。

- Adabas
- IMS
- シーケンシャル
- VSAM

以下の表に、非リレーショナルのデータ型とトランスフォーメーションのデータ型の比較を示します。

非リレーショナル	精度	トランスフォーメーション	範囲
BIN	10	バイナリ	1～104,857,600 バイト
CHAR	10	String	1～104,857,600 文字 固定長または可変長文字列。
DATE	10	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 結合された日時の値、ナノ秒単位の精度。
DOUBLE	18	Double	精度 15 桁 倍精度浮動小数点数値。
FLOAT	7	Double	精度 15 桁 倍精度浮動小数点数値。
NUM8	3	Integer	精度 10、位取り 0 整数値。
NUM8U	3	Integer	精度 10、位取り 0 整数値。
NUM16	5	Integer	精度 10、位取り 0 整数値。
NUM16U	5	Integer	精度 10、位取り 0 整数値。
NUM32	10	Integer	精度 10、位取り 0 整数値。
NUM32U	10	Double	精度 15 桁 倍精度浮動小数点数値。

非リレーショナル	精度	トランスフォーメーション	範囲
NUM64	19	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。
NUM64U	19	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。
NUMCHAR	100	String	1～104,857,600 文字 固定長または可変長文字列。
PACKED	15	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。
TIME	5	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 結合された日時の値、ナノ秒単位の精度。
TIMESTAMP	5	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 結合された日時の値、ナノ秒単位の精度。
UNPACKED	15	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。
UZONED	15	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。
VARBIN	10	バイナリ	1～104,857,600 バイト
VARCHAR	10	String	1～104,857,600 文字 固定長または可変長文字列。
ZONED	15	Decimal	精度 1～28 桁、位取り 0～28 宣言された精度と位取りを持つ 10 進値。位取りは、精度以下にする必要があります。負の位取りを持つ値または 28 より大きい精度が宣言された値を渡した場合、倍精度に変換されます。

ODBC データ型とトランスフォーメーションデータ型

ODBC データ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションデータ型にマッピングされます。

次の表では、Microsoft Access、Excel などの ODBC データ型をトランスフォーメーションデータ型と比較します。

データ型	トランスフォーメーション	範囲
Bigint	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0
バイナリ	Binary	1～104,857,600 バイト
Bit	String	1～104,857,600 文字
文字	String	1～104,857,600 文字
日付	日付/時刻	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Decimal	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
Double	Double	精度 15
Float	Double	精度 15
Integer	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Long Varbinary	Binary	1～104,857,600 バイト
Nchar	String	1～104,857,600 文字
Nvarchar	String	1～104,857,600 文字
Ntext	Text	1～104,857,600 文字
Numeric	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
Real	Double	精度 15

データ型	トランスフォーメーション	範囲
Smallint	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
テキスト	テキスト	1～104,857,600 文字
Time	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
タイムスタンプ	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Tinyint	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Varbinary	Binary	1～104,857,600 バイト
Varchar	String	1～104,857,600 文字

Oracle のデータ型およびトランスフォーメーション データ型

Oracle のデータ型は、データ統合サービスがプラットフォーム間のデータ移動に使用するトランスフォーメーションのデータ型に対応します。

次の表で、Oracle のデータ型とトランスフォーメーションデータ型を比較します。

Oracle	範囲	トランスフォーメーション	範囲
Blob	4GB まで	Binary	1～104,857,600 バイト
Char (L)	1～2,000 バイト	String	1～104,857,600 文字
Clob	4GB まで	Text	1～104,857,600 文字
Date	紀元前 4712 年 1 月 1 日～ 西暦 4712 年 12 月 31 日 精度 19、位取り 0	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Float	精度 1～15、位取り 0	Double	精度 15、位取り 0

Oracle	範囲	トランスフォーメーション	範囲
Long	2GB まで	Text	1～104,857,600 文字 マッピングに Long データが含まれている場合、統合サービスはそれをトランスフォーメーションの文字列データ型に変換し、104,857,600 文字に切り詰めます。
Long Raw	2GB まで	Binary	1～104,857,600 バイト
Nchar	1～2,000 バイト	String	1～104,857,600 文字
Nclob	4GB まで	Text	1～104,857,600 文字
Number	精度 1～38	Double	精度 15
Number (P,S)	精度 1～38、 位取り 0～38	Decimal	最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。 この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。
Nvarchar2	1～4,000 バイト	String	1～104,857,600 文字
RAW	1～2,000 バイト	Binary	1～104,857,600 バイト
タイムスタンプ	紀元前 4712 年 1 月 1 日～ 西暦 9999 年 12 月 31 日 精度 19～29、位取り 0～9 (精度はナノ秒まで)	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)

Oracle	範囲	トランスフォーメーション	範囲
Timestamp with Time Zone	紀元前 4712 年 1 月 1 日～ 西暦 9999 年 12 月 31 日 -12:00～+14:00 精度 26 - 36、位取り 0 - 9 (精度はナノ秒まで)	timestampWithTZ	西暦 1947 年 8 月 1 日～西暦 2040 年 12 月 31 日 -12:00～+14:00 精度 36、スケール 9。 (精度はナノ秒まで) Timestamp with Time Zone データ型では、次のタイムゾーン地域はサポートされません。 - AFRICA_CAIRO - AFRICA_MONROVIA - エジプト - AMERICA_MONTR EAL
Timestamp with Local Time Zone	紀元前 4712 年 1 月 1 日～ 西暦 9999 年 12 月 31 日 精度 19～29、位取り 0～9 (精度はナノ秒まで)	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Varchar	1～4,000 バイト	String	1～104,857,600 文字
Varchar2	1～4,000 バイト	String	1～104,857,600 文字
XMLType	4GB まで	Text	1～104,857,600 文字

Number (P,S) データ型

Developer tool は、負の位取りを持つ Oracle Number(P,S)値をサポートしています。

ただし、精度 38 より大きい位取り、または負の精度を持つ Number (P,S) 値はサポートしていません。

負の位取りの Oracle Number をインポートした場合、Developer tool には Decimal データ型として表示されます。ただし、データ統合サービスにより倍精度に変換されます。

Char、Varchar、Clob データタイプ

データ統合サービスでは、Unicode データ移動モードを使用する場合、Oracle データベースのカラムに設定した長さのセマンティクスに応じて、Char カラム、Varchar カラム、および Clob カラムの精度が読み込まれます。

バイトセマンティクスを使用してカラム長を決定する場合は、データ統合サービスは精度をバイト数として読み込みます。Char セマンティクスを使用する場合は、データ統合サービスは精度を文字数として読み込みます。

サポートされていない Oracle のデータ型

Developer tool は Oracle の一部のデータ型をサポートしていません。

Developer tool でサポートされていない Oracle のデータ型は以下のとおりです。

- Bfile
- Interval Day to Second
- Interval Year to Month
- Mslabel
- Raw Mslabel
- Rowid

SAP HANA およびトランスフォーメーションデータ型

SAP HANA データ型は、Data Integration Service がプラットフォーム間のデータ移動に使用するトランスフォーメーションのデータ型に対応します。

以下の表に、SAP HANA データ型およびトランスフォーメーションデータ型の比較を示します。

SAP HANA データ型	範囲	トランスフォーメーションデータ型	範囲
英数字	精度 1 - 127	Nstring	1～104,857,600 文字
Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0
バイナリ	バイナリデータのバイトの格納に使用	Binary	1～104,857,600 バイト
Blob	2GB まで	Binary	1～104,857,600 バイト
Clob	2GB まで	Text	1～104,857,600 文字
Date	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 精度 10、位取り 0	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)

SAP HANA データ型	範囲	トランスフォーメーションデータ型	範囲
Decimal (精度, 位取り) または Dec (p, s)	精度 1 - 34	Decimal	<p>最大 38 桁の精度をサポートするトランスフォーメーションでは、精度が 1～38 桁、スケールが 0～38 になります。</p> <p>最大 28 桁の精度をサポートするトランスフォーメーションでは、精度が 1～28 桁、スケールが 0～28 になります。</p> <p>この最大桁数を上回る精度を指定した場合、データ統合サービスは 10 進値を高精度モードの倍精度浮動小数点数に変換します。</p>
ダブル	単精度 64 ビット浮動小数点数を指定	Double	精度 15
Float	精度 1 - 53	Double	精度 15
Integer	-2,147,483,648～2,147,483,647	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
NClob	2GB まで	Ntext	1～104,857,600 文字
Nvarchar	精度 1 - 5000	Nstring	1～104,857,600 文字
Real	単精度 32 ビット浮動小数点数を指定	Real	精度 7、位取り 0
Seconddate	0001-01-01 00:00:01 - 9999-12-31 24:00:00	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
Shorttext	可変長文字列を指定、このデータ型ではテキスト検索および文字列検索機能をサポート	Nstring	1～104,857,600 文字
Smalldecimal	精度 1 - 16	Decimal	精度 1～28、位取り 0～28
Smallint	-32,768～32,767	Small Integer	精度 5、位取り 0
Text	可変長文字列を指定、このデータ型ではテキスト検索機能をサポート	Text	1～104,857,600 文字
時間	24 時間制の時刻	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
タイムスタンプ	0001-01-01 00:00:00.0000000 - 9999-12-31 23:59:59.9999999	Date/Time	西暦 0001 年 1 月 1 日～西暦 9999 年 12 月 31 日 (精度はナノ秒まで)

SAP HANA データ型	範囲	トランスフォーメーションデータ型	範囲
Tinyint	0～255	Small Integer	精度 5、位取り 0
Varchar	精度 1 - 5000	String	1～104,857,600 文字
Varbinary	1 - 5000 バイト	Binary	1～104,857,600 バイト

XML データ型とトランスフォーメーションデータ型

XML のデータ型は、データ統合サービスがデータのプラットフォーム間の移動に使用するトランスフォーメーションデータ型にマッピングされます。

データ統合サービスは、W3C が 2001 年 5 月 2 日の勧告で指定した XML データ型をすべてサポートしています。しかし、データ統合サービスは XML の値の範囲全体には対応していない場合があります。XML データタイプの詳細については、次のサイトで XML データタイプの W3C 仕様を参照してください:

<http://www.w3.org/TR/xmlschema-2>

以下の表に、XML データ型とトランスフォーメーションデータ型の比較を示します。

データ型	トランスフォーメーション	範囲
anyURI	String	1～104,857,600 文字
base64Binary	バイナリ	1～104,857,600 バイト
boolean	String	1～104,857,600 文字
byte	Integer	-2,147,483,648～2,147,483,647
日付	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
dateTime	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日 (精度はナノ秒まで)
decimal	Decimal	精度 1～28、位取り 0～28
double	Double	15 桁精度
duration	String	1～104,857,600 文字
ENTITIES	String	1～104,857,600 文字
ENTITY	String	1～104,857,600 文字
float	Double	15 桁精度

データ型	トランスフォーメーション	範囲
gDay	String	1～104,857,600 文字
gMonth	String	1～104,857,600 文字
gMonthDay	String	1～104,857,600 文字
gYear	String	1～104,857,600 文字
gYearMonth	String	1～104,857,600 文字
hexBinary	バイナリ	1～104,857,600 バイト
ID	String	1～104,857,600 文字
IDREF	String	1～104,857,600 文字
IDREFS	String	1～104,857,600 文字
整数型	Integer	-2,147,483,648～2,147,483,647
Integer	Integer	-2,147,483,648～2,147,483,647
language	String	1～104,857,600 文字
長整数型	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
名前	String	1～104,857,600 文字
NCName	String	1～104,857,600 文字
negativeInteger	Integer	-2,147,483,648～2,147,483,647
NMTOKEN	String	1～104,857,600 文字
NMTOKENS	String	1～104,857,600 文字
nonNegativeInteger	Integer	-2,147,483,648～2,147,483,647
nonPositiveInteger	Integer	-2,147,483,648～2,147,483,647
normalizedString	String	1～104,857,600 文字
NOTATION	String	1～104,857,600 文字
positiveInteger	Integer	-2,147,483,648～2,147,483,647
QName	String	1～104,857,600 文字
short	Integer	-2,147,483,648～2,147,483,647
string	String	1～104,857,600 文字

データ型	トランスフォーメーション	範囲
時間	日付/時刻	西暦 0001 年 1 月 1 日 - 西暦 9999 年 12 月 31 日（精度はナノ秒まで）
トークン	String	1～104,857,600 文字
unsignedByte	Integer	-2,147,483,648～2,147,483,647
unsignedInt	Integer	-2,147,483,648～2,147,483,647
unsignedLong	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsignedShort	Integer	-2,147,483,648～2,147,483,647

データの変換

あるデータ型から別のデータ型にデータを変換できます。

別のデータ型に変換するには、以下のいずれかの方法を使用します。

- ポート間でデータを渡す（ポート対ポートのデータ変換）。
- トランスフォーメーション関数を使ってデータを変換する。
- トランスフォーメーション算術演算子を使ってデータを変換する。

ポート対ポートのデータ変換

データ統合サービスは、ポートのデータ型に基づいてデータを変換します。データがポートを通過するたびに、データ統合サービスはポートに割り当てられているデータ型を識別し、必要に応じてサポートされているデータ型に基づいてデータを変換します。

同じデータ型のポート間でデータを渡す

同じデータ型のポート間でデータを渡すためのマッピングを作成できます。同じデータ型のポート間でデータを渡す場合、データ統合サービスはデータまたはデータ型を変更せずにデータを渡します。

同じ数値データ型のポート間でデータを渡し、データがトランスフォーメーション間で転送される場合、データ統合サービスは必ずしもデータの転送先のポートの精度と位取りにデータを変換しません。

以下の方法を使用すると、異なる精度、位取り、および精度モードの 10 進型データをポート間で転送できます。

10 進型データを低精度モードで渡す

低精度モードでデータを 10 進型ポートに渡す場合、データ統合サービスは精度 15 および位取り 0 ですべての 10 進型ポートを倍精度浮動小数点数に変換します。例えば、低精度モードで 10 進型 (14, 3) から 10 進型 (9, 5) に値を渡すとします。データ統合サービスは、値を内部的に格納し、低精度モードでのデータの切り詰めは行いません。

高精度モードで位取りを下げて 10 進型データを渡す

高精度モードで位取りを下げて 10 進型ポート間でデータを 10 進型ポートに渡す場合は、位取りが切り詰められます。例えば、10 進型 (18, 5) から 10 進型 (18, 2) に値を渡すとします。18.01234 を渡す場合、データ統合サービスはデータの位取りを切り詰めて、出力データは 18.01 になります。

高精度モードで精度を下げて 10 進型データを渡す

高精度モードで精度を下げてデータを 10 進型ポートに渡すことができます。例えば、高精度モードで 10 進型 (19, 5) から 10 進型 (17, 2) に値を渡すとします。出力フィールドに 17 桁を超える値が含まれる場合、データ統合サービスはその行を拒否します。

異なるデータ型のポート間でデータを渡す

異なるデータ型のポート間でデータを渡す場合、データ統合サービスはトランスフォーメーション言語の変換関数を使用してデータ型間の変換を行います。

例えば、文字列のポートを整数のポートに接続するとします。データ統合サービスがマッピングを実行する場合は、TO_INTEGER 関数を使用して入力データを文字列データ型から整数データ型に変換します。

データ統合サービスがポート間の変換を実行する場合は、渡すデータが変換先のデータ型に対して有効である必要があります。データ統合サービスが変換できない値があると、トランスフォーメーションの行エラーが発生します。例えば、「9,000,000,000,000,000,000.777」という値を含む文字列のポートを Bigint 型のポートに接続するとします。この場合、データ統合サービスは String の値を Bigint の値に変換できずに、エラーを返します。

データ統合サービスは、トランスフォーメーション間、およびパイプライン内の最後のトランスフォーメーションとターゲットとの間でポート対ポート変換を行います。

以下の表に、データ統合サービスで実行されるポート対ポート変換を示します。

データ型	Bigint	整数型	10 進型	Double	String、Text	Date/Time	Binary	Timestamp with Time Zone
Bigint	いいえ	はい	はい	はい	はい	いいえ	いいえ	いいえ
Integer	はい	いいえ	はい	はい	はい	いいえ	いいえ	いいえ
Decimal	はい	はい	いいえ	はい	はい	いいえ	いいえ	いいえ
Double	はい	はい	はい	いいえ	はい	いいえ	いいえ	いいえ
String、Text	はい	はい	はい	はい	はい	はい	いいえ	はい
Date/Time	いいえ	いいえ	いいえ	いいえ	はい	はい	いいえ	いいえ
Binary	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	はい	いいえ
Timestamp with Time Zone	いいえ	いいえ	いいえ	いいえ	はい	いいえ	いいえ	はい

付録 B

キーボードショートカット

この付録では、以下の項目について説明します。

- [詳細マッピングオプションのキーボードショートカット, 257 ページ](#)
- [オブジェクトのキーボードショートカット, 258 ページ](#)
- [ポートのキーボードショートカット, 259 ページ](#)
- [\[トランスフォーメーション\] パレットのキーボードショートカット, 260 ページ](#)
- [ワークベンチのキーボードショートカット, 260 ページ](#)

詳細マッピングオプションのキーボードショートカット

キーボードショートカットを使用して、[詳細オプションを使用してマッピングを実行] ダイアログボックスに移動できます。

キーボードショートカットで次のタスクを実行できます。

マッピング設定の選択。

Shift+A を押します。

カスタムマッピング設定の指定。

Shift+C を押します。

デフォルトのデータ統合サービスの使用。

Shift+U を押します。

データ統合サービス。

Shift+I を押します。その他のオプションを表示するには、**Shift+B** を押します。

トレースレベルのオーバーライド。

Shift+T を押します。

最適化レベル。

Shift+Z を押します。

マッピングでのデフォルト値の適用。

Shift+D を押します。

パラメータセットの適用。

Shift+S を押します。その他のオプションを表示するには、**Shift+O** を押します。

パラメータファイルの適用。

Shift+F を押します。その他のオプションを表示するには、**Shift+O** を押します。

オブジェクトのキーボードショートカット

キーボードショートカットを使用するとエディタ内のオブジェクトで作業ができます。

エディタのオブジェクトを選択した場合、オブジェクトの外観を変更したり、オブジェクトを移動できます。オブジェクトが選択されているときには、緑色の境界線がオブジェクトの周囲に表示されます。オブジェクトがフォーカスされていることを示す点線も選択したオブジェクトの周囲に表示されます。その他のオブジェクトが選択できる状態でオブジェクトをフォーカスできます。エディタ内で複数のオブジェクトを選択する場合に選択したオブジェクトから他のオブジェクトにフォーカスを移動します。

注: エディタ内のオブジェクトの移動順序は上から下、左から右です。

キーボードショートカットで次のタスクを実行できます。

オブジェクトを選択します。

エディタが選択されているときに、**Tab** を押します。このオブジェクトもフォーカスされています。**Tab** を再度押して次のオブジェクトを選択します。

前のオブジェクトを選択します。

Shift+Tab を押します。

特定のオブジェクトを検索します。

Ctrl+O を押します。

次のオブジェクトをフォーカスします。

オブジェクトがフォーカスされているときに **Ctrl+Tab** を押すと、フォーカスを次のオブジェクトに移動できます。前のオブジェクトは選択されたままです。**Ctrl+Tab** を再度押して次のオブジェクトをフォーカスします。

複数のオブジェクトを個別に選択します。

オブジェクトがフォーカスされているときに **Ctrl+Tab** を押すと、フォーカスを次に選択するオブジェクトに移動できます。次に **Ctrl+Spacebar** を押してフォーカスしたオブジェクトを選択します。この手順を繰り返して複数のオブジェクトを個別に選択します。

選択したオブジェクトをアイコン化します。

Ctrl+I を押します。

選択したオブジェクトをリストアします。

Ctrl+R を押します。

選択したオブジェクトをサイズ変更します。

Ctrl+Shift を押し続けながら、**Up**、**Down**、**Left**、**Right** 方向キーを使用します。

選択したオブジェクトを移動します。

方向キーを使用すると、1 つまたは複数のオブジェクトをどの方向にも 1 ピクセルずつ移動できます。

オブジェクトからエディタに移動します。

1 つまたは複数のオブジェクトが選択されているかフォーカスされているときに、**Esc** を押します。

ポートのキーボードショートカット

キーボードショートカットを使用してポートで作業ができます。

ポートを選択した場合、ポートを編集してそのポートを他のポートにリンクできます。ポートが選択された場合、ポート全体が緑色で表示されます。ポートがフォーカスされていることを示す点線も選択したポートのポート値の周囲に表示されます。その他のポートが選択できる状態でポートをフォーカスできます。複数のポートを個別に選択する場合に 1 つのポートから他のポートにフォーカスを移動します。

キーボードショートカットで次のタスクを実行できます。

ポートを選択します。

オブジェクトが選択され、フォーカスされているときに、**Ctrl+G** を押します。**Up** および **Down** 方向キーを押して異なるポートを選択します。

複数のポートを選択します。

Shift+Up または **Shift+Down** を押します。

複数のポートを個別に選択します。

Ctrl キーを押しながら、**Up** および **Down** 方向キーを使用して選択するポートをフォーカスします。**Ctrl+Spacebar** を押してポートを選択します。この手順を繰り返して複数のポートを個別に選択します。

次のポート値をフォーカスして編集します。

Tab を押します。

前のポート値をフォーカスして編集します。

Shift+Tab を押します。

ポートからオブジェクトに移動します。

Esc キーを押します。

2 つのオブジェクトのポートをリンクします。

最初のオブジェクトでリンクするポートを選択し、**Ctrl+L** を押します。選択したポートは強調表示されます。2 番目のオブジェクトに移動してリンクするポートを選択します。**Ctrl+L** を押してポートをリンクします。

リンクモードを終了します。

Ctrl+Alt+L を押して最初のオブジェクトで選択したポートをクリアします。

リンクを選択します。

選択したポートで **Ctrl+G** を押して、そのポートからの出力リンクを選択します。

次のリンクを選択します。

Tab を押します。

前のリンクを選択します。

Shift+Tab を押します。

[トランスフォーメーション] パレットのキーボードショートカット

キーボードショートカットを使用して [トランスフォーメーション] パレットを操作できます。

キーボードショートカットで次のタスクを実行できます。

エディタから [トランスフォーメーション] パレットに移動する。

エディタが選択されているときに、**Ctrl+Shift+P** を押します。

[トランスフォーメーション] パレットで移動する。

Tab キーを押すか、矢印キーの上向きまたは下向きを押します。

[トランスフォーメーション] パレットからエディタに移動する。

Esc キーを押します。

ワークベンチのキーボードショートカット

キーボードショートカットを使用してワークベンチ内のビューやエディタに移動できます。

エディタを選択した場合、エディタ内を移動できます。選択された場合、緑色の境界線がエディタの周囲に表示されます。

エディタおよびビューをフォーカスできます。フォーカスされている場合、エディタまたはビューの周囲に青色の境界線が表示されます。点線がフォーカスされているビューのタブの周囲に表示されます。ビューがフォーカスされている場合、そのビューを選択したり、他のビューをフォーカスできます。

キーボードショートカットで次のタスクを実行できます。

ビューをフォーカスします。

エディタが選択されているときに、**Shift+Tab** を押します。**Left** および **Right** 方向キーを使用して同じエディタ内の他のビューをフォーカスします。また、**Shift+Tab** をワークベンチの別のビューがフォーカスされるまで押します。

ビューで最初の領域を選択します。

ビューがエディタ内でフォーカスされた場合、**Ctrl+Tab** を押します。**Tab** ボタンを 3 回押して次のウィジェットを選択します。

ビューで前の領域を選択します。

Ctrl+Shift+Tab を押します。

マッピングの [プロパティ] ビューで、説明を選択して参照します。

Ctrl+A を押します。

エディタを最小化します。

ビューがフォーカスされた場合、**Shift+Tab** を押して **[最小化]** コントロールを選択し、次に **Space** キーを押します。

エディタを最大化します。

ビューがフォーカスされている場合、**Shift+Tab** を押して **[最小化]** コントロールを選択します。**Right** 方向キーを使用して **[最大化]** コントロールを選択し、**Space** キーを押します。

ワークベンチの領域を選択します。

エディタが選択されている場合、**Ctrl+Tab** を押して、ビューなど、ワークベンチのその他の領域を選択します。**Ctrl+Tab** を再度押して次の領域を選択します。

ワークベンチの前の領域を選択します。

Ctrl+Shift+Tab を押します。

ダイアログボックスのエラーメッセージに移動します。

Ctrl+M を押します。

付録 C

接続プロパティ

この付録では、以下の項目について説明します。

- [接続プロパティの概要, 263 ページ](#)
- [Adabas 接続のプロパティ, 264 ページ](#)
- [Amazon Redshift 接続のプロパティ, 266 ページ](#)
- [Amazon S3 接続のプロパティ, 269 ページ](#)
- [ブロックチェーン接続プロパティ, 272 ページ](#)
- [Cassandra 接続のプロパティ, 273 ページ](#)
- [Confluent Kafka 接続, 275 ページ](#)
- [Databricks 接続プロパティ, 277 ページ](#)
- [Greenplum 接続プロパティ, 279 ページ](#)
- [Google Analytics 接続のプロパティ, 281 ページ](#)
- [Google BigQuery 接続のプロパティ, 281 ページ](#)
- [Google Cloud Spanner 接続のプロパティ, 283 ページ](#)
- [Google Cloud Storage 接続のプロパティ, 284 ページ](#)
- [Google PubSub 接続のプロパティ, 285 ページ](#)
- [Hadoop 接続プロパティ, 286 ページ](#)
- [HBase 接続プロパティ, 292 ページ](#)
- [HDFS 接続プロパティ, 293 ページ](#)
- [MapR-DB の HBase 接続のプロパティ, 295 ページ](#)
- [Hive 接続のプロパティ, 295 ページ](#)
- [HTTP 接続のプロパティ, 299 ページ](#)
- [IBM DB2 接続のプロパティ, 301 ページ](#)
- [IBM DB2 for i5/OS 接続のプロパティ, 304 ページ](#)
- [IBM DB2 for z/OS 接続のプロパティ, 308 ページ](#)
- [IMS 接続のプロパティ, 310 ページ](#)
- [JDBC 接続のプロパティ, 313 ページ](#)
- [JDBC V2 接続のプロパティ, 315 ページ](#)
- [JD Edwards EnterpriseOne 接続プロパティ, 317 ページ](#)
- [Kafka 接続のプロパティ, 318 ページ](#)
- [Kudu 接続プロパティ, 321 ページ](#)
- [LDAP 接続プロパティ, 322 ページ](#)

- [Microsoft Azure Blob ストレージ接続のプロパティ, 323 ページ](#)
- [Microsoft Azure Cosmos DB SQL API 接続のプロパティ, 324 ページ](#)
- [Microsoft Azure Data Lake Storage Gen1 接続のプロパティ, 325 ページ](#)
- [Microsoft Azure Data Lake Storage Gen2 接続のプロパティ, 326 ページ](#)
- [Microsoft Azure SQL Data Warehouse 接続プロパティ, 327 ページ](#)
- [MS SQL Server 接続のプロパティ, 329 ページ](#)
- [Netezza 接続のプロパティ, 333 ページ](#)
- [OData 接続のプロパティ, 334 ページ](#)
- [ODBC 接続のプロパティ, 335 ページ](#)
- [Oracle 接続のプロパティ, 336 ページ](#)
- [Salesforce 接続のプロパティ, 339 ページ](#)
- [Salesforce Marketing Cloud 接続のプロパティ, 340 ページ](#)
- [SAP 接続のプロパティ, 341 ページ](#)
- [シーケンシャル接続のプロパティ, 344 ページ](#)
- [Snowflake 接続プロパティ, 346 ページ](#)
- [Teradata Parallel Transporter 接続のプロパティ, 348 ページ](#)
- [Tableau 接続のプロパティ, 350 ページ](#)
- [Tableau V3 接続のプロパティ, 351 ページ](#)
- [Twitter Streaming 接続プロパティ, 352 ページ](#)
- [VSAM 接続のプロパティ, 353 ページ](#)
- [Web サービス接続のプロパティ, 356 ページ](#)
- [データベース接続における識別子のプロパティ, 357 ページ](#)

接続プロパティの概要

接続プロパティを使用すると、Informatica クライアントでデータソースに接続できます。

この章では、Informatica クライアントで作成および管理できる各接続の接続プロパティについて説明します。

Adabas 接続のプロパティ

Adabas データベースにアクセスするには Adabas 接続を使用します。Adabas 接続は、メインフレームデータベースタイプの接続です。Adabas 接続は Developer ツールで作成します。Adabas 接続は、Administrator ツールまたは Developer ツールで管理できます。

以下の表に、Adabas 接続のプロパティを示します。

オプション	説明
場所	Adabas に接続する PowerExchange リスナの場所のノード名。ノード名は、PowerExchange の構成ファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。
ユーザー名	データベースユーザー名。 サポートされている Linux または UNIX システムのデータベースで、PowerExchange LDAP ユーザー認証を有効にした場合、ユーザー名はエンタープライズユーザー名になります。詳細については、『PowerExchange リファレンスマニュアル』を参照してください。
パスワード	データベースユーザー名のパスワードまたは有効な PowerExchange のパスフレーズ。 PowerExchange パスフレーズは 9～128 文字の長さで指定し、次の文字を含めることができます。 - 英字（大文字と小文字） - 数字（0～9） - スペース - 次に示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? 注: 先頭に書かれている特殊文字はアポストロフィです。 パスフレーズに一重引用符 (')、二重引用符 (")、または通貨記号を含めることはできません。 パスフレーズを使用するには、DBMOVER メンバのセキュリティ設定を SECURITY=(1,N)以上にして PowerExchange リスナを実行します。詳細については、『PowerExchange リファレンスマニュアル』の「SECURITY 文」を参照してください。 IBM IRRPHREX 出口に指定できる文字は、PowerExchange パスフレーズに指定できる文字に影響しません。 注: 有効な RACF のパスフレーズは最長 100 文字です。PowerExchange は、検証のために RACF にパスフレーズを渡すときに 100 文字を超えるパスフレーズを切り捨てます。
コードページ	必須。データソースの読み取りや書き込みに使用するコードページの名前。通常この値は ISO コードページの名前です (ISO-8859-6 など)。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。接続のパススルーセキュリティを有効にすると、ドメインは、接続オブジェクトに定義されている資格情報ではなく、クライアントのユーザー名とパスワードを使用して対応するデータベースにログインします。

オプション	説明
暗号化タイプ	<p>データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - AES <p>デフォルトは [なし] です。</p> <p>注: [暗号化タイプ] および [レベル] 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。PowerExchange ネットワークへの SSL 認証の実装の詳細については、『PowerExchange リファレンスマニュアル』を参照してください。</p>
[暗号化] レベル	<p>[暗号タイプ] で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。</p> <ul style="list-style-type: none"> - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 <p>[暗号化タイプ] に [AES] を選択しない場合、このオプションは無視されます。デフォルトは 1 です。</p>
ペーシングサイズ	<p>オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ペーシングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。</p> <p>最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。</p>
行として解釈	<p>オプション。このオプションは、ペーシングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ペーシングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ペーシングサイズはキロバイトで表示されています。</p>
圧縮	<p>オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。</p>
オフロード処理	<p>オプション。バルクデータの処理をソースマシンからデータ統合サービスのマシンにオフロードするかどうかを制御します。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - AUTO。オフロード処理を使用するかどうかデータ統合サービスによって決定されます。 - はい。オフロード処理を使用します。 - いいえ。オフロード処理は使用しません。 <p>デフォルトは [AUTO] です。</p>
ワーカースレッド	<p>オプション。オフロード処理の有効時にデータ統合サービスがバルクデータを処理するために使用するスレッドの数。最適なパフォーマンスを得るためには、データ統合サービスマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。</p>

オプション	説明
配列サイズ	オプション。ワーカースレッドのためのストレージアレイのレコード数。このオプションは、[ワーカースレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 1~5000 です。デフォルトは 25 です。
書き込みモード	<p>オプション。データ統合サービスが PowerExchange リスナにデータを送信する際のモード。次のうちいずれかの書き込みモードを選択します。</p> <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。このオプションはエラーリカバリを優先する場合に選択します。ただし、このオプションを使用するとパフォーマンスが落ちる可能性があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションは、エラーが発生したときにターゲットテーブルを再ロードしてもよい場合に使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションはエラー検出も有効にします。このオプションは、CONFIRMWRITEOFF のスピードと CONFIRMWRITEON のデータの完全性を結合します。 <p>デフォルトは [CONFIRMWRITEON] です。</p>

Amazon Redshift 接続のプロパティ

Amazon Redshift 接続をセットアップする際には、接続プロパティを設定する必要があります。

次の表に、Amazon Redshift 接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[] \:;'"<,>./
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。データベースで Amazon Redshift を選択します。

[詳細] タブには、Amazon Redshift 接続の接続属性が含まれています。以下の表に、接続属性を示します。

プロパティ	説明
ユーザー名	Amazon Redshift アカウントのユーザー名。
パスワード	Amazon Redshift アカウントのパスワード。

プロパティ	説明
アクセスキー ID	Amazon S3 バケットアクセスキー ID。 注: AWS Identity and Access Management (IAM) 認証を使用しない場合は必須です。
シークレットアクセスキー	Amazon S3 バケットシークレットアクセスキー ID。 注: AWS Identity and Access Management (IAM) 認証を使用しない場合は必須です。
マスタ対称キー	オプション。クライアントサイド暗号化を有効にする場合に、256 ビットの AES 暗号化キーを Base64 形式で指定します。暗号化キーは、サードパーティ製ツールを使用して生成できます。 この値を指定する場合は、詳細ターゲットプロパティで、暗号化タイプとしてクライアントサイド暗号化を指定してください。
JDBC URL	Amazon Redshift 接続 URL。

プロパティ	説明
クラスタリー ジョン	<p>オプション。アクセスするバケットが存在する AWS クラスタリージョンです。</p> <p>【JDBC URL】 接続プロパティで指定したカスタム JDBC URL にクラスタリージョン名が含まれていない場合にクラスタリージョンを選択します。</p> <p>【クラスタリージョン】 と 【JDBC URL】 の両方の接続プロパティでクラスタリージョンを指定した場合、【JDBC URL】 接続プロパティで指定したクラスタリージョンは無視されます。</p> <p>【JDBC URL】 接続プロパティで指定したクラスタリージョン名を使用するには、このプロパティでクラスタリージョンとして【なし】を選択します。</p> <p>次のいずれかのクラスタリージョンを選択します。</p> <p>次のいずれかのリージョンを選択します。</p> <ul style="list-style-type: none"> - アジアパシフィック（ムンバイ） - アジアパシフィック（ソウル） - アジアパシフィック（シンガポール） - アジアパシフィック（シドニー） - アジアパシフィック（東京） - AWS GovCloud（米国） - カナダ（中部） - 中国（北京） - 中国（寧夏） - 欧州（アイルランド） - 欧州（フランクフルト） - EU（ロンドン） - EU（パリ） - 南米（サンパウロ） - 米国東部（オハイオ） - 米国東部（バージニア北部） - 米国西部（北カリフォルニア） - 米国西部（オレゴン） <p>デフォルトは【なし】です。</p> <p>PowerExchange for Amazon Redshift が使用する AWS SDK によってサポートされるクラスタリージョンに対してのみ、データの読み取りと書き込みを行うことができます。</p>
顧客マスタキ ー ID	<p>オプション。AWS Key Management Service（AWS KMS）によって生成された顧客マスタキー ID、またはアカウント間アクセス用のカスタムキーの Amazon リソースネーム（ARN）を指定します。Amazon S3 バケットが存在するリージョンに対応する顧客マスタキーを生成する必要があります。次のいずれかの値を指定できます。</p> <p>顧客が生成した顧客マスタキー</p> <p>クライアントサイドまたはサーバーサイドの暗号化を有効にします。</p> <p>デフォルトの顧客マスタキー</p> <p>クライアントサイドまたはサーバーサイドの暗号化を有効にします。アカウントの管理者ユーザーのみがデフォルトの顧客マスタキー ID を使用してクライアントサイド暗号化を有効にできます。</p>

Amazon S3 接続のプロパティ

Amazon S3 接続をセットアップする際には、接続プロパティを設定する必要があります。

次の表に、Amazon S3 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[]] \\:;'"<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	Amazon S3 接続タイプ。
アクセスキー	Amazon S3 バケットにアクセスするためのアクセスキー。次の認証方法に基づいてアクセスキー値を指定します。 <ul style="list-style-type: none">- 基本認証: 実際のアクセスキー値を指定します。- IAM 認証: アクセスキー値を指定しません。- 引き継がれたロールによる一時的なセキュリティ資格情報: Amazon S3 バケットへのアクセス権限のない IAM ユーザーのアクセスキーを指定します。
秘密鍵	Amazon S3 バケットにアクセスするためのシークレットアクセスキー。 秘密鍵はアクセスキーに関連付けられており、アカウントを一意に識別します。次の認証方法に基づいてアクセスキー値を指定します。 <ul style="list-style-type: none">- 基本認証: 実際のアクセスシークレット値を指定します。- IAM 認証: アクセスシークレット値を指定しません。- 引き継がれたロールによる一時的なセキュリティ資格情報: Amazon S3 バケットへのアクセス権限のない IAM ユーザーのアクセスシークレットを指定します。
IAM ロール ARN	動的に生成された一時的なセキュリティ資格情報を使用するためにユーザーに引き継がれた IAM ロールの ARN。 一時的なセキュリティ資格情報を使用して AWS リソースにアクセスする場合はこのプロパティの値を入力します。 IAM 認証とともに一時的なセキュリティ資格情報を使用する場合は、[アクセスキー] および [秘密鍵] 接続プロパティを入力しないでください。IAM 認証を使用せずに一時的なセキュリティ資格情報を使用する場合は、[アクセスキー] および [秘密鍵] 接続プロパティの値を入力する必要があります。 IAM ロールの ARN の取得方法の詳細については、AWS のマニュアルを参照してください。
フォルダパス	Amazon S3 オブジェクトへの完全なパス。パスには、バケット名と任意のフォルダ名を含める必要があります。 フォルダパスの末尾にスラッシュを使用しないでください。例: <バケット名>/<フォルダ名>。
マスタ対称キー	オプション。クライアントサイド暗号化を有効にする場合に、256 ビットの AES 暗号化キーを Base64 形式で指定します。マスタ対称キーは、サードパーティ製ツールを使用して生成できます。

プロパティ	説明
S3 アカウントタイプ	<p>Amazon S3 アカウントのタイプ。</p> <p>【Amazon S3 ストレージ】 または 【S3 互換ストレージ】 を選択します。</p> <p>Amazon S3 サービスを使用するには、【Amazon S3 ストレージ】 オプションを選択します。Scality RING などのサードパーティストレージプロバイダのエンドポイントを指定するには、【S3 互換ストレージ】 オプションを選択します。</p> <p>デフォルトでは、【Amazon S3 ストレージ】 が選択されています。</p>
REST エンドポイント	<p>S3 ストレージエンドポイント。</p> <p>【S3 互換ストレージ】 オプションを選択した場合に、S3 ストレージエンドポイントを HTTP/HTTPS 形式で指定します。例えば、http://s3.isv.scality.com と指定します。</p>
リージョン名	<p>アクセスするバケットが存在する AWS リージョンを選択します。</p> <p>次のいずれかのリージョンを選択します。</p> <ul style="list-style-type: none"> - アジアパシフィック (ムンバイ) - アジアパシフィック (ソウル) - アジアパシフィック (シンガポール) - アジアパシフィック (シドニー) - アジアパシフィック (東京) - AWS GovCloud (米国) - カナダ (中部) - 中国 (北京) - 中国 (香港) - 中国 (寧夏) - 欧州 (アイルランド) - 欧州 (フランクフルト) - EU (ロンドン) - EU (パリ) - 南米 (サンパウロ) - 米国東部 (オハイオ) - 米国東部 (バージニア北部) - 米国西部 (北カリフォルニア) - 米国西部 (オレゴン) <p>デフォルトは【米国東部 (バージニア北部)】です。</p> <p>S3 互換ストレージには適用されません。</p>

プロパティ	説明
顧客マスターキー ID	<p>オプション。AWS Key Management Service (AWS KMS) によって生成された顧客マスターキー ID またはエイリアス名、またはアカウント間アクセス用のカスタムキーの Amazon リソースネーム (ARN) を指定します。Amazon S3 バケットが存在するリージョンの顧客マスターキーを生成する必要があります。</p> <p>次のいずれかの値を指定できます。</p> <p>顧客が生成した顧客マスターキー</p> <p>クライアントサイドまたはサーバーサイドの暗号化を有効にします。</p> <p>デフォルトの顧客マスターキー</p> <p>クライアントサイドまたはサーバーサイドの暗号化を有効にします。アカウントの管理者ユーザーのみがデフォルトの顧客マスターキー ID を使用してクライアントサイド暗号化を有効にできます。</p>
フェデレーション SSO IDp	<p>AWS アカウントで使用する、統合ユーザーシングルサインオンの SAML 2.0 対応 ID プロバイダ。</p> <p>PowerExchange for Amazon S3 では、[ADFS 3.0] ID プロバイダのみがサポートされます。統合ユーザーシングルサインオンを使用しない場合は、[なし] を選択します。</p>

統合ユーザーシングルサインオン接続のプロパティ

【統合 SSO IdP】で [ADFS 3.0] を選択した場合は、次のプロパティを設定します。

プロパティ	説明
統合ユーザー名	ID プロバイダ経由で AWS アカウントにアクセスする統合ユーザーのユーザー名。
統合ユーザーパスワード	ID プロバイダ経由で AWS アカウントにアクセスする統合ユーザーのパスワード。
IdP SSO URL	AWS に使用する ID プロバイダのシングルサインオン URL。
SAML ID プロバイダ ARN	ID プロバイダを信頼できるプロバイダとして登録するために AWS 管理者が作成した、SAML ID プロバイダの ARN。
ロール ARN	統合ユーザーに引き継がれた IAM ロールの ARN。

ブロックチェーン接続プロパティ

ブロックチェーン接続を設定するときは、接続プロパティを設定する必要があります。

以下の表に、ブロックチェーン接続の一般的な接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。接続を識別するために使用できる文字列を入力します。説明は、4,000 文字を超えることはできません。
Swagger ファイルパス	ブロックチェーンと通信するための REST API が含まれた Swagger ファイルパスの絶対パス。Swagger ファイルは、データ統合サービスマシン上に保存された JSON ファイルである必要があります。YAML など異なるファイル形式の Swagger ファイルは JSON 形式に変換してください。
ベース URL	必須。ブロックチェーンのアセットにアクセスするために使用されるベース URL です。
認証タイプ*	ランタイムエンジンが REST サーバーへの接続に使用する認証方法。[なし]、[基本]、[ダイジェスト]、[OAuth] を使用できます。
認証ユーザー ID*	REST サーバーの認証を受けるユーザー名。
認証パスワード*	REST サーバーの認証を受けるユーザー名のパスワード。
OAuth コンシューマキー*	認証タイプが [OAuth] の場合に必要です。REST サーバーに関連付けられたクライアントキー。
OAuth コンシューマシークレット*	認証タイプが [OAuth] の場合に必要です。REST サーバーに接続するクライアントのパスワード。
OAuth トークン*	認証タイプが [OAuth] の場合に必要です。REST サーバーに接続するためのアクセストークン。
OAuth トークンシークレット*	認証タイプが [OAuth] の場合に必要です。OAuth トークンに関連付けられるパスワード。
プロキシタイプ*	プロキシのタイプ。[プロキシなし]、[プラットフォームプロキシ]、[カスタム] を使用できます。
プロキシの詳細*	<host>:<port>形式を使用したプロキシ設定。
トラストストアのファイルパス*	SSL 証明書を含むトラストストアファイルの絶対パス。

プロパティ	説明
トラストストアのパスワード*	トラストストアファイルのパスワード。
キーストアのファイルパス*	REST サーバーとの双方向の安全な接続を確立するために必要なキーと証明書を含むキーストアファイルの絶対パス。
キーストアのパスワード*	キーストアファイルのパスワード
詳細プロパティ	<p>ブロックチェーンのアセットにアクセスするための詳細プロパティのリスト。名前と値のペアをセミコロンで区切って詳細プロパティを指定します。</p> <p>以下の詳細プロパティを使用できます。</p> <ul style="list-style-type: none"> - baseUrl。Swagger ファイルにベース URL が含まれていない場合に必要です。ブロックチェーンのアセットにアクセスするために使用されるベース URL です。 - X-API-KEY。API キーを使用して REST サーバーの認証を受ける場合に必要です。 <p>接続で設定する詳細プロパティは、ブロックチェーンデータオブジェクトの対応する詳細プロパティの値をオーバーライドします。例えば、接続とデータオブジェクトの両方にベース URL を指定すると、接続の値はデータオブジェクトの値をオーバーライドします。</p> <p>注: 詳細プロパティには、[オペレーションレベル] > [オブジェクトレベル] > [接続レベル] という優先レベルがあります。[オペレーションレベル] に設定されたプロパティは、[オブジェクトレベル] または [接続レベル] に設定されたプロパティよりも優先されます。</p>
クッキー	<p>REST API の実装方法に応じて必要となります。REST サーバーに渡されるクッキーの情報を指定する、クッキーのプロパティのリストです。名前と値のペアをセミコロンで区切ってプロパティを指定します。</p> <p>接続に設定するクッキープロパティは、ブロックチェーンデータオブジェクトの対応するクッキープロパティの値をオーバーライドします。</p>
<p>* プロパティは無視されます。この機能を使用するには、プロパティを詳細プロパティとして設定し、Swagger ファイルのプロパティ名に基づいて名前と値のペアを指定します。</p> <p>例えば、次の名前と値のペアを設定して、基本的な認証を使用します。</p> <pre>Authorization=Basic <credentials></pre> <p>注: [テスト接続] を使用してブロックチェーン接続を検証することはできません。</p>	

Cassandra 接続のプロパティ

Box 接続を設定するときは、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Cassandra 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。 ID では大文字と小文字が区別されません。ID は 255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。 Data Engineering Streaming には適用されません。
タイプ	接続タイプ。 [Cassandra] を選択します。
ホスト名	Cassandra サーバーのホスト名または IP アドレスです。
ポート	Cassandra サーバーのポート番号。 デフォルトは 9042 です。
ユーザー名	Cassandra サーバーにアクセスするためのユーザー名。
パスワード	Cassandra サーバーにアクセスするためのユーザー名に対応するパスワード。
デフォルトキースペース	デフォルトで使用する Cassandra キースペースの名前。
SQL 識別子文字	データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。 データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリを生成する場合は、識別子を区切り文字で囲みません。 データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。
SSL モード	[無効] を選択します。 PowerExchange for Cassandra JDBC には適用できません。 接続に使用する暗号化タイプを示す SSL モード。次の SSL モードからモードを選択できます。 - 無効 - 一方向 - 双方向

プロパティ	説明
SSL トラストストアパス	PowerExchange for Cassandra JDBC に対して、または【無効】状態の SSL モードが選択されている場合は適用されません。 信頼済み SSL サーバーの証明書を含む SSL トラストストアファイルの絶対パスおよびファイル名。
SSL トラストストアパスワード	PowerExchange for Cassandra JDBC に対して、または【無効】状態の SSL モードが選択されている場合は適用されません。 SSL トラストストアのパスワード。
SSL キーストアパス	PowerExchange for Cassandra JDBC に対して、または【無効】状態の SSL モードが選択されている場合は適用されません。 SSL サーバーのプライベートキーと証明書を含む SSL キーストアファイルの絶対パスおよびファイル名。
SSL キーストアパスワード	PowerExchange for Cassandra JDBC に対して、または【無効】状態の SSL モードが選択されている場合は適用されません。 SSL キーストアのパスワード。
追加接続プロパティ	以下の形式で、1 つ以上の JDBC 接続パラメータを入力します。 <param1>=<value>;<param2>=<value>;<param3>=<value> PowerExchange for Cassandra JDBC は、次の JDBC 接続パラメータをサポートしています。 <ul style="list-style-type: none"> - BinaryColumnLength - DecimalColumnScale - EnableCaseSensitive - EnableNullInsert - EnablePaging - RowsPerPage - StringColumnLength - VTTableNameSeparator

Confluent Kafka 接続

Confluent Kafka 接続はメッセージング接続です。Confluent Kafka 接続を使用して、ソースまたはターゲットとして、Kafka Broker または Confluent Kafka Broker にアクセスします。Developer tool または infacmd を使用して、Confluent Kafka 接続を作成および管理できます。

Confluent Kafka 接続を設定する場合は、次のプロパティを設定します。

- 接続が読み取りまたは書き込みを行う、Kafka Broker または Confluent Kafka Broker のリスト。
- データベースへの接続が失敗した場合に統合サービスが再接続を試行する秒数。
- Confluent Kafka メッセージングブローカのバージョン。

全般プロパティ

次の表に、Confluent Kafka 接続の一般的な接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。接続を識別するために使用できる文字列を入力します。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。ドメイン名を選択します。
タイプ	接続タイプ。[Messaging/ConfluentKafka] を選択します。

Confluent Kafka Broker のプロパティ

次の表に、Confluent Kafka 接続の Kafka Broker プロパティを示します。

プロパティ	説明
Kafka Broker リスト	Confluent Kafka メッセージングブローカの設定を保持する Confluent Kafka Broker のカンマ区切りリスト。 Confluent Kafka Broker を指定するには、次の形式を使用します: <IP address>:<port>
再試行タイムアウト	データ統合サービスが、データの読み取りまたは書き込みのために Confluent Kafka Broker への再接続を試行した後の秒数。指定した時間にソースまたはターゲットが利用できない場合は、データ損失を回避するために、マッピング実行が停止します。
Kafka Broker のバージョン	Confluent Kafka メッセージングブローカのバージョン。
追加接続プロパティ	オプション。Kafka Broker に接続するための接続プロパティのカンマ区切りリスト。
スキーマレジストリの URL	接続するスキーマレジストリプロバイダの場所とポート。

追加接続プロパティ

追加の接続プロパティを指定する場合は、次の構文を使用できます。

```
request.timeout.ms=<value>,session.timeout.ms=<value>,  
fetch.max.wait.ms=<value>,heartbeat.interval.ms=<value>,  
security.protocol=SASL_PLAINTEXT,sasl.kerberos.  
service.name=<kerberos name>,sasl.mechanism=GSSAPI,  
sasl.jaas.config=com.sun.security.auth.module.  
Krb5Login Module;required useKeyTab=true
```

```
doNotPrompt=true storeKey=true client=true  
keyTab="<Keytab Location>" principal="<principal>";
```

SSL プロパティ

次の表に、Confluent Kafka 接続の SSL プロパティを示します。

プロパティ	説明
SSL モード	オプション。接続に使用する暗号化タイプを示す SSL モード。 次のいずれかの SSL モードを選択できます。 <ul style="list-style-type: none">- 無効- 一方向- 双方向 デフォルト値は [無効] です。
SSL トラストストア ファイルパス	[一方向] の SSL モードが選択されている場合に必要です。 信頼済み SSL サーバーの証明書を含む SSL トラストストアファイルの絶対パスおよびファイル名。
SSL トラストストア パスワード	[一方向] の SSL モードが選択されている場合に必要です。 SSL トラストストアのパスワード。
SSL キーストアファ イルパス	[双方向] の SSL モードが選択されている場合に必要です。 SSL サーバーのプライベートキーと証明書を含む SSL キーストアファイルの絶対パスおよびファイル名。
SSL キーストアパス ワード	[双方向] の SSL モードが選択されている場合に必要です。 SSL キーストアのパスワード。
追加セキュリティプ ロパティ	オプション。安全な方法で Confluent Kafka Broker に接続するための接続プロパティのカンマ区切りリスト。

infacmd を使用した Confluent Kafka 接続の作成

infacmd コマンドラインプログラムを使用して、Confluent Kafka 接続を作成できます。

UNIX で Confluent Kafka 接続を作成するには、次のコマンドを実行します。

```
sh infacmd.sh createConnection -dn <domain name> -un <domain user> -pd <domain password> -cn <connection name>  
-cid <connection id> -ct ConfluentKafka -o "kfkBrkList='<host1:port1>,<host2:port2>,<host3:port3>'  
kafkabrokerversion='<version>' schemaregistryurl='<schema registry URL>'"
```

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

Databricks 接続プロパティ

Databricks 接続を使用して、Databricks クラスタでマッピングを実行します。

Hadoop 接続は、クラスタ型の接続です。Hadoop 接続は、管理者ツールまたは Developer tool で作成および管理できます。infacmd を使用して Hadoop 接続を作成できます。データ統合サービスと Hadoop クラスタ間の通信を有効にするには、Hadoop 接続のプロパティを設定します。

以下の表に、Hadoop 接続の一般的な接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。`~`!\$%^&*()-+={ }\:;'"<,>./?
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
接続タイプ	[Databricks] を選択します。
クラスタ設定	Databricks 環境に関連付けられているクラスタ設定の名前。 クラウドプロビジョニング設定を設定しない場合に必要です。
クラウドプロビジョニング設定	Databricks クラウドプラットフォームと関連付けられているクラウドプロビジョニング設定の名前。 クラスタ設定を設定しない場合に必要です。
一時領域ディレクトリ	Databricks Spark エンジンがランタイムファイルをステージングするディレクトリ。 存在しないディレクトリを指定すると、データ統合サービスは実行時にディレクトリを作成します。 ディレクトリパスを指定しない場合、ランタイムステージングファイルは /<クラスタのステージングディレクトリ>/DATABRICKS に書き込まれます。
詳細プロパティ	Hadoop 環境に固有な詳細プロパティのリスト。 データ統合サービスおよび Databricks 接続で Databricks 環境のランタイムプロパティを設定できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを設定する場合は、Databricks 接続でオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。 1. Databricks 接続の詳細プロパティ 2. データ統合サービスのカスタムプロパティ 注: Informatica は、サードパーティが提供するドキュメント、Informatica が提供するドキュメント、または Informatica のグローバルカスタマサポートに問い合わせることなく、これらのプロパティ値を変更することはお勧めしません。プロパティに関する知識がないまま値を変更すると、パフォーマンスの低下やその他の予期しない結果が発生する可能性があります。

詳細プロパティ

[Spark 設定] セクションの **【詳細プロパティ】** で、次のプロパティを設定します。

infaspark.json.parser.mode

破損した JSON レコードの処理方法をパーサーに指定します。次のいずれかのモードに値を設定できます。

- DROPMALFORMED。パーサーは破損したすべてのレコードを無視します。デフォルトのモードです。
- PERMISSIVE。パーサーは、破損したレコードの標準ではないフィールドを NULL として受け取ります。

- FAILFAST。パーサーが破損したレコードを検出し、Spark アプリケーションがダウンした場合、パーサーは例外を生成します。

`infaspark.json.parser.multiLine`

JSON ファイルの複数行のレコードをパーサーが読み取れるかどうかを指定します。この値は、true または false に設定できます。デフォルトは false です。Spark バージョン 2.2.x 以上を使用する非ネイティブのディストリビューションにのみ適用します。

`infaspark.flatfile.writer.nullValue`

Databricks Spark エンジンがターゲットに書き込むときに、NULL 値は空の文字列（「」）に変換されます。例: 12, AB, "", 23p09udj。

Databricks Spark エンジンは文字列カラムに空の文字列を書き込むことができますが、空の文字列を非文字列カラムに書き込もうとすると、タイプの不一致でマッピングが失敗します。

Databricks Spark エンジンで空の文字列を NULL 値に戻してターゲットに書き込めるようにするには、Databricks Spark 接続でプロパティを設定します。

TRUE に設定します。

`infaspark.pythontx.exec`

Databricks Spark エンジンで Python トランスフォーメーションを実行するために必要です。Databricks クラスタのワーカーノード上にある Python 実行可能バイナリの場所を設定します。

実行時にクラスタをプロビジョニングするときに、Databricks クラウドプロビジョニング構成でこのプロパティを設定します。それ以外の場合は、Databricks 接続で設定します。

例えば、以下を設定します。

```
infaspark.pythontx.exec=/databricks/python3/bin/python3
```

`infaspark.pythontx.executorEnv.PYTHONHOME`

Databricks Spark エンジンで Python トランスフォーメーションを実行するために必要です。Databricks クラスタのワーカーノード上にある Python インストールディレクトリの場所を設定します。

実行時にクラスタをプロビジョニングするときに、Databricks クラウドプロビジョニング構成でこのプロパティを設定します。それ以外の場合は、Databricks 接続で設定します。

例えば、以下を設定します。

```
infaspark.pythontx.executorEnv.PYTHONHOME=/databricks/python3
```

Greenplum 接続プロパティ

Greenplum 接続を使用して、Greenplum データベースに接続します。Greenplum 接続は、リレーショナルタイプの接続です。Greenplum 接続は、Administrator ツールまたは Developer ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Greenplum 接続の作成時には、メタデータおよびデータアクセスの情報を入力します。

以下の表に、Greenplum 接続のプロパティを示します。

プロパティ	説明
名前	Greenplum リレーショナル接続の名前。
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。

メタデータをインポートするには、ユーザー名、パスワード、ドライバ名、接続文字列が必要です。以下の表に、メタデータアクセスのプロパティを示します。

プロパティ	説明
ユーザー名	Greenplum データベースにアクセスする権限を持つユーザー名。
パスワード	Greenplum データベースに接続するためのパスワード。
ドライバ名	Greenplum JDBC ドライバの名前。 例: com.pivotal.jdbc.GreenplumDriver このドライバの詳細については、Greenplum のマニュアルを参照してください。
接続文字列	以下の接続 URL を使用します。 jdbc:pivotal:greenplum://<hostname>:<port>;DatabaseName=<database_name> この接続 URL の詳細については、Greenplum のマニュアルを参照してください。

PowerExchange for Greenplum では、ホスト名、ポート番号、データベース名を使用して制御ファイルを作成し、ロード仕様を Greenplum gpload バルクロードユーティリティに提供します。これは [SSL の有効化] オプションと証明書パスを使用して、SSL を介した Greenplum サーバーへの安全な通信を確立します。

以下の表に、データアクセスの接続プロパティを示します。

プロパティ	説明
ホスト名	Greenplum サーバーのホスト名または IP アドレス。
ポート番号	Greenplum サーバーのポート番号。0 を入力すると、gpload ユーティリティは環境変数 \$PGPORT から読み取ります。デフォルトは 5432 です。
データベース名	データベースの名前。
SSL の有効化	SSL を介した gpload ユーティリティと Greenplum サーバー間の安全な通信を確立するには、このオプションを選択します。
証明書パス	Greenplum サーバーの SSL 証明書が保存されている場所へのパス。 証明書パスに存在している必要があるファイルの詳細については、gpload のマニュアルを参照してください。

Google Analytics 接続のプロパティ

Salesforce Analytics 接続をセットアップする際には、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Google Analytics 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+=[{}] \\:;'"<,>.<?/?
ID	データ統合サービスが接続の識別に使用する文字列。 ID では大文字と小文字が区別されません。ID は 255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。
サービスアカウント ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある client_email 値を指定します。
サービスアカウントキー	サービスアカウントの作成後にダウンロードする JSON ファイル内にある private_key 値を指定します。
APIVersion	【Core Reporting API v3】を選択します。 注:

Google BigQuery 接続のプロパティ

Salesforce Analytics 接続をセットアップする際には、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Google BigQuery 接続のプロパティを示します。

プロパティ	説明
サービスアカウント ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある client_email 値を指定します。
サービスアカウントキー	サービスアカウントの作成後にダウンロードする JSON ファイル内にある private_key 値を指定します。

プロパティ	説明
接続モード	<p>Google BigQuery との間でのデータの読み書きに使用するモード。</p> <p>次のいずれかの接続モードを選択します。</p> <ul style="list-style-type: none"> - 簡易。レコードデータ型フィールド内の各フィールドを、マッピング内の個別のフィールドとしてフラット化します。 - 混合。レコードデータ型のフィールドを含む Google BigQuery テーブル内のすべての最上位のフィールドを表示します。Google BigQuery コネクタは、最上位のレコードデータ型のフィールドを、マッピング内の文字列データ型の単一のフィールドとして表示します。 - 複合。Google BigQuery テーブル内のすべての列を、マッピング内の文字列データ型の単一のフィールドとして表示します。 <p>デフォルトは [簡易] です。</p>
スキーマ定義のファイルパス	<p>で Google BigQuery テーブルのサンプルスキーマと一緒に JSON ファイルを作成する必要があるクライアントマシン上のディレクトリを指定します。JSON ファイル名は、Google BigQuery テーブル名と同じです。</p> <p>または、で Google BigQuery テーブルのサンプルスキーマと一緒に JSON ファイルを作成する必要がある Google Cloud Storage 内のストレージパスを指定します。JSON ファイルは、Google Cloud Storage 内の指定したパスからローカルマシンにダウンロードできます。</p>
プロジェクト ID	<p>サービスアカウントの作成後にダウンロードする JSON ファイル内にある project_id 値を指定します。</p> <p>同じサービスアカウントを使用して複数のプロジェクトを作成した場合、接続先のデータセットが含まれるプロジェクトの ID を入力します。</p>
ストレージパス	<p>このプロパティは、大量のデータを読み書きするときに適用されます。</p> <p>データを一時的に格納するために、がローカルステージファイルを作成する Google Cloud Storage 内のパス。</p> <p>バケット名、またはバケット名とフォルダ名のいずれかを入力できます。</p> <p>例えば、gs://<bucket_name>または gs://<bucket_name>/<folder_name>を入力します。</p>
データセット ID	PowerExchange for Google BigQuery には該当しません。
従来の SQL をカスタムクエリに使用	PowerExchange for Google BigQuery には該当しません。
カスタムクエリのデータセット名	PowerExchange for Google BigQuery には該当しません。
リージョン ID	<p>Google BigQuery データセットが存在するリージョン名。</p> <p>例えば、ラスベガスリージョンにある Google BigQuery データセットに接続するには、[リージョン ID] として us-west4 を指定します。</p> <p>注: [ストレージパス] 接続プロパティで、Google BigQuery のデータセットと同じリージョンにあるバケット名、またはバケット名とフォルダ名を指定していることを確認します。</p> <p>Google BigQuery でサポートされているリージョンの詳細については、次の Google BigQuery のドキュメントを参照してください https://cloud.google.com/bigquery/docs/locations</p>

プロパティ	説明
オプションのプロパティ	<p>カスタムプロパティを介して特定のソースおよびターゲット機能を設定できるかどうかを指定します。</p> <p>以下のいずれかのオプションを選択することができます。</p> <ul style="list-style-type: none"> - なし。カスタムプロパティを設定しない場合に選択します。 - 必須。カスタムプロパティを指定して、ソースおよびターゲットの機能を設定する場合。デフォルトは [なし] です。
オプションのプロパティの指定	<p>特定のソースおよびターゲット機能を有効にするための、カスタムプロパティのカンマ区切りのキーと値のペア。</p> <p>オプションのプロパティで 【必須】 を選択した場合にのみ表示されます。</p> <p>指定できるカスタムプロパティのリストの詳細については、次の Informatica Knowledge Base の記事を参照してください: https://kb.informatica.com/faq/7/Pages/26/632722.aspx</p>

接続モード

Google BigQuery 接続は、次のいずれかの接続モードを使用するように設定できます。

簡易モード

混合モード

混合モードを使用する場合、Google BigQuery コネクタは、レコードデータ型のフィールドを含む Google BigQuery テーブル内のすべての最上位のフィールドを表示します。

複合モード

Google Cloud Spanner 接続のプロパティ

Google Cloud Spanner 接続を作成する際には、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Google Cloud Spanner 接続のプロパティを示します。

プロパティ	説明
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[] \:;'"<, > . ? /</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。</p> <p>ID では大文字と小文字が区別されません。ID は 255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。</p> <p>デフォルト値は接続名です。</p>
説明	<p>オプション。接続の説明。説明は、4,000 文字を超えることはできません。</p>
場所	<p>接続を作成するドメイン。</p>

プロパティ	説明
タイプ	接続タイプ。
プロジェクト ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>project_id</code> 値を指定します。 同じサービスアカウントを使用して複数のプロジェクトを作成した場合、接続先のバケットが含まれるプロジェクトの ID を入力します。
サービスアカウント ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>client_email</code> 値を指定します。
サービスアカウントキー	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>private_key</code> 値を指定します。
インスタンス ID	Google Cloud Spanner 内で作成したインスタンスの名前。

Google Cloud Storage 接続のプロパティ

Google Cloud Storage 接続を作成する際には、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Google Cloud Storage 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意する必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={ }\:;'"<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。 ID では大文字と小文字が区別されません。ID は 255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。
プロジェクト ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>project_id</code> 値を指定します。 同じサービスアカウントを使用して複数のプロジェクトを作成した場合、接続先のバケットが含まれるプロジェクトの ID を入力します。

プロパティ	説明
サービスアカウント ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>client_email</code> 値を指定します。
サービスアカウントキー	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>private_key</code> 値を指定します。

Google PubSub 接続のプロパティ

Google PubSub 接続を作成する際には、接続プロパティを設定する必要があります。

次の表に、Google PubSub 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。~`!\$%^&*()-+= {[}] \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。接続を識別するために使用できる文字列を入力します。説明は、4,000 文字を超えることはできません。
接続タイプ	接続タイプ。[PubSub] 接続タイプを選択します。
クライアントの電子メール	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>client_email</code> 値。
クライアント ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>client_id</code> 値。
プライベートキー ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>private_key_id</code> 値。
秘密鍵	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>private_key</code> 値。
プロジェクト ID	サービスアカウントの作成後にダウンロードする JSON ファイル内にある <code>project_id</code> 値。

Hadoop 接続プロパティ

Hadoop 接続を使用して、Hadoop クラスタで実行するマッピングを設定します。Hadoop 接続は、クラスタ型の接続です。Hadoop 接続は、管理者ツールまたは Developer tool で作成および管理できます。infacmd を使用して Hadoop 接続を作成できます。Hadoop 接続プロパティは、特に明記されている場合を除き、大文字と小文字が区別されます。

Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。

1. infacmd ms runMapping で-cp オプションを指定して設定された、マッピングのカスタムプロパティ
2. Hadoop 環境のマッピングのランタイムプロパティ
3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ
4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス
5. データ統合サービスのカスタムプロパティ

注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策：ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例：beeline -u "jdbc:hive2://<domain host>:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez

Hadoop クラスタプロパティ

データ統合サービスと Hadoop クラスタ間の通信を有効にするには、Hadoop 接続のプロパティを設定します。

以下の表に、Hadoop 接続の一般的な接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。接続を識別するために使用できる文字列を入力します。説明は、4,000 文字を超えることはできません。
クラスタ設定	Hadoop 環境に関連付けられているクラスタ設定の名前。 クラウドプロビジョニング設定を行わない場合に必要です。
クラウドプロビジョニング設定	Amazon AWS や Microsoft Azure などのクラウドプラットフォームに関連付けられているクラウドプロビジョニング設定の名前。 クラスタ設定を行わない場合に必要です。

プロパティ	説明
クラスタ環境変数*	<p>Hadoop クラスタが使用する環境変数。</p> <p>Cloudera CDH 6.x または Cloudera CDP クラスタを使用する場合は、ロケール設定をクラスタ環境変数として設定します。Cloudera Manager で、環境変数を次の YARN プロパティに追加する必要もあります。</p> <p>yarn.nodemanager.env-whitelist</p> <p>例えば、変数 ORACLE_HOME は、Oracle データベースクライアントソフトウェアがインストールされているディレクトリを表します。</p> <p>Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。</p> <ol style="list-style-type: none"> 1. infacmd ms runMapping で-cp オプションを指定して設定された、マッピングのカスタムプロパティ 2. Hadoop 環境のマッピングのランタイムプロパティ 3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ 4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス 5. データ統合サービスのカスタムプロパティ <p>注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策: ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例: beeline -u "jdbc:hive2://<domain host>:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez</p>
クラスタのライブラリパス*	<p>クラスタ上の共有ライブラリのパス。</p> <p>\$DEFAULT_CLUSTER_LIBRARY_PATH 変数には、デフォルトディレクトリの一覧が含まれています。</p>
クラスタのクラスパス*	<p>Hadoop jar ファイルおよび必要なライブラリにアクセスするためのクラスパス。</p> <p>\$DEFAULT_CLUSTER_CLASSPATH 変数には、デフォルトの jar ファイルおよびライブラリへのパスの一覧が含まれています。</p> <p>Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。</p> <ol style="list-style-type: none"> 1. infacmd ms runMapping で-cp オプションを指定して設定された、マッピングのカスタムプロパティ 2. Hadoop 環境のマッピングのランタイムプロパティ 3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ 4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス 5. データ統合サービスのカスタムプロパティ <p>注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策: ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例: beeline -u "jdbc:hive2://<domain host>:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez</p>

プロパティ	説明
クラスタの実行可能ファイルパス*	クラスタ上の実行可能ファイルのパス。 \$DEFAULT_CLUSTER_EXEC_PATH 変数には、デフォルトの実行可能ファイルへのパスの一覧が含まれています。
* Informatica は、サードパーティが提供するドキュメント、Informatica が提供するドキュメント、または Informatica のグローバルカスタマサポートに問い合わせることなく、これらのプロパティ値を変更することはお勧めしません。プロパティに関する知識がないまま値を変更すると、パフォーマンスの低下やその他の予期しない結果が発生する可能性があります。	

共通プロパティ

以下の表に、Hadoop 接続に対して設定する共通の接続プロパティを示します。

プロパティ	説明
偽装ユーザー名	Hadoop クラスターで Kerberos 認証を使用する場合に必要です。Hadoop 偽装ユーザー Hadoop 環境でマッピングを実行するためにデータ統合サービスが偽装するユーザー名。 Data Engineering Integration は、すべての Hadoop ディストリビューションのオペレーティングシステムプロファイルをサポートします。Hadoop ランタイム環境では、データ統合サービスが Hadoop クラスタに処理をプッシュし、ランタイムエンジンがオペレーティングシステムプロファイルで指定されている Hadoop 偽装プロパティを使用してマッピングを実行します。
一時テーブル圧縮コーデック	圧縮コーデッククラス名の Hadoop 圧縮ライブラリ。 注: Spark エンジンには、一時テーブルの圧縮設定をサポートしていません。Spark エンジンでマッピングを実行すると、Spark エンジンでは一時テーブルを非圧縮ファイル形式で格納します。
コーデッククラス名	データ圧縮を有効にし、一時ステージングテーブルのパフォーマンスを改善するコーデッククラス名。
Hive ステージングデータベース名	Hive ステージングテーブルの名前空間。指定されたデータベース名を持たないテーブルには、名前として default を使用してください。 名前空間を設定しない場合、データ統合サービスは Hive ターゲット接続の Hive データベース名を使用してステージングテーブルを作成します。 ネイティブ環境でマッピングを実行して Hive にデータを書き込む場合、Hive 接続の Hive ステージングデータベース名を設定する必要があります。データ統合サービスは、Hadoop 接続で設定された値を使用します。
環境 SQL	Hadoop 環境を設定する SQL コマンド。データ統合サービスは、HiveServer2 ジョブによって生成された各 Hive スクリプトの開始時に環境 SQL を実行します。 環境 SQL の使用には、以下のルールおよびガイドラインが適用されます。 - 環境 SQL は、PreSQL コマンド内またはカスタムクエリ内で使用する Hadoop パラメータまたは Hive パラメータを定義するために使用できます。 - 環境 SQL プロパティに複数の値を使用する場合は、値の間にスペースを含めないようにしてください。

プロパティ	説明
エンジン タイプ	<p>データ統合サービスは HiveServer2 を使用して、Spark エンジンで HiveServer2 タスクを実行することで、いくつかのジョブの一部を処理します。Administrator ツールを使用してクラスタ設定をインポートした場合は、接続を作成することを選択できます。エンジンタイププロパティは、ディストリビューションに基づいてデフォルトで設定されます。</p> <p>接続を手動で作成する場合は、エンジンタイプを設定する必要があります。</p> <p>次の Hadoop ディストリビューションに基づいてエンジンタイプを指定できます。</p> <ul style="list-style-type: none"> - Amazon EMR。 Tez - Azure HDI。 Tez - Cloudera CDH。 MRv2 - Cloudera CDP。 Tez - Dataproc。 MRv2 - Hortonworks HDP。 Tez - MapR。 MRv2
詳細プロ パティ	<p>Hadoop 環境に固有な詳細プロパティのリスト。プロパティは、Blaze エンジン、エンジン、および Hive エンジンに共通です。詳細プロパティには、デフォルトプロパティのリストが含まれます。</p> <p>Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。</p> <ol style="list-style-type: none"> 1. infacmd ms runMapping で -cp オプションを指定して設定された、マッピングのカスタムプロパティ 2. Hadoop 環境のマッピングのランタイムプロパティ 3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ 4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス 5. データ統合サービスのカスタムプロパティ <p>注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策: ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例: <code>beeline -u "jdbc:hive2://<domain host>:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez</code></p> <p>注: Informatica は、サードパーティが提供するドキュメント、Informatica が提供するドキュメント、または Informatica のグローバルカスタマサポートに問い合わせることなく、これらのプロパティ値を変更することはお勧めしません。プロパティに関する知識がないまま値を変更すると、パフォーマンスの低下やその他の予期しない結果が発生する可能性があります。</p>

拒否ディレクトリのプロパティ

以下の表に、Hadoop 拒否ディレクトリに設定する接続プロパティを示します。

プロパティ	説明
Hadoop への拒否ファイルの書き込み	Blaze エンジンを使用してマッピングを実行する場合は、このチェックボックスを選択して、拒否ファイルを移動する場所を指定します。選択すると、拒否ファイルはプロパティにリストされている HDFS の場所（拒否ファイルディレクトリ）に移動されます。 デフォルトでは、データ統合サービスは、RejectDir システムパラメータに基づいて拒否ファイルを格納します。
拒否ファイルディレクトリ	マッピングを実行するときの HDFS 上の Hadoop マッピングファイルのディレクトリ。

Blaze 設定

以下の表に、Blaze エンジンに対して設定する接続プロパティを示します。

プロパティ	説明
Blaze ステージングディレクトリ	Blaze エンジンが一時ファイルを保存するために使用するディレクトリの HDFS ファイルパス。ディレクトリが存在することを確認してください。YARN ユーザー、Blaze エンジンユーザー、およびマッピング偽装ユーザーには、このディレクトリへの書き込み権限が必要です。デフォルトは/blaze/workdir です。このプロパティをオフにすると、ステージングファイルは Hadoop ステージングディレクトリ/tmp/blaze_<user name>に書き込まれます。
Blaze ユーザー名	Blaze サービスと Blaze サービスログの所有者。 Hadoop クラスターで Kerberos 認証を使用する場合、既定のユーザーはデータ統合サービスの SPN ユーザーです。Hadoop クラスターで Kerberos 認証を使用せず、Blaze ユーザーが構成されていない場合、既定のユーザーはデータ統合サービスユーザーです。
最小ポート	Blaze エンジンのポート番号範囲の最小値。デフォルトは 12300 です。
最大ポート	Blaze エンジンのポート番号範囲の最大値。デフォルトは 12600 です。
YARN キュー名	クラスタ上の利用可能なリソースを指定する Blaze エンジンが使用する YARN スケジューラのキュー名。 注: クラスタで YARN プリエンプションが有効な場合、Blaze エンジンに関連付けられているキューでプリエンプションが無効になっていることを Hadoop 管理者に確認してください。
Blaze ジョブ監視アドレス	Blaze ジョブ監視のホスト名およびポート番号。 次の形式を使用します。 <hostname>:<port> ここで - <hostname>は、Blaze ジョブ監視サーバーのホスト名または IP アドレスです。 - <port>は、Blaze ジョブ監視がリモートプロシージャコール (RPC) をリスンするポートです。 例えば、次のように入力します。myhostname:9080

プロパティ	説明
Blaze YARN ノードラベル	<p>Blaze エンジンが実行される Hadoop クラスタ上のノードを決定するノードラベル。ノードラベルを指定しない場合、Blaze エンジンはデフォルトのパーティションのノードで実行されます。</p> <p>Hadoop クラスタがノードラベルの論理演算子をサポートしている場合は、ノードラベルの一覧を指定できます。ノードラベルを一覧表示するには、演算子&& (AND)、 (OR)、および! (NOT) を使用します。</p> <p>注: Cloudera CDH クラスタではノードラベルを使用できません。</p>
詳細プロパティ	<p>Blaze エンジンに固有な詳細プロパティのリスト。詳細プロパティには、デフォルトプロパティのリストが含まれます。</p> <p>Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。</p> <ol style="list-style-type: none"> 1. infacmd ms runMapping で-cp オプションを指定して設定された、マッピングのカスタムプロパティ 2. Hadoop 環境のマッピングのランタイムプロパティ 3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ 4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス 5. データ統合サービスのカスタムプロパティ <p>注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策: ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例: beeline -u "jdbc:hive2://<domain>host:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez</p> <p>注: Informatica は、サードパーティが提供するドキュメント、Informatica が提供するドキュメント、または Informatica のグローバルカスタマサポートに問い合わせることなく、これらのプロパティ値を変更することはお勧めしません。プロパティに関する知識がないまま値を変更すると、パフォーマンスの低下やその他の予期しない結果が発生する可能性があります。</p>

Spark 設定

以下の表に、Spark エンジンに対して設定する接続プロパティを示します。

プロパティ	説明
Spark ステージングディレクトリ	<p>Spark エンジンがジョブ実行用の一時ファイルを保存するために使用するディレクトリの HDFS ファイルパス。YARN ユーザー、データ統合サービスユーザー、およびマッピング偽装ユーザーがこのディレクトリに対する書き込みアクセス許可を持っている必要があります。</p>
Spark イベントログディレクトリ	<p>オプション。Spark エンジンがイベントをログ記録するために使用するディレクトリの HDFS ファイルパス。</p>

プロパティ	説明
YARN キュー名	クラスタで使用可能なリソースを指定する Spark エンジンによって使用される YARN スケジューラキュー名。この名前は、大文字と小文字が区別されます。
詳細プロパティ	<p>Spark エンジンに固有な詳細プロパティのリスト。詳細プロパティには、デフォルトプロパティのリストが含まれます。</p> <p>Hadoop 環境のランタイムプロパティは、データ統合サービス、Hadoop 接続、およびマッピングで構成できます。下位レベルで値を設定することにより、上位レベルで構成されたプロパティをオーバーライドできます。例えば、データ統合サービスのカスタムプロパティでプロパティを構成する場合は、Hadoop 接続またはマッピングでオーバーライドできます。データ統合サービスは、次の優先順位に基づいてプロパティのオーバーライドを処理します。</p> <ol style="list-style-type: none"> 1. infacmd ms runMapping で-cp オプションを指定して設定された、マッピングのカスタムプロパティ 2. Hadoop 環境のマッピングのランタイムプロパティ 3. ランタイムエンジン用の Hadoop 接続の詳細プロパティ 4. Hadoop 接続の詳細な全般プロパティ、環境変数、およびクラスパス 5. データ統合サービスのカスタムプロパティ <p>注: マッピングで Hive Server2 を使用してジョブまたはジョブの一部を実行する場合、preSQL または post-SQL クエリまたは SQL オーバーライドステートメントでクラスタレベルで設定されているプロパティをオーバーライドすることはできません。回避策：ドメインのクラスタ設定を使用してクラスタのプロパティをオーバーライドする代わりに、オーバーライド設定を JDBC URL に渡します。例：beeline -u "jdbc:hive2://<domain host>:<port_number>/tpch_text_100" --hiveconf hive.execution.engine=tez</p> <p>注: Informatica は、サードパーティが提供するドキュメント、Informatica が提供するドキュメント、または Informatica のグローバルカスタマサポートに問い合わせることなく、これらのプロパティ値を変更することはお勧めしません。プロパティに関する知識がないまま値を変更すると、パフォーマンスの低下やその他の予期しない結果が発生する可能性があります。</p>

HBase 接続プロパティ

HBase 接続を使用して HBase にアクセスします。HBase 接続は NoSQL 接続です。HBase 接続は、Administrator ツールまたは Developer ツールで作成および管理できます。Hbase 接続プロパティは、特に明記されている場合を除き、大文字と小文字が区別されます。

以下の表に、HBase 接続のプロパティを示します。

プロパティ	説明
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。</p> <p>~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。</p>
説明	<p>接続の説明。説明は、4,000 文字を超えることはできません。</p>

プロパティ	説明
場所	接続を作成するドメイン。
タイプ	接続タイプ。HBase を選択します。
データベースタイプ	接続先のデータベースの種類。 HBase テーブルの接続を作成するには、[HBase] を選択します。

HDFS 接続プロパティ

Hadoop ファイルシステム (HDFS) 接続を使用して、Hadoop クラスタ内のデータにアクセスします。HDFS 接続は、ファイルシステムタイプの接続です。HDFS 接続は、Administrator ツール、Analyst ツール、または Developer ツールで作成および管理できます。HDFS 接続プロパティは、特に明記されている場合を除き、大文字と小文字が区別されます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、HDFS 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成するドメイン。Analyst ツールでは使用できません。
タイプ	接続タイプ。デフォルトは Hadoop ファイルシステムです。
ユーザー名	HDFS にアクセスするためのユーザー名。
NameNode の URI	ストレージシステムにアクセスするための URI。fs.defaultFS の値は、クラスタ設定の core-site.xml 設定セットで確認できます。 クラスタ設定をインポートするときに接続を作成する際は、NameNode URI プロパティがデフォルトで設定され、クラスタ設定を更新するたびに更新されます。 Cloudera CDP Public Cloud コンピューティングクラスタを使用しており、HDFS が Cloudera Data Lake クラスタ上にある場合は、Hadoop 接続の Spark プロパティ内のプロパティ <code>spark.yarn.access.hadoopFileSystems</code> を、ここで設定した値と同じ値に設定します。

複数のストレージタイプへのアクセス

さまざまな種類のストレージに接続するには、接続パラメータの NameNode URI プロパティを使用します。次の表は、ストレージの種類とストレージの種類別の NameNode URI 形式を示しています。

記憶領域	NameNode URI 形式
HDFS	<code>hdfs://<namenode>:<port></code> ここで、 <ul style="list-style-type: none">- <namenode>は、NameNode のホスト名または IP アドレスです。- <port>は、NameNode がリモートプロシージャコール (RPC) をリスンするポートです。 NameNode 高可用性の場合の <code>hdfs://<nameservice></code> 。
MapR-FS	<code>maprfs:///</code>
HDInsight 内の WASB	<code>wasb://<container_name>@<account_name>.blob.core.windows.net/<path></code> ここで、 <ul style="list-style-type: none">- <container_name>は、特定の Azure Storage Blob コンテナを示します。 注: <container_name>は省略可能です。- <account_name>は、Azure Storage Blob オブジェクトを示します。 例: <code>wasb://infabdmoffering1storage.blob.core.windows.net/infabdmoffering1cluster/mr-history</code>
HDInsight 内の ADLS	<code>adl://home</code>

Azure HDInsight クラスタからクラスタ構成を作成する場合、クラスタ構成ではプライマリストレージとして ADLS または WASB のいずれかを使用します。ADLS または WASB をセカンダリストレージとして指定したクラスタ設定は作成できません。HDFS 接続の NameNode URI プロパティを編集して、ローカルの HDFS の場所に接続することができます。

MapR-DB の HBase 接続のプロパティ

HBase 接続を使用して、MapR テーブルに接続します。HBase 接続は NoSQL 接続です。HBase 接続は、Administrator ツールまたは Developer ツールで作成および管理できます。Hbase 接続プロパティは、特に明記されている場合を除き、大文字と小文字が区別されます。

次の表は、MapR-DB の HBase 接続プロパティを示しています。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[HBase] を選択します。
データベースタイプ	接続先のデータベースの種類。 MapR-DB テーブルの接続を作成するには、MapR-DB を選択します。
クラスタ設定	Hadoop 環境に関連付けられているクラスタ設定の名前。
MapR-DB のデータベースパス	接続先の MapR-DB テーブルが格納されているデータベースのパス。有効な MapR クラスタパスを入力してください。 MapR-DB の HBase データオブジェクトを作成する場合は、 [データベースパス] フィールドで指定した MapR-DB パスに存在するテーブルだけを参照できます。指定したパスのサブディレクトリで利用できるテーブルにはアクセスできません。 例えば、パスを /user/customers/ として指定すると、customers ディレクトリ内のテーブルにアクセスできます。ただし、customers ディレクトリに regions という名前のサブディレクトリが含まれている場合、次のディレクトリ内にあるテーブルにはアクセスできません。 /user/customers/regions

Hive 接続のプロパティ

Hive 接続を利用して、Hive データにアクセスします。Hive 接続は、データベースタイプの接続です。Hive 接続は、Administrator ツール、Analyst ツール、または Developer tool で作成および管理できます。Hive 接続プロパティは、特に明記されている場合を除き、大文字と小文字が区別されます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Hive 接続のプロパティを示します。

プロパティ	説明
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。</p> <p>~ ` ! \$ % ^ & * () - + = { []] \ : ; " ' < , > . ? /</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。</p>
説明	<p>接続の説明。説明は、4000 文字を超えることはできません。</p>
場所	<p>接続を作成するドメイン。Analyst ツールでは使用できません。</p>
タイプ	<p>接続タイプ。Hive を選択します。</p>
LDAP ユーザー名	<p>Hadoop クラスタでマッピングを実行するためにデータ統合サービスが偽装するユーザーの LDAP ユーザー名。ユーザー名は、ネイティブ環境の [メタデータ接続文字列] または [データアクセス接続文字列] に指定する JDBC 接続文字列によって決まります。</p> <p>Hadoop クラスタが Kerberos 認証を使用する場合、JDBC 接続文字列のプリンシパル名とこのユーザー名が同じである必要があります。Hadoop クラスタが Kerberos 認証を使用しない場合、ユーザー名は JDBC ドライバの動作によって異なります。Hive JDBC ドライバを使用すると、さまざまな方法でユーザー名を指定することができ、そのユーザー名を JDBC URL の一部にすることができます。</p> <p>Hadoop クラスタが Kerberos 認証を使用しない場合、ユーザー名は JDBC ドライバの動作によって決まります。</p> <p>ユーザー名を指定しないと、Hadoop クラスタは次の基準に基づいてジョブを認証します。</p> <ul style="list-style-type: none"> - Hadoop クラスタが Kerberos 認証を使用しない。データ統合サービスを実行するマシンのオペレーティングシステムのプロファイルユーザー名に基づいてジョブを認証します。 - Hadoop クラスタが Kerberos 認証を使用する。ジョブの認証は、データ統合サービスの SPN に基づいて行います。LDAP ユーザー名は無視されます。
パスワード	<p>LDAP ユーザー名のパスワード。</p>

プロパティ	説明
環境 SQL	<p>Hadoop 環境を設定する SQL コマンド。ネイティブ環境タイプでは、データ統合サービスは Hive メタストアへの接続を作成するたびに環境 SQL を実行します。Hadoop クラスタでのプロファイルの実行に Hive 接続を使用すると、データ統合サービスは各 Hive セッションの最初に環境 SQL を実行します。</p> <p>どちらの接続モードでも、環境 SQL の使用には以下のルールとガイドラインが適用されます。</p> <ul style="list-style-type: none"> - 環境 SQL は、Hive クエリを指定するために使用します。 - 環境 SQL を使用して Hive ユーザー定義関数のクラスパスを設定し、続いて環境 SQL または PreSQL を使用して Hive ユーザー定義関数を指定します。データオブジェクトプロパティで PreSQL を使用してクラスパスを指定することはできません。Hive ユーザー定義関数を使用する場合は、jar ファイルを次のディレクトリにコピーする必要があります。 <Informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>/extras/hive-auxjars - 環境 SQL は、PreSQL コマンド内またはカスタムクエリ内で使用する Hadoop パラメータまたは Hive パラメータを定義するために使用できません。 - 環境 SQL プロパティに複数の値を使用する場合は、値の間にスペースを含めないようにしてください。
SQL 識別子文字	<p>特殊文字と SQL の予約済み SQL キーワード（WHERE など）の識別に使用される文字のタイプ。データ統合サービスは特殊文字と予約済み SQL キーワードを選択した文字で囲みます。データ統合サービスはこの文字を【大文字小文字が混在した識別子をサポート】プロパティにも使用します。</p>

ソースまたはターゲットとして Hive にアクセスするためのプロパティ

以下の表に、ソースまたはターゲットとして Hive にアクセスするために設定する接続プロパティを示します。

プロパティ	説明
JDBC ドライバクラス名	<p>Hive JDBC ドライバクラスの名前。このオプションを空白のままにすると、ディストリビューションに同梱されているデフォルトの Apache Hive JDBC ドライバが使用されます。デフォルトの Apache Hive JDBC ドライバが要件に合わない場合は、ドライバクラス名を指定することで、サードパーティの Hive JDBC ドライバを使用して Apache Hive JDBC ドライバをオーバーライドできます。</p>
メタデータ接続文字列	<p>Hadoop サーバーからメタデータにアクセスするために使用される JDBC 接続 URI。 PowerExchange for Hive を使用して、HiveServer サービスまたは HiveServer2 サービスと通信できます。HiveServer に接続するには、次の形式で接続文字列を指定します。</p> <pre>jdbc:hive2://<hostname>:<port>/<db></pre> <p>ここで</p> <ul style="list-style-type: none"> - <hostname>は、HiveServer2 が実行されているマシンの名前または IP アドレスです。 - <port>は、HiveServer2 がリスンするポート番号です。 - <db>は、接続先のデータベース名です。データベース名を指定しないと、データ統合サービスはデフォルトのデータベースの詳細を使用します。 <p>HiveServer 2 に接続するには、その特定の Hadoop ディストリビューション用に Apache Hive が実装する接続文字列形式を使用します。Apache Hive 接続文字列形式の詳細については、Apache Hive のドキュメントを参照してください。</p> <p>ユーザーの偽装については、hive.server2.proxy.user=<xyz> を JDBC 接続 URI に追加する必要があります。ユーザーの偽装を設定しない場合は、現在のユーザーの資格情報を使用して HiveServer2 に接続します。</p> <p>Hadoop クラスタが SSL または TLS 認証を使用する場合は、ssl=true を JDBC 接続 URI に追加する必要があります。例: jdbc:hive2://<hostname>:<port>/<db>;ssl=true</p> <p>SSL または TLS 認証に自己署名証明書を使用する場合は、クライアントコンピュータとデータ統合サービスコンピュータで証明書ファイルが使用可能であることを確認してください。詳細については、『<i>Data Engineering Integration ガイド</i>』を参照してください。</p>
Hive JDBC サーバーをバイパス	<p>JDBC ドライバモード。埋め込み JDBC ドライバモードを使用するには、チェックボックスを選択します。</p> <p>JDBC 埋め込みモードを使用するには、以下のタスクを実行します。</p> <ul style="list-style-type: none"> - Hive クライアントと Informatica サービスが同一のマシンにインストールされていることを確認します。 - Hadoop クラスタでマッピングを実行するように Hive 接続プロパティを設定します。 <p>非埋め込みモードを選択する場合は、データアクセス接続文字列を設定する必要があります。</p> <p>JDBC 埋め込みモードを使用することをお勧めします。</p>
厳密に定義された認証	<p>Hive ソースで厳密に定義された認証に従うオプションを選択すると、マッピングでは次のことに従います。</p> <ul style="list-style-type: none"> - 行レベルとカラムレベルの制限。Sentry または Ranger のセキュリティモードが有効になっている Hadoop クラスタに適用されます。 - データマスキングルール。Dynamic Data Masking によって機密データを含むカラムに設定されたマスキングルールに適用されます。 <p>このオプションを選択しない場合、Blaze および Spark エンジンでは制限とマスキングルールを無視し、制限されたデータまたは機密データが結果に含まれます。</p>

プロパティ	説明
データアクセス接続文字列	<p>Hadoop データストアからデータにアクセスするための接続文字列。HiveServer に接続するには、次の形式で非埋め込み JDBC モードの接続文字列を指定します。</p> <pre>jdbc:hive2://<hostname>:<port>/<db></pre> <p>ここで</p> <ul style="list-style-type: none"> - <hostname>は、HiveServer2 が実行されているマシンの名前または IP アドレスです。 - <port>は、HiveServer2 がリスンするポート番号です。 - <db>は、接続先のデータベースです。データベース名を指定しないと、データ統合サービスはデフォルトのデータベースの詳細を使用します。 <p>HiveServer 2 に接続するには、特定の Hadoop ディストリビューション用に Apache Hive が実装する接続文字列形式を使用します。Apache Hive 接続文字列形式の詳細については、Apache Hive のドキュメントを参照してください。</p> <p>ユーザーの偽装については、hive.server2.proxy.user=<xyz>を JDBC 接続 URI に追加する必要があります。ユーザーの偽装を設定しない場合は、現在のユーザーの資格情報を使用して HiveServer2 に接続します。</p> <p>Hadoop クラスタが SSL または TLS 認証を使用する場合は、ssl=true を JDBC 接続 URI に追加する必要があります。例: jdbc:hive2://<hostname>:<port>/<db>;ssl=true</p> <p>SSL または TLS 認証に自己署名証明書を使用する場合は、クライアントコンピュータとデータ統合サービスコンピュータで証明書ファイルが使用可能であることを確認してください。詳細については、『<i>Data Engineering Integration ガイド</i>』を参照してください。</p>
HDFS 上の Hive ステージングディレクトリ	<p>Hive ステージングテーブル用の HDFS ディレクトリ。Hadoop 偽装ユーザーとマッピング偽装ユーザーに実行権限を付与する必要があります。</p> <p>このオプションは適用可能で、ネイティブ環境の Hive ターゲットにデータを書き込むときに必要です。</p>
Hive ステージングデータベース名	<p>Hive ステージングテーブルの名前空間。</p> <p>Hive ステージングデータベース名は、データアクセス接続文字列から自動的に更新されます。デフォルト名をオーバーライドする場合は、Hive 接続で Hive ステージングデータベース名を設定する必要があります。</p> <p>このオプションは、Hive ターゲットにデータを書き込むためにネイティブ環境でマッピングを実行する際に適用できます。</p> <p>Blaze または Spark エンジンでマッピングを実行する場合、Hive 接続で Hive ステージングデータベース名を設定する必要はありません。データ統合サービスは、Hadoop 接続で設定された値を使用します。</p>

HTTP 接続のプロパティ

REST Web サービスコンシューマトランスフォーメーションを Web サービスに接続するには、HTTP 接続を使用します。HTTP 接続は、Web タイプの接続です。HTTP 接続は Developer ツールで作成します。HTTP 接続は、Administrator ツールまたは Developer ツールで管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、HTTP 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
ユーザー名	Web サービスに接続するユーザー名。HTTP 認証または WS-Security を有効にした場合は、ユーザー名を入力します。 Web サービスコンシューマトランスフォーメーションに WS-Security ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して動的ユーザー名を受け取ります。データ統合サービスは、接続で定義されているユーザー名をオーバーライドします。
パスワード	ユーザー名に対するパスワード。HTTP 認証または WS-Security を有効にした場合は、パスワードを入力します。 Web サービスコンシューマトランスフォーメーションに WS-Security ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して動的パスワードを受け取ります。データ統合サービスは、接続で定義されているパスワードをオーバーライドします。
エンドポイント URL	アクセス先の Web サービスの URL。データ統合サービスは、WSDL ファイルで定義されている URL をオーバーライドします。 Web サービスコンシューマトランスフォーメーションにエンドポイント URL ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して URL を動的に受け取ります。データ統合サービスは、接続で定義されている URL をオーバーライドします。
タイムアウト	データ統合サービスが、接続を閉じる前に、Web サービスプロバイダからの応答を待機する秒数。1~10000 秒の間でタイムアウト値を指定します。
HTTP 認証タイプ	HTTP 経由のユーザー認証のタイプ。次のいずれかの値を選択します。 <ul style="list-style-type: none"> - なし: 認証しません。 - 自動: データ統合サービスが Web サービスプロバイダの認証タイプを選択します。 - 基本: Web サービスプロバイダのドメインに対するユーザー名およびパスワードの入力が要求されます。データ統合サービスは、認証のためにユーザー名とパスワードを Web サービスプロバイダに送ります。 - ダイジェスト: Web サービスプロバイダのドメインに対するユーザー名およびパスワードの入力が要求されます。データ統合サービスは、ユーザー名とパスワードから暗号化されたメッセージダイジェストを生成し、それを Web サービスプロバイダに送ります。プロバイダはユーザー名とパスワードの一時的な値を生成し、それをドメインコントローラの Active Directory に格納します。その値とメッセージダイジェストが比較されます。これらが一致する場合、Web サービスプロバイダはユーザを認証します。 - NTLM: ドメイン名、サーバー名、またはデフォルトのユーザー名とパスワードの入力が要求されます。Web サービスプロバイダは、接続先のドメインに基づいてユーザを認証します。Windows Domain Controller からユーザー名とパスワードが取得され、ユーザが入力したユーザー名とパスワードと比較されます。これらが一致する場合、Web サービスプロバイダはユーザを認証します。NTLM 認証では、暗号化されたパスワードがドメインコントローラのアクティブディレクトリに保存されることはありません。

プロパティ	説明
トラストサーティフィケーションファイル	Web サービスの SSL 証明書の認証時にデータ統合サービスが使用するトラストサーティフィケーションのバンドルを含むファイル。ファイル名と完全ディレクトリパスを入力します。 デフォルトは、<Informatica installation directory>/services/shared/bin/ca-bundle.crt です。
クライアント証明書ファイル名	クライアントの認証時に Web サービスで使用されるクライアント証明書。Web サービスがデータ統合サービスを認証する必要がある場合にクライアントサーティフィケーションファイルを指定します。
クライアント証明書のパスワード	クライアントサーティフィケーションのパスワード。Web サービスがデータ統合サービスを認証する必要がある場合にクライアント証明書のパスワードを指定します。
クライアント証明書のタイプ	クライアントサーティフィケーションファイルの形式。次のいずれかの値を選択します。 - PEM: 「.pem」 拡張子のファイルです。 - DER: 「.cer」 または 「.der」 拡張子のファイルです。 Web サービスがデータ統合サービスを認証する必要がある場合にクライアント証明書のタイプを指定します。
プライベートキーファイル名	クライアントサーティフィケーションのプライベートキーファイル。Web サービスがデータ統合サービスを認証する必要がある場合にプライベートキーファイルを指定します。
プライベートキーのパスワード	クライアントサーティフィケーションのプライベートキーのパスワード。Web サービスがデータ統合サービスを認証する必要がある場合にプライベートキーのパスワードを指定します。
プライベートキーのタイプ	プライベートキーのタイプ。サポートされているタイプは PEM です。

IBM DB2 接続のプロパティ

IBM DB2 にアクセスするには、IBM DB2 接続を使用します。IBM DB2 接続は、リレーショナルデータベース接続です。IBM DB2 接続は、Administrator ツール、Developer ツール、または Analyst ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、DB2 接続のプロパティを示します。

プロパティ	説明
データベースタイプ	データベースのタイプ。
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /

プロパティ	説明
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
ユーザー名	データベースユーザー名。
パスワード	データベースユーザー名のパスワード。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。接続のパススルーセキュリティを有効にすると、ドメインは、接続オブジェクトに定義されている資格情報ではなく、クライアントのユーザー名とパスワードを使用して対応するデータベースにログインします。
データアクセス接続文字列	データベースのメタデータへのアクセスに使用する DB2 接続 URL。 dbname dbname は、DB2 クライアントで設定されているエイリアスです。
メタデータアクセスプロパティ: 接続文字列	<p>次のメタデータ接続文字列 URL を使用します。</p> <pre>jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name></pre> <p>テーブルをインポートすると、デフォルトでは、デフォルトスキーマ名の下にすべてのテーブルが表示されます。デフォルトスキーマではなく特定のスキーマの下にテーブルを表示するには、テーブルをインポートするスキーマ名を指定します。スキーマ名を指定するには、URL に ischema_name パラメータを追加します。例えば、特定のスキーマからテーブルをインポートするには、次の構文を使用します。</p> <pre>jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name>;ischema_name=<schema_name></pre> <p>複数のスキーマからテーブルを検索してインポートするには、ischema_name パラメータに複数のスキーマ名を指定します。スキーマ名では大文字小文字が区別されます。複数のスキーマ名を指定する場合、特殊文字は使用できません。複数のスキーマ名を区切るには、パイプ () 文字を使用します。例えば、3 つのスキーマからテーブルを検索してインポートするには、次の構文を使用します。</p> <pre>jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name>;ischema_name=<schema_name1> <schema_name2> <schema_name3></pre> <p>複数のスキーマ名を指定する場合、指定したスキーマ名の下にテーブルを表示するには 【デフォルトのスキーマだけを表示する】 オプションの選択を解除する必要があります。</p>

プロパティ	説明
AdvancedJDBCSecurityOptions	<p>セキュアデータベースへのメタデータアクセスのためのデータベースパラメータ。Informatica は、AdvancedJDBCSecurityOptions フィールドの値を機密データとして扱い、パラメータ文字列を暗号化して格納します。</p> <p>セキュアデータベースに接続するには、以下のパラメータを含めます。</p> <ul style="list-style-type: none"> - EncryptionMethod。必須。ネットワーク上で送信される際にデータが暗号化されるかどうかを示します。このパラメータは SSL に設定する必要があります。 - ValidateServerCertificate。オプション。データベースサーバーによって送信される証明書を Informatica が検証するかどうかを示します。 このパラメータを True に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証します。HostNameInCertificate パラメータを指定すると、Informatica は証明書内のホスト名も検証します。 このパラメータを false に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証しません。指定するトラストストア情報がすべて無視されます。 - HostNameInCertificate。オプション。セキュアデータベースをホストするマシンのホスト名。ホスト名を指定すると、Informatica は接続文字列に含められたそのホスト名を SSL 証明書内のホスト名と照らして検証します。 - cryptoProtocolVersion。オプション。IBM DB2 インスタンスに対して TLS を有効にする場合は、次のように cryptoProtocolVersion パラメータを設定します。 cryptoProtocolVersion=TLSv<version number> 例: cryptoProtocolVersion=TLSv1.2 注: バージョン番号はサーバーに設定した TLS バージョンと同じにする必要があります。 - TrustStore。必須。トラストストアファイルのパスおよびファイル名です。 注: SSL または TLS を設定してファイル名のみを指定する場合は、トラストストアファイルを Informatica のインストールディレクトリにコピーする必要があります。接続をテストしメタデータをインポートするには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica client installation directory>/clients/DeveloperClient マッピングを実行するには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica server installation directory>/tomcat/bin - TrustStorePassword。必須。セキュアデータベースに対するトラストストアファイルのパスワード。 注: セキュア JDBC パラメータは接続文字列に自動的に追加されます。セキュア JDBC パラメータを接続文字列に直接指定する場合は、AdvancedJDBCSecurityOptions フィールドにはパラメータを入力しないでください。
データアクセスプロパティ: 接続文字列	<p>データベースのデータへのアクセスに使用する接続文字列。</p> <p>IBM DB2 の場合、これは<database name>です</p>
コードページ	<p>ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。</p>
環境 SQL	<p>データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。</p>

プロパティ	説明
トランザクション SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、トランザクション環境 SQL を各トランザクションの開始時に実行します。
リトライ期限	このプロパティは、将来使用するために予約されています。
テーブルスペース	データベースのテーブルスペース名。
SQL 識別子文字	データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。 データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。 データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。
大文字と小文字が混在する識別子をサポート	データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を [SQL 識別子文字] プロパティに対して選択された文字で囲みます。 [SQL 識別子文字] プロパティが [なし] に設定されている場合、 [大文字と小文字が混在する識別子をサポート] プロパティは無効になっています。
ODBC プロバイダ	ODBC。ODBC を接続するデータベースのタイプ。プッシュダウンの最適化用に、データベースタイプを指定して、データ統合サービスでネイティブデータベース SQL を生成できるようにします。次のオプションがあります。 - その他 - Sybase - Microsoft_SQL_Server デフォルトは [その他] です。

IBM DB2 for i5/OS 接続のプロパティ

IBM DB2 for i5/OS 接続を使用し、IBM DB2 for i5/OS 内のテーブルにアクセスします。IBM DB2 for i5/OS 接続はリレーショナルデータベース接続です。IBM DB2 for i5/OS 接続は、Administrator ツールまたは Developer ツールで管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、DB2 for i5/OS 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、255 文字を超えることはできません。
接続タイプ	接続タイプ (DB2I)。
ユーザー名	データベースユーザー名。
パスワード	指定したユーザー名のパスワードまたは有効な PowerExchange のパスフレーズ。 PowerExchange のパスフレーズは 9～31 文字の範囲で作成します。使用できる文字は次のとおりです。 - 英字（大文字と小文字） - 数字（0～9） - スペース - 次に示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? 注: 先頭に書かれている特殊文字はアポストロフィです。 パスフレーズに一重引用符（'）、二重引用符（"）、または通貨記号を含めることはできません。 パスフレーズを使用するには、DBMOVER メンバのセキュリティ設定を SECURITY=(1,N)以上ににして PowerExchange リスナを実行します。詳細については、『PowerExchange リファレンスマニュアル』の「SECURITY 文」を参照してください。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。
データベース名	データベースインスタンス名。
場所	DB2 に接続する PowerExchange リスナの場所のノード名。ノード名は、PowerExchange の構成ファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。
環境 SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。

プロパティ	説明
データベースファイルの上書き	<p>i5/OS のデータベースファイルオーバーライドを次の形式で指定します。</p> <p>from_file/to_library/to_file/to_member</p> <p>説明：</p> <ul style="list-style-type: none"> - from_file はオーバーライドされるファイルです。 - to_library は新しく使用するライブラリです。 - to_file は新しく使用するライブラリ内のファイルです。 - to_member はオプションで、新しく使用するライブラリおよびファイル内のメンバです。何も指定されない場合、*FIRST が使用されます。 <p>単一の接続において 8 回まで、一意のファイルオーバーライドを指定することができます。1 つのオーバーライドは 1 つのソースまたはターゲットに適用されます。複数のファイルオーバーライドを指定する場合は、ファイルオーバーライドの文字列を二重引用符 (") で囲み、各ファイルオーバーライドの間にスペースを入れます。</p> <p>注: 【ライブラリリスト】 と 【データベースファイルの上書き】 の両方を指定していて、テーブルにその両方が存在する場合、【データベースファイルの上書き】 の値が優先されます。</p>
ライブラリリスト	<p>Select、Insert、Delete、または Update 文のテーブル名を修飾するために PowerExchange が検索するライブラリのリスト。テーブル名が修飾されていない場合、PowerExchange はリストを検索します。</p> <p>ライブラリはコンマで区切ります。</p> <p>注: 【ライブラリリスト】 と 【データベースファイルの上書き】 の両方を指定していて、テーブルにその両方が存在する場合、【データベースファイルの上書き】 の値が優先されます。</p>
コードページ	<p>ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。</p>
使用する SQL 識別子文字	<p>データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。</p> <p>データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。</p> <p>データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。</p>
大文字と小文字が混在する識別子をサポート	<p>データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を 【SQL 識別子文字】 プロパティに対して選択された文字で囲みます。</p> <p>【SQL 識別子文字】 プロパティが [なし] に設定されている場合、【大文字と小文字が混在する識別子をサポート】 プロパティは無効になっています。</p>
分離レベル	<p>トランザクションのコミット範囲。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - CS。カーソルの安定性。 - RR。繰り返し可能な読み込み。 - CHG。変更。 - ALL <p>デフォルトは CS です。</p>

プロパティ	説明
暗号化タイプ	<p>オプション。データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - AES <p>デフォルトは [なし] です。</p> <p>注: [暗号化タイプ] および [暗号化レベル] 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。</p> <p>PowerExchange ネットワークへの SSL 認証の実装の詳細については、『<i>PowerExchange リファレンスマニュアル</i>』を参照してください。</p>
暗号化レベル	<p>[暗号タイプ] で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。</p> <ul style="list-style-type: none"> - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 <p>[暗号化タイプ] に [AES] を選択しない場合、このオプションは無視されます。</p> <p>デフォルトは 1 です。</p>
ページングサイズ	<p>オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ページングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。</p> <p>最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。</p>
行として解釈	<p>オプション。このオプションは、ページングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ページングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ページングサイズはキロバイトで表示されています。</p>
圧縮	<p>オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。</p>
配列サイズ	<p>オプション。ワーカースレッドのためのストレージアレイのレコード数。このオプションは、[ワーカースレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 25~5000 です。デフォルトは 25 です。</p>
書き込みモード	<p>オプション。データ統合サービスが PowerExchange リスナにデータを送信するモード。次のうちのいずれかの書き込みモードを選択します。</p> <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。このオプションはエラーリカバリを優先する場合に選択します。ただし、このオプションを使用するとパフォーマンスが落ちる可能性があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションは、エラーが発生したときにターゲットテーブルを再ロードしてもよい場合に使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションはエラー検出も有効にします。このオプションは、CONFIRMWRITEOFF のスピードと CONFIRMWRITEON のデータの完全性を結合します。 <p>デフォルトは [CONFIRMWRITEON] です。</p>
拒否ファイル	<p>拒否ファイルに対して PWXR のデフォルトのプレフィックスをオーバーライドします。書き込みモードが ASYNCHRONOUSWITHFAULTTOLERANCE のときは、PowerExchange は拒否ファイルをターゲットマシンに作成します。拒否ファイルを作成しないようにするには、PWXDISABLE と入力します。</p>

IBM DB2 for z/OS 接続のプロパティ

IBM DB2 for z/OS 接続を使用し、IBM DB2 for z/OS 内のテーブルにアクセスします。IBM DB2 for z/OS 接続はリレーショナルデータベース接続です。IBM DB2 for z/OS 接続は、Administrator ツールまたは Developer ツールで管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、DB2 for z/OS 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、255 文字を超えることはできません。
接続タイプ	接続タイプ (DB2Z)。
ユーザー名	データベースユーザー名。
パスワード	指定したユーザー名のパスワードまたは有効な PowerExchange のパスフレーズ。 PowerExchange パスフレーズは 9~128 文字の長さで指定し、次の文字を含めることができます。 - 英字 (大文字と小文字) - 数字 (0~9) - スペース - 次に示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? 注: 先頭に書かれている特殊文字はアポストロフィです。 パスフレーズに一重引用符 (')、二重引用符 (")、または通貨記号を含めることはできません。 パスフレーズを使用するには、DBMOVER メンバのセキュリティ設定を SECURITY=(1,N)以上にし、PowerExchange リスナを実行します。詳細については、『PowerExchange リファレンスマニュアル』の「SECURITY 文」を参照してください。 IBM IRRPHREX 出口に指定できる文字は、PowerExchange パスフレーズに指定できる文字に影響しません。 注: 有効な RACF のパスフレーズは最長 100 文字です。PowerExchange は、検証のために RACF にパスフレーズを渡すときに 100 文字を超えるパスフレーズを切り捨てます。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。
DB2 サブシステム ID	DB2 サブシステムの名前。

プロパティ	説明
場所	DB2 に接続する PowerExchange リスナの場所のノード名。ノード名は、PowerExchange の構成ファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。
環境 SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。
関連 ID	DB2 要求用の DB2 関連 ID を形成するために、PWX のプレフィックスと連結される値。
コードページ	ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。
使用する SQL 識別子文字	データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。 データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。 データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。
大文字と小文字が混在する識別子をサポート	データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を 【SQL 識別子文字】 プロパティに対して選択された文字で囲みます。 【SQL 識別子文字】 プロパティが [なし] に設定されている場合、 【大文字と小文字が混在する識別子をサポート】 プロパティは無効になっています。
暗号化タイプ	オプション。データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。 - なし - AES デフォルトは [なし] です。 注: 【暗号化タイプ】 および 【レベル】 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。 PowerExchange ネットワークへの SSL 認証の実装の詳細については、『 <i>PowerExchange リファレンスマニュアル</i> 』を参照してください。
暗号化レベル	【暗号タイプ】 で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。 - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 【暗号化タイプ】 に [AES] を選択しない場合、このオプションは無視されます。 デフォルトは 1 です。
ページングサイズ	オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ページングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。 最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。
行として解釈	オプション。このオプションは、ページングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ページングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ページングサイズはキロバイトで表示されています。

プロパティ	説明
圧縮	オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。
オフロード処理	オプション。バルクデータの処理をソースマシンからデータ統合サービスのマシンにオフロードするかどうかを制御します。次のいずれかのオプションを選択します。 <ul style="list-style-type: none"> - AUTO。オフロード処理を使用するかどうかはデータ統合サービスによって決定されます。 - はい。オフロード処理を使用します。 - いいえ。オフロード処理は使用しません。 デフォルトは [いいえ] です。
ワーカースレッド	オプション。オフロード処理の有効時にデータ統合サービスがバルクデータを処理するために使用するスレッドの数。最適なパフォーマンスを得るためには、データ統合サービスマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。
配列サイズ	オプション。ワーカースレッドのためのストレージアレイのレコード数。このオプションは、 [ワーカースレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 1~5000 です。デフォルトは 25 です。
書き込みモード	データ統合サービスが PowerExchange リスナにデータを送信するモード。次のうちいずれかの書き込みモードを設定します。 <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。エラーのリカバリを優先する場合に選択します。このオプションを選択すると、パフォーマンスが低下する場合があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。エラーが発生した際にターゲットテーブルをリロードできる場合には、このオプションを使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションには、エラーを検出する機能も用意されています。このオプションは書き込み確認オンのデータ整合性で、書き込み確認オフの速度を上げる。 デフォルトは CONFIRMWRITEON です。
拒否ファイル	拒否ファイルに対して PWXR のデフォルトのプレフィックスをオーバーライドします。書き込みモードが ASYNCHRONOUSWITHFAULTTOLERANCE のときは、PowerExchange は拒否ファイルをターゲットマシンに作成します。拒否ファイルを作成しないようにするには、PWXDISABLE と入力します。

IMS 接続のプロパティ

IMS データベースにアクセスするには IMS 接続を使用します。IMS 接続は、非リレーショナルメインフレームデータベースタイプの接続です。データ統合サービスは PowerExchange 経由で IMS に接続します。IMS 接続

は Developer ツールで作成します。IMS 接続は、Administrator ツールまたは Developer ツールで管理できません。

以下の表に、IMS 接続のプロパティを示します。

オプション	説明
場所	IMS に接続する PowerExchange リスナの場所のノード名。ノード名は、PowerExchange のコンフィギュレーションファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。
ユーザー名	データベースユーザー名。
パスワード	<p>指定したデータベースユーザー名のパスワードまたは有効な PowerExchange のパスフレーズ。</p> <p>PowerExchange パスフレーズは 9～128 文字の長さで指定し、次の文字を含めることができます。</p> <ul style="list-style-type: none"> - 英字（大文字と小文字） - 数字（0～9） - スペース - 次を示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>注: 先頭に書かれている特殊文字はアポストロフィです。</p> <p>パスフレーズに一重引用符 (')、二重引用符 (")、または通貨記号を含めることはできません。</p> <p>IBM IRRPHREX イグジットに指定できる文字は、PowerExchange パスフレーズに指定できる文字に影響しません。</p> <p>注: 有効な RACF のパスフレーズは最長 100 文字です。PowerExchange は、検証のために RACF にパスフレーズを渡すときに 100 文字を超えるパスフレーズを切り捨てます。</p> <p>IMS への接続のパスフレーズを使用するときは、次の要件を確実に満足させてください。</p> <ul style="list-style-type: none"> - PowerExchange リスナは、DBMOVER メンバでセキュリティ設定を SECURITY=(1,N)かそれ以上に設定したうえで実行する必要があります。詳細は、『PowerExchange リファレンスマニュアル』の「SECURITY ステートメント」を参照してください。 - ODBA の IMS へのアクセスは、『PowerExchange ナビゲータユーザーガイド』で説明されているとおりに設定する必要があります。 - 必ずアクセス方式に IMS ODBA を指定している IMS データマップを使用してください。DL/1 BATCH アクセス方式を指定するデータマップは使用しないでください。これは、パスフレーズをサポートしていない netport ジョブの使用がこのアクセス方式で必須になっているためです。 - IMS データベースは、ODBA の IMS へのアクセスを使用するために、IMS 制御領域でオンラインにする必要があります。
コードページ	必須。データソースの読み取りや書き込みに使用するコードページの名前。通常この値は ISO コードページの名前です (ISO-8859-6 など)。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。

オプション	説明
暗号化タイプ	<p>データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - AES <p>デフォルトは [なし] です。</p> <p>注: [暗号化タイプ] および [レベル] 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。PowerExchange ネットワークへの SSL 認証の実装の詳細については、『<i>PowerExchange リファレンスマニュアル</i>』を参照してください。</p>
[暗号化] レベル	<p>[暗号タイプ] で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。</p> <ul style="list-style-type: none"> - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 <p>[暗号化タイプ] に [AES] を選択しない場合、このオプションは無視されます。デフォルトは 1 です。</p>
ペーシングサイズ	<p>オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ペーシングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。</p> <p>最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。</p>
行として解釈	<p>オプション。このオプションは、ペーシングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ペーシングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ペーシングサイズはキロバイトで表示されています。</p>
圧縮	<p>オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。</p>
オフロード処理	<p>オプション。バルクデータの処理をソースマシンからデータ統合サービスのマシンにオフロードするかどうかを制御します。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - AUTO。オフロード処理を使用するかどうかがデータ統合サービスによって決定されます。 - はい。オフロード処理を使用します。 - いいえ。オフロード処理は使用しません。 <p>デフォルトは [AUTO] です。</p>
ワーカースレッド	<p>オプション。オフロード処理の有効時にデータ統合サービスがバルクデータを処理するために使用するスレッドの数。最適なパフォーマンスを得るためには、データ統合サービスマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。</p>

オプション	説明
配列サイズ	オプション。ワーカースレッドのためのストレージアレイのレコード数。このオプションは、 [ワーカースレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 1~5000 です。デフォルトは 25 です。
書き込みモード	<p>オプション。データ統合サービスが PowerExchange リスナにデータを送信する際のモード。次のうちいずれかの書き込みモードを選択します。</p> <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。このオプションはエラーリカバリを優先する場合に選択します。ただし、このオプションを使用するとパフォーマンスが落ちる可能性があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションは、エラーが発生したときにターゲットテーブルを再ロードしてもよい場合に使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションはエラー検出も有効にします。このオプションは、CONFIRMWRITEOFF のスピードと CONFIRMWRITEON のデータの完全性を結合します。 <p>デフォルトは [CONFIRMWRITEON] です。</p>

JDBC 接続のプロパティ

JDBC 接続を使用してデータベースのテーブルにアクセスすることができます。JDBC 接続は、Administrator ツール、Developer ツール、または Analyst ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、JDBC 接続のプロパティを示します。

プロパティ	説明
データベースタイプ	データベースのタイプ。
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。</p> <p>~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。</p>
説明	接続の説明。説明は、765 文字を超えることはできません。
ユーザー名	データベースユーザー名。
パスワード	データベースユーザー名のパスワード。

プロパティ	説明
JDBC ドライバクラス名	<p>JDBC ドライバクラスの名前。</p> <p>以下の表に、該当するデータベースタイプに入力できる、ドライバクラス名を示します。</p> <ul style="list-style-type: none"> - Oracle 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.oracle.OracleDriver</code> - IBM DB2 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.db2.DB2Driver</code> - Microsoft SQL Server 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.sqlserver.SQLServerDriver</code> - Sybase ASE 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.sybase.SybaseDriver</code> - Informix 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.informix.InformixDriver</code> - MySQL 用 DataDirect JDBC ドライバクラス名: <code>com.informatica.jdbc.mysql.MySQLDriver</code> - Databricks Delta Lake 用 JDBC ドライバ: Databricks からダウンロードしたドライバの名前。 ドライバの詳細については、『<i>Data Engineering Integration ガイド</i>』の「Databricks の統合を開始する前に」の章でストレージアクセスの設定に関するトピックを参照してください。 <p>特定のデータベースで使用するドライバクラスの詳細については、ベンダ提供のドキュメントを参照してください。</p>
接続文字列	<p>データベースへの接続に使用する接続文字列。以下の接続文字列を使用します。</p> <p><code>jdbc:<subprotocol>:<subname></code></p> <p>特定のドライバで使用する接続文字列の詳細については、ベンダ提供のマニュアルを参照してください。</p>
環境 SQL	<p>オプション。データベースに接続するとき、SQL コマンドを入力してデータベースの環境設定を行います。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。</p> <p>注: Sqoop を有効にすると、Sqoop はこのプロパティを無視します。</p>
トランザクション SQL	<p>オプション。データベースに接続するとき、SQL コマンドを入力してデータベースの環境設定を行います。データ統合サービスにより、トランザクション環境 SQL が各トランザクションの開始時に実行されます。</p> <p>注: Sqoop を有効にすると、Sqoop はこのプロパティを無視します。</p>
SQL 識別子文字	<p>データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。</p> <p>データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。</p> <p>データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。</p> <p>注: Sqoop を有効にすると、Sqoop はこのプロパティを無視します。</p>
大文字と小文字が混在する識別子をサポート	<p>データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を【SQL 識別子文字】プロパティに対して選択された文字で囲みます。</p> <p>【SQL 識別子文字】プロパティが[なし]に設定されている場合、【大文字と小文字が混在する識別子をサポート】プロパティは無効になっています。</p> <p>注: Sqoop を有効にする場合、ターゲットをランタイムに作成または置換する DDL スクリプトを生成して実行すると、Sqoop はこのプロパティを受け入れます。他のすべてのシナリオでは、Sqoop はこのプロパティを無視します。</p>

プロパティ	説明
Sqoop コネクタを使用	<p>JDBC 接続を使用するデータオブジェクトの Sqoop 接続を可能にします。データ統合サービスは、Sqoop を使用して Hadoop ランタイム環境でマッピングを実行します。</p> <p>JDBC 準拠のデータベースに基づくリレーショナ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、および論理データオブジェクトに Sqoop 接続を設定できます。</p> <p>Sqoop 接続を有効にするには【Sqoop v1.x】を選択します。</p> <p>デフォルトは【なし】です。</p>
Sqoop 引数	<p>Sqoop がデータベースに接続するために使用する必要のある引数を入力します。引数が複数の場合はスペースで区切ります。</p> <p>Sqoop 用の Teradata Connector for Hadoop (TDCH) 専用コネクタを使用して Blaze エンジンでマッピングを実行するには、Sqoop 引数で TDCH 接続ファクトリクラスを定義する必要があります。接続ファクトリクラスは、使用する TDCH Sqoop コネクタによって異なります。</p> <ul style="list-style-type: none"> - Cloudera Connector Powered by Teradata を使用するには、次の Sqoop 引数を設定します。 <ul style="list-style-type: none"> -Dsqaop.connection.factories=com.cloudera.connector.teradata.TeradataManagerFactory - Hortonworks Connector for Teradata (Teradata Connector for Hadoop 提供) を使用するには、次の Sqoop 引数を設定します。 <ul style="list-style-type: none"> -Dsqaop.connection.factories=org.apache.sqaop.teradata.TeradataManagerFactory <p>Spark エンジンでマッピングを実行するために、Sqoop 引数で TDCH 接続ファクトリクラスを定義する必要はありません。デフォルトでは、データ統合サービスは Cloudera Connector Powered by Teradata と Hortonworks Connector for Teradata (Teradata Connector for Hadoop 提供) を呼び出します。</p> <p>注: 専用の Cloudera コネクタや Hortonworks コネクタではなく、汎用の JDBC コネクタを使用してマッピングを実行するには、JDBC 接続で--driver および--connection-manager の Sqoop 引数を定義する必要があります。マッピングの読み取りまたは書き込みのトランスフォーメーションで--driver 引数および--connection-manager 引数を定義すると、Sqoop は引数を無視します。</p> <p>Sqoop 引数が入力されなかった場合、データ統合サービスは、JDBC 接続プロパティに基づいて、Sqoop コマンドを作成します。</p>

JDBC V2 接続のプロパティ

JDBC V2 接続をセットアップする際には、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、JDBC V2 接続のプロパティを示します。

プロパティ	説明
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*() -+={[]}\ ';",<,>./?</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。</p>
説明	<p>接続の説明。説明は、4,000 文字を超えることはできません。</p>

プロパティ	説明
場所	接続を作成するドメイン。
タイプ	接続タイプ。[JDBC V2] を選択します。

【詳細】 タブには、JDBC V2 接続の接続属性が含まれています。以下の表に、接続属性を示します。

プロパティ	説明
ユーザー名	データベースユーザー名。 Type 4 JDBC ドライバをサポートするデータベースへのアクセス権限のあるユーザー名。
パスワード	データベースユーザー名のパスワード。
スキーマ名	オプション。データベース内で接続するスキーマ名。 スキーマ名を指定しない場合は、データベース内で使用できるすべてのスキーマがリストされます。
JDBC ドライバクラス名	JDBC ドライバクラスの名前。 以下の表に、該当するデータベースタイプに入力できる、ドライバクラス名を示します。 <ul style="list-style-type: none"> - Azure SQL データベースの JDBC ドライバクラス名: com.microsoft.sqlserver.jdbc.SQLServerDriver - Aurora PostgreSQL の JDBC ドライバクラス名: org.postgresql.Driver - SAP HANA データベースの JDBC ドライバクラス名: com.sap.db.jdbc.Driver 特定のデータベースで使用するドライバクラスの詳細については、サードパーティベンダ提供のドキュメントを参照してください。
接続文字列	データベースへの接続に使用する接続文字列。 以下の接続文字列を使用します。 jdbc:<subprotocol>:<subname> 次のリストに、該当するデータベースの種類で入力できる接続文字列の例を示します。 <ul style="list-style-type: none"> - Azure SQL データベース JDBC ドライバの接続文字列: jdbc:sqlserver://<host>:<port>;database=<database_name> - Aurora PostgreSQL JDBC ドライバの接続文字列: jdbc:postgresql://<host>:<port>[/<database_name>] - SAP HANA データベースドライバの接続文字列: jdbc:sap://<host>:<port>/?databaseName=<Database_Name> 特定のドライバで使用する接続文字列の詳細については、サードパーティベンダ提供のドキュメントを参照してください。
サブタイプ	接続先のデータベースタイプ。 以下のデータベースタイプから接続先を選択できます。 <ul style="list-style-type: none"> - [Azure SQL データベース]。Azure SQL データベースに接続します。 - [PostgreSQL]。Aurora PostgreSQL データベースに接続します。 - [SAP HANA データベース]。SAP HANA データベースに接続します。 - [その他]。C タイプ 4 JDBC ドライバをサポートする任意のデータベースに接続します。

プロパティ	説明
大文字と小文字が混在する識別子をサポート	<p>データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を【SQL 識別子文字】プロパティに対して選択された文字で囲みます。</p> <p>例えば、Aurora PostgreSQL データベースは大文字と小文字の混在をサポートします。Aurora PostgreSQL データベースに接続するには、このプロパティを有効にする必要があります。</p> <p>【SQL 識別子文字】プロパティが【なし】に設定されている場合、【大文字と小文字が混在する識別子をサポート】プロパティは無効になっています。</p>
SQL 識別子文字	<p>データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。</p> <p>データベースで通常識別子が使用される場合、【なし】を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。</p> <p>データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。</p> <p>注: SAP HANA データベースサブタイプを指定する場合は、【SQL 識別子文字】を【なし】として選択します。</p>

JD Edwards EnterpriseOne 接続プロパティ

JD Edwards EnterpriseOne 接続を使用して、JD Edwards EnterpriseOne オブジェクトに接続します。

次の表に、JD Edwards EnterpriseOne 接続プロパティを示します。

プロパティ	説明
名前	<p>接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。128 文字以内で指定し、空白および以下の特殊文字は使用できません。</p> <p>~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /</p>
ID	<p>データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。</p>
説明	<p>接続の説明。説明は、765 文字を超えることはできません。</p>
場所	<p>接続を作成する Informatica ドメイン。</p>
タイプ	<p>接続タイプ。JD Edwards EnterpriseOne を選択します。</p>
ホスト名	<p>JD Edwards EnterpriseOne サーバーのホスト名。</p>
エンタープライズポート	<p>JD Edwards EnterpriseOne サーバーのポート番号。デフォルトは 6016 です。</p>
ユーザー名	<p>JD Edwards EnterpriseOne データベースユーザーの名前。</p>

プロパティ	説明
パスワード	JD Edwards EnterpriseOne データベースユーザーのパスワード。
環境	接続先の JD Edwards EnterpriseOne 環境の名前。
ロール	JD Edwards EnterpriseOne ユーザーのロール。デフォルトは [*すべて] です。
ユーザー名	JD Edwards EnterpriseOne データベースユーザーの名前。
パスワード	データベースユーザーのパスワード。
ドライバクラス名	<p>以下の表に、該当するデータベースタイプに入力できる、ドライバクラス名を示します。</p> <ul style="list-style-type: none"> - Oracle 用 DataDirect JDBC ドライバクラス名: com.informatica.jdbc.oracle.OracleDriver - IBM DB2 用 DataDirect JDBC ドライバクラス名: com.informatica.jdbc.db2.DB2Driver - Microsoft SQL Server 用 DataDirect JDBC ドライバクラス名: com.informatica.jdbc.sqlserver.SQLServerDriver <p>特定のデータベースで使用するドライバクラスの詳細については、ベンダ提供のドキュメントを参照してください。</p>
接続文字列	<p>データベースへの接続に使用する接続文字列。以下の接続文字列を使用します。</p> <p>JDBC 接続文字列では、次の構文を使用します。</p> <ul style="list-style-type: none"> - Oracle の場合: jdbc:informatica:oracle://<host name>:<port>,ServiceName=<db service name> - DB2 の場合: jdbc:informatica:db2://<host name>:<port>;databaseName=<db name> - Microsoft SQL の場合: jdbc:informatica:sqlserver://<host name>:<port>;databaseName=<db name>

Kafka 接続のプロパティ

Kafka 接続はメッセージング接続です。Kafka 接続を使用して、ソースまたはターゲットとして、Apache Kafka Broker にアクセスします。Developer tool または infacmd で Kafka 接続を作成および管理できます。

Kafka 接続を設定する場合は、次のプロパティを設定します。

- 接続が読み取りまたは書き込みを行う Kafka Broker のリスト。
- データベースへの接続が失敗した場合に統合サービスが再接続を試行する秒数。
- Kafka メッセージングブローカのバージョン。Kafka メッセージングブローカのバージョンを、Apache 0.10.1.1 以降に設定します。

全般プロパティ

次の表に、Kafka 接続の一般的な接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。接続を識別するために使用できる文字列を入力します。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。ドメイン名を選択します。
タイプ	接続タイプ。[メッセージ/Kafka] を選択します。

Kafka Broker のプロパティ

次の表に、Kafka 接続の Kafka Broker プロパティを示します。

プロパティ	説明
Kafka Broker リスト	Kafka メッセージングブローカの設定を保持する Kafka Broker のカンマ区切りリスト。 Kafka Broker を指定するには、次の形式を使用します。 <IP Address>:<port>
再試行タイムアウト	統合サービスが、データの読み取りまたは書き込みのために Kafka Broker への再接続を試行した後の秒数。指定した時間にソースまたはターゲットが利用できない場合は、データ損失を回避するために、マッピング実行が停止します。

プロパティ	説明
Kafka Broker のバージョン	Kafka メッセージングブローカのバージョンを、Apache 0.10.1.1 以降に設定します。
追加接続プロパティ	<p>オプション。Kafka ブローカーに接続するための接続プロパティのカンマ区切りリスト。</p> <p>例えば、次のような構文を使用できます。</p> <pre>request.timeout.ms=<value>,session.timeout.ms=<value>, fetch.max.wait.ms=<value>,heartbeat.interval.ms=<value>, security.protocol=SASL_PLAINTEXT,sasl.kerberos. service.name=<kerberos name>,sasl.mechanism=GSSAPI, sasl.jaas.config=com.sun.security.auth.module. Krb5Login Modulerequired useKeyTab=true doNotPrompt=true storeKey=true client=true keyTab="<Keytab Location>" principal="<principal>"</pre> <p>Kafka Broker への接続にかかる時間を短縮するには、次のプロパティを設定していることを確認してください。</p> <ul style="list-style-type: none"> - request.timeout.ms - session.timeout.ms - fetch.max.wait.ms - heartbeat.interval.ms <p>安全な方法で Kafka Broker に接続するには、security.protocol プロパティに次のいずれかの値を設定していることを確認してください。</p> <ul style="list-style-type: none"> - SASL_SSL - SSL <p>security.protocol プロパティのデフォルト値は SASL_PLAINTEXT です。</p> <p>テクニカルプレビュー: 追加の接続プロパティは、テクニカルプレビューに利用できます。テクニカルプレビュー機能はサポートされていますが、保証対象外で本番環境には対応していません。非本番環境のみでの使用をお勧めします。</p> <p>接続のプロパティの詳細については、https://kafka.apache.org/documentation/を参照してください。</p>

SSL プロパティ

次の表に、Kafka 接続の SSL プロパティを示します。

プロパティ	説明
SSL モード	<p>必須。接続に使用する暗号化タイプを示す SSL モード。</p> <p>次の SSL モードからモードを選択できます。</p> <ul style="list-style-type: none"> - 無効 - 一方向 - 双方向
SSL トラストストアファイルパス	<p>【一方向】 の SSL モードが選択されている場合に必要です。</p> <p>信頼済み SSL サーバーの証明書を含む SSL トラストストアファイルの絶対パスおよびファイル名。</p>
SSL トラストストアパスワード	<p>【一方向】 の SSL モードが選択されている場合に必要です。</p> <p>SSL トラストストアのパスワード。</p>

プロパティ	説明
SSL キーストアファイルパス	【双方向】 の SSL モードが選択されている場合に必要です。 SSL サーバーのプライベートキーと証明書を含む SSL キーストアファイルの絶対パスおよびファイル名。
SSL キーストアパスワード	【双方向】 の SSL モードが選択されている場合に必要です。 SSL キーストアのパスワード。

infacmd を使用した Kafka 接続の作成

infacmd コマンドラインプログラムを使用して、Kafka 接続を作成できます。

UNIX で Kafka 接続を作成するには、次のコマンドを実行します。

```
sh infacmd.sh createConnection -dn <domain name> -un <domain user> -pd <domain password> -cn <connection name>
-cid <connection id> -ct Kafka -o kfkBrkList=<host1:port1>,<host2:port2>,<host3:port3>
kafkabrokerVersion=<version> additionalConnectionProperties=<additional properties>
```

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

Kudu 接続プロパティ

Kudu 接続を使用して Kudu にアクセスします。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Kudu 接続は、Administrator ツールまたは Developer tool で作成および管理できます。次の表に、Kudu 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={} \`;"'<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[Kudu] を選択します。

次の表に、メタデータアクセスのプロパティを示します。

プロパティ	説明
Kudu マスタ URL	Kudu マスタテーブルの URL。
Kudu ライブラリバージョン	Kudu ライブラリのバージョン番号。
クラスタ設定	接続に使用する Hadoop クラスタ。

LDAP 接続プロパティ

LDAP 接続を使用して、LDAP オブジェクトに接続します。

次の表に、LDAP 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成する Informatica ドメイン。
タイプ	接続タイプ。LDAP を選択します。
ホスト名	LDAP ディレクトリサーバーのホスト名。デフォルトは localhost です。
ポート	LDAP ディレクトリサーバーのポート番号。デフォルトは 389 です。
匿名接続	LDAP ディレクトリサーバーとの匿名接続を確立します。匿名接続を選択し、認証不要の匿名ユーザーとしてディレクトリサーバーにアクセスします。 注: アクティブディレクトリとは匿名接続を確立できません。
ユーザー名	LDAP ディレクトリサーバーに接続するための LDAP ユーザー名。
パスワード	LDAP ディレクトリサーバーに接続するためのパスワード。
セキュアな接続	TLS プロトコル経由で LDAP ディレクトリサーバーとのセキュアな接続を確立します。

プロパティ	説明
TrustStore のファイル名	LDAP ディレクトリサーバーとのセキュアな接続を確立するための TLS 証明書を含むトラストストアのファイル名。 デフォルトは、infa_truststore.jks です。 セキュアな接続を選択した場合は必須です。トラストストアのファイル名とパスワードについては、LDAP 管理者にお問い合わせください。
TrustStore のパスワード	SSL 証明書を含むトラストストアファイルのパスワード。
KeyStore のファイル名	LDAP ディレクトリサーバーとのセキュアな通信を確立するために必要なキーと証明書を含むキーストアのファイル名。 セキュアな接続を選択した場合は必須です。キーストアのファイル名とパスワードについては、LDAP 管理者にお問い合わせください。
KeyStore のパスワード	通信を安全に行うために必要なキーストアファイルのパスワードです。

Microsoft Azure Blob ストレージ接続のプロパティ

Microsoft Azure SQL Blob Storage 接続を使用して Microsoft Azure Blob Storage にアクセスします。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Microsoft Azure Blob Storage 接続は、Administrator ツールまたは Developer tool で作成および管理できます。次の表に、Microsoft Azure Blob ストレージ接続のプロパティを示します。

プロパティ	説明
名前	Microsoft Azure Blob ストレージ接続の名前。
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[Azure Blob Storage] を選択します。

【接続の詳細】タブには、Microsoft Azure Blob ストレージ接続の接続属性が含まれています。以下の表に、接続属性を示します。

プロパティ	説明
アカウント名	Microsoft Azure ストレージアカウントの名前。
認証タイプ	認証タイプ。以下のいずれかの認証メカニズムを選択できます。 <ul style="list-style-type: none">- 共有キー認証- 共有アクセス署名
アカウントキー	Microsoft Azure Storage アクセスキー。共有キー認証を選択する場合に該当します。
SAS トークン	Microsoft Azure ポータルでアカウント用に生成した SAS トークンを含む SAS URI。共有アクセス署名認証タイプを選択する場合に該当します。 注: 有効な SAS トークンを含む有効な SAS URI を指定する必要があります。
コンテナ名	ルートコンテナまたは絶対パスを持つサブフォルダ。 注: 複合ファイルをインポートするには、ルートコンテナのみを指定します。
エンドポイントサフィックス	Microsoft Azure エンドポイントのタイプ。以下の任意のエンドポイントを選択できます。 <ul style="list-style-type: none">- core.windows.net: デフォルト- core.usgovcloudapi.net: 米国合衆国政府の Microsoft Azure エンドポイントを選択する場合- core.chinacloudapi.cn: 該当なし

Microsoft Azure Cosmos DB SQL API 接続のプロパティ

Cosmos DB 接続を使用して、Cosmos DB データベースに接続します。Cosmos DB 接続の作成時には、メタデータおよびデータアクセスの情報を入力します。

プロパティ	説明
名前	
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	
タイプ	Microsoft Azure Cosmos DB SQL API
Cosmos DB URI	Microsoft Azure Cosmos DB アカウントの URI。

プロパティ	説明
キー	
データベース	JSON ドキュメントとの間での読み書きするコレクションが格納されているデータベース名。

注: Cosmos DB URI とキー値は、Azure ポータルのキー設定から見つかります。詳細については、Azure 管理者に問い合わせてください。

Microsoft Azure Data Lake Storage Gen1 接続のプロパティ

Microsoft Azure Data Lake Storage Gen1 接続を使用して Microsoft Azure Data Lake Storage Gen1 にアクセスします。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Microsoft Azure Data Lake Storage Gen1 接続は、Administrator ツールまたは Developer tool で作成および管理できます。以下の表に、Microsoft Azure Data Lake Storage Gen1 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[] \\:;'"<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[Microsoft Azure Data Lake Storage Gen1] を選択します。

次の表に、メタデータアクセスのプロパティを示します。

プロパティ	説明
ADLS アカウント名	Microsoft Azure Data Lake Storage Gen1 の名前。
ClientID	Active Directory で OAuth 認証を完了するためのアプリケーションの ID。
クライアントシークレット	Active Directory で OAuth 認証を完了するためのクライアント秘密鍵。

プロパティ	説明
ディレクトリ	データの読み取りまたは書き込みに使用する Microsoft Azure Data Lake Storage Gen1 ディレクトリ。デフォルトはルートディレクトリです。
AuthEndpoint	クライアント ID に基づいてアクセスコードが生成され、クライアントシークレットが完了する OAuth 2.0 トークンエンドポイント。

クライアント ID、クライアントシークレット、および認証エンドポイントの作成に関する詳細については、Azure 管理者にお問い合わせいただくか、Microsoft Azure Data Lake Storage Gen1 のドキュメントを参照してください。

Microsoft Azure Data Lake Storage Gen2 接続のプロパティ

Microsoft Azure Data Lake Storage Gen2 接続を使用して Microsoft Azure Data Lake Storage Gen2 にアクセスします。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Microsoft Azure Data Lake Storage Gen2 接続は、Administrator ツールまたは Developer tool で作成および管理できます。以下の表に、Microsoft Azure Data Lake Storage Gen2 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[]] \\:;""'<,>.:?/
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[Microsoft Azure Data Lake Storage Gen2] を選択します。

次の表に、メタデータアクセスのプロパティを示します。

プロパティ	説明
アカウント名	Microsoft Azure Data Lake Storage Gen2 アカウント名またはサービス名。
認証タイプ	Microsoft Azure Data Lake Storage Gen2 アカウントにアクセスするための認証タイプ。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none">- 【サービスプリンシパル認証】。クライアント ID、クライアントシークレット、およびテナント ID を使用して Microsoft Azure Data Lake Storage Gen2 に接続する場合に選択します。- 【共有キー認証】。アカウントキーを使用して Microsoft Azure Data Lake Storage Gen2 に接続する場合に選択します。 注: ストリーミングマッピングで共有キー認証を使用することはできません。
クライアント ID	サービスプリンシパル認証に適用されます。Azure Active Directory (AD) で OAuth 認証を完了するためのアプリケーションの ID。
クライアントシークレット	サービスプリンシパル認証に適用されます。Azure Active Directory で OAuth 認証を完了するためのクライアントシークレットキー。
テナント ID	サービスプリンシパル認証に適用されます。Azure Active Directory のディレクトリ ID。
アカウントキー	共有キー認証に適用されます。Microsoft Azure Data Lake Storage Gen2 アカウントのアカウントキー。
ファイルシステム名	Microsoft Azure Data Lake Storage Gen2 の既存のファイルシステムの名前。
ディレクトリパス	ファイルシステム名を使用していない既存のディレクトリのパス。デフォルトのディレクトリはありません。以下のいずれかの構文を選択できます。 <ul style="list-style-type: none">- / (ルートディレクトリの場合)。- /dir1- dir1/dir2
Adls Gen2 エンドポイント	Microsoft Azure エンドポイントのタイプ。以下のいずれかのエンドポイントを選択できます。 <ul style="list-style-type: none">- core.windows.net: デフォルト- core.usgovcloudapi.net: Azure Government エンドポイントを選択する場合

クライアント ID、クライアントシークレット、テナント ID、およびファイルシステム名の作成に関する詳細については、Azure 管理者にお問い合わせいただくか、Microsoft Azure Data Lake Storage Gen2 のドキュメントを参照してください。

Microsoft Azure SQL Data Warehouse 接続プロパティ

Microsoft Azure SQL Data Warehouse 接続を使用して Microsoft Azure SQL Data Warehouse にアクセスします。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

Microsoft SQL Server 接続は、Administrator ツールまたは Developer tool で作成および管理できます。次の表に、Microsoft Azure SQL Data Warehouse 接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+=[] \\:;'"<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。[Azure SQL Data Warehouse] を選択します。

次の表に、メタデータアクセスのプロパティを示します。

プロパティ	説明
Azure DW JDBC URL	Microsoft Azure Data Warehouse JDBC 接続文字列。例えば、次の接続文字列を入力できます: jdbc:sqlserver://<Server>.database.windows.net:1433;database=<Database> 管理者は Microsoft Azure ポータルから URL をダウンロードできます。
Azure DW JDBC ユーザー名	Microsoft Azure SQL Data Warehouse アカウントに接続するためのユーザー名。Microsoft Azure SQL Data Warehouse のデータを読み取り、書き込み、切り詰めるための権限が必要です。
Azure DW JDBC パスワード	Microsoft Azure SQL Data Warehouse アカウントに接続するためのパスワード。
Azure DW スキーマ名	Microsoft Azure SQL Data Warehouse 内のスキーマの名前。
Azure Storage のタイプ	ファイルをステージングする Azure ストレージのタイプ。次のいずれかのストレージタイプを選択できます。 - Azure Blob デフォルト。Microsoft Azure Blob Storage を使用してファイルをステージングする場合。 - ADLS Gen2Microsoft Azure Data Lake Storage Gen2 をストレージとして使用して、ファイルをステージングする場合。
Azure Blob アカウント名	ファイルをステージングする Microsoft Azure ストレージアカウントの名前。
Azure Blob アカウントキー	Blob ストレージアカウントへのアクセスを認証するキー。
ADLS Gen2 ストレージアカウント名	ファイルをステージングするための Microsoft Azure Data Lake Storage Gen2 ストレージアカウントの名前。

プロパティ	説明
ADLS Gen2 アカウントキー	ファイルをステージングするための Microsoft Azure Data Lake Storage Gen2 アクセスキー。
Blob エンドポイント	Microsoft Azure エンドポイントのタイプ。 次のいずれかのオプションを選択します。 <ul style="list-style-type: none"> - core.windows.net: デフォルト - core.usgovcloudapi.net: 米国政府の Microsoft Azure Data Warehouse エンドポイントにアクセスする場合に選択します。 - core.chinacloudapi.cn: 中国リージョンの Microsoft Azure Data Warehouse エンドポイントにアクセスする場合に選択します。 ネイティブ環境および Spark エンジンでマッピングを実行する場合は、米国政府の Microsoft Azure エンドポイントを設定できます。
VNet ルール	仮想ネットワーク (VNet) にある Microsoft Azure SQL Data Warehouse エンドポイントへの接続を有効にします。

MS SQL Server 接続のプロパティ

Microsoft SQL Server に接続するには、Microsoft SQL Server 接続を使用します。Microsoft SQL Server 接続は、Microsoft SQL Server リレーショナルデータベースへの接続です。Microsoft SQL Server 接続は、Administrator ツールまたは Developer tool で作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、MS SQL Server 接続のプロパティを示します。

プロパティ	説明
データベースタイプ	データベースタイプ。
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
信頼関係接続の使用	アプリケーションサービスで Windows 認証を使用してデータベースにアクセスできるようにします。アプリケーションサービスを起動するユーザー名は、データベースにアクセスできる Windows ユーザーである必要があります。デフォルトでは、このオプションはクリアされています。 注: Windows 認証と NTLM 認証は、Linux でホストされる Microsoft SQL Server 2017 バージョンでは認定されていません。
ユーザー名	データベースユーザー名。Microsoft SQL Server が NTLMv1 または NTLMv2 認証を使用する場合に必須です。

プロパティ	説明
パスワード	データベースユーザー名のパスワード。Microsoft SQL Server が NTLMv1 または NTLMv2 認証を使用する場合に必須です。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。接続のパススルーセキュリティを有効にすると、ドメインは、接続オブジェクトに定義されている資格情報ではなく、クライアントのユーザー名とパスワードを使用して対応するデータベースにログインします。
メタデータアクセスプロパティ: 接続文字列	<p>データベースからメタデータにアクセスするために使用する接続文字列。以下の接続文字列を使用します。</p> <pre>jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=<database name></pre> <p>NTLM 認証を使用した接続をテストするには、接続文字列に次のパラメータを追加します。</p> <ul style="list-style-type: none"> - AuthenticationMethod。使用する NTLM 認証のバージョン。 注: UNIX では、NTLMv1 と NTLMv2 はサポートされますが、NTLM はサポートされません。 - ドメイン。SQL サーバーが属するドメイン。 <p>次の例では、Informatica.com という名前の NT ドメインで NTLMv2 認証を使用する SQL サーバーへの接続文字列を示します。</p> <pre>jdbc:informatica:sqlserver:// host01:1433;DatabaseName=SQL1;AuthenticationMethod=ntlm2java;Domain=Informatica.com</pre> <p>NTLM 認証で接続する場合、MS SQL Server 接続プロパティの【信頼関係接続の使用】オプションを有効にできます。NTLMv1 または NTLMv2 認証で接続する場合、接続プロパティにユーザー名とパスワードを入力する必要があります。</p>

プロパティ	説明
AdvancedJDBCSecurityOptions	<p>セキュアデータベースへのメタデータアクセスのためのデータベースパラメータ。Informatica は、AdvancedJDBCSecurityOptions フィールドの値を機密データとして扱い、パラメータ文字列を暗号化して格納します。</p> <p>セキュアデータベースに接続するには、以下のパラメータを含めます。</p> <ul style="list-style-type: none"> - EncryptionMethod。必須。ネットワーク上で送信される際にデータが暗号化されるかどうかを示します。このパラメータは SSL に設定する必要があります。 - ValidateServerCertificate。オプション。データベースサーバーによって送信される証明書を Informatica が検証するかどうかを示します。 このパラメータを True に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証します。HostNameInCertificate パラメータを指定すると、Informatica は証明書内のホスト名も検証します。 このパラメータを false に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証しません。指定するトラストストア情報がすべて無視されます。 - HostNameInCertificate。オプション。セキュアデータベースをホストするマシンのホスト名。ホスト名を指定すると、Informatica は接続文字列に含められたそのホスト名を SSL 証明書内のホスト名と照らして検証します。 - cryptoProtocolVersion。オプション。Microsoft SQL Server インスタンスに対して TLS を有効にする場合は、次のように cryptoProtocolVersion パラメータを設定します。 cryptoProtocolVersion=TLSv<version number> 例: cryptoProtocolVersion=TLSv1.2 注: バージョン番号はサーバーに設定した TLS バージョンと同じにする必要があります。 - TrustStore。必須。トラストストアファイルのパスおよびファイル名です。 注: SSL または TLS を設定してファイル名のみを指定する場合は、トラストストアファイルを Informatica のインストールディレクトリにコピーする必要があります。接続をテストしメタデータをインポートするには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica client installation directory>/clients/DeveloperClient マッピングを実行するには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica server installation directory>/tomcat/bin - TrustStorePassword。必須。セキュアデータベースに対するトラストストアファイルのパスワード。 ODBC には適用されません。 注: セキュア JDBC パラメータは接続文字列に自動的に追加されます。セキュア JDBC パラメータを接続文字列に直接指定する場合は、AdvancedJDBCSecurityOptions フィールドにはパラメータを入力しないでください。
データアクセスプロパティ: プロバイダタイプ	<p>Microsoft SQL Server データベースへの接続に使用する接続プロバイダ。次のプロバイダタイプを選択することができます。</p> <ul style="list-style-type: none"> - ODBC - Oledb (廃止予定) <p>デフォルトは ODBC です。</p> <p>注: Microsoft SQL Server 接続ユーザーインターフェースには OLEDB プロバイダタイプが廃止予定と表示されますが、Informatica は OLEDB プロバイダタイプをサポートしています。OLEDB プロバイダタイプのサポート情報の詳細については、ナレッジベースの記事 KB 522895 を参照してください。</p>

プロパティ	説明
DSN の使用	<p>データ統合サービスで接続のデータソース名を使用できるようになります。</p> <p>[DSN の使用] オプションを使用した場合、データ統合サービスはデータベース名とサーバー名を DSN から取得します。</p> <p>[DSN の使用] オプションを選択しない場合、データベース名とサーバー名を指定する必要があります。</p>
接続文字列	<p>DSN モードを有効にしない場合、次の接続文字列を使用します。</p> <p><server name>@<database name></p> <p>DSN モードを有効にする場合、次の接続文字列を使用します。</p> <p><DSN Name></p>
コードページ	ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。
ドメイン名	ドメインの名前。
パケットサイズ	データ転送で使用されるパケットサイズ。Microsoft SQL Server のネイティブドライバの最適化に使用されます。
所有者名	<p>スキーマの所有者の名前。</p> <p>注: 動的マッピングまたは [DDL の生成と実行] オプションを使用してテーブル DDL を生成すると、DDL メタデータにスキーマ名と所有者名のプロパティが含まれません。</p>
スキーマ名	<p>データベースのスキーマの名前。スキーマ名がデータベースユーザー名と異なる場合は、プロファイリングウェアハウスのスキーマ名を指定する必要があります。スキーマ名がデータベースのユーザー名と異なり、ユーザーが管理するキャッシュテーブルを設定する場合は、データオブジェクトキャッシュデータベースのスキーマ名を指定する必要があります。</p> <p>注: 動的マッピングまたは [DDL の生成と実行] オプションを使用してテーブル DDL を生成すると、DDL メタデータにスキーマ名と所有者名のプロパティが含まれません。</p>
環境 SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。
トランザクション SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、トランザクション環境 SQL を各トランザクションの開始時に実行します。
リトライ期限	このプロパティは、将来使用するために予約されています。
SQL 識別子文字	<p>データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。</p> <p>データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。</p> <p>データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。</p>

プロパティ	説明
大文字と小文字が混在する識別子をサポート	データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を【SQL 識別子文字】プロパティに対して選択された文字で囲みます。 【SQL 識別子文字】プロパティが【なし】に設定されている場合、【大文字と小文字が混在する識別子をサポート】プロパティは無効になっています。
ODBC プロバイダ	ODBC。ODBC の接続先となるデータベースのタイプ。プッシュダウンの最適化用に、データベースタイプを指定して、データ統合サービスでネイティブデータベース SQL を生成できるようにします。次のオプションがあります。 - その他 - Sybase - Microsoft_SQL_Server デフォルトは【その他】です。

Netezza 接続のプロパティ

Netezza 接続を使用して、Netezza データベースにアクセスします。Netezza 接続は、データベース接続です。Netezza 接続は、Administrator ツールまたは Developer tool で作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Netezza 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。【Netezza】を選択します。
ユーザー名	Netezza データベースにアクセスするための適切な権限を持つユーザー名。
パスワード	上記データベースユーザー名のパスワード。
JDBC Url	Netezza データベースへの接続時に Developer tool で使用する必要がある JDBC URL。次の形式を使用します。 jdbc:netezza://<ホスト名>:<ポート>/<データベース名>

プロパティ	説明
接続文字列	Netezza データベースへの接続のために使用する ODBC データソースの名前。
タイムアウト	Developer tool が接続を閉じる前に Netezza データベースからの応答を待機する秒数。

OData 接続のプロパティ

OData 接続を使用して、OData URL にアクセスします。OData 接続は、Web 接続です。OData 接続は、Administrator ツールまたは Developer tool で作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、OData 接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。【OData】を選択します。
ユーザー名	オプション。OData リソースからデータを読み取るための適切な権限を持つユーザー名。
パスワード	オプション。OData URL ユーザー名のパスワード。
URL	読み取る必要があるデータを公開している OData サービスのルート URL。
セキュリティタイプ	オプション。OData サーバーへの安全な接続を確立するために Developer tool で使用する必要があるセキュリティプロトコル。 次のいずれかの値を選択します。 - なし - SSL - TLS デフォルトは「なし」です。
トラストストアファイル名	セキュリティタイプを選択する場合、必須です。 OData サーバーの公開証明書が含まれるトラストストアファイルの名前。 デフォルトは、infa_truststore.jks です。

プロパティ	説明
パスワード	セキュリティタイプを選択する場合、必須です。 OData サーバーの公開証明書が含まれるトラストストアファイルのパスワード。
キーストアファイル名	セキュリティタイプを選択する場合、必須です。 OData サーバーのプライベートキーが含まれるキーストアファイルの名前。 デフォルトは、infa_truststore.jks です。
パスワード	セキュリティタイプを選択する場合、必須です。 OData サーバーのプライベートキーが含まれるキーストアファイルのパスワード。

ODBC 接続のプロパティ

ODBC データにアクセスするには、ODBC 接続を使用します。ODBC 接続はリレーショナルデータベース接続です。ODBC 接続は、Administrator ツール、Developer ツール、または Analyst ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、ODBC 接続のプロパティを示します。

プロパティ	説明
データベースタイプ	データベースのタイプ。
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~`!\$%^&*() - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
ユーザー名	データベースユーザー名。
パスワード	データベースユーザー名のパスワード。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。接続のパススルーセキュリティを有効にすると、ドメインは、接続オブジェクトに定義されている資格情報ではなく、クライアントのユーザー名とパスワードを使用して対応するデータベースにログインします。
データアクセスプロパティ: 接続文字列	データベースのメタデータへのアクセスに使用する ODBC 接続 URL。 <data source name>

プロパティ	説明
コードページ	ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。
環境 SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。
トランザクション SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、トランザクション環境 SQL を各トランザクションの開始時に実行します。
リトライ期限	このプロパティは、将来使用するために予約されています。
SQL 識別子文字	データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。 データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。 データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。
大文字と小文字が混在する識別子をサポート	データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を 【SQL 識別子文字】 プロパティに対して選択された文字で囲みます。 【SQL 識別子文字】 プロパティが 【なし】 に設定されている場合、 【大文字と小文字が混在する識別子をサポート】 プロパティは無効になっています。
ODBC プロバイダ	ODBC を接続するデータベースのタイプ。プッシュダウンの最適化用に、データベースタイプを指定して、データ統合サービスでネイティブデータベース SQL を生成できるようにします。次のオプションがあります。 - その他 - Sybase - Microsoft_SQL_Server - Snowflake デフォルトは 【その他】 です。

注: UNIX または Linux 上でデータ統合サービスを実行する場合、ODBC 接続を使用して Microsoft SQL Server に接続します。Windows 上でデータ統合サービスを実行する場合、ネイティブ接続を使用して Microsoft SQL Server に接続します。

Oracle 接続のプロパティ

Oracle データベースに接続するには、Oracle 接続を使用します。Oracle 接続は、リレーショナル接続タイプです。Oracle 接続は、Administrator ツール、Developer ツール、または Analyst ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Oracle 接続のプロパティを示します。

プロパティ	説明
データベースタイプ	データベースタイプ。
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
ユーザー名	データベースユーザー名。
パスワード	データベースユーザー名のパスワード。
パススルーセキュリティ有効	接続のパススルーセキュリティを有効にする。接続のパススルーセキュリティを有効にすると、ドメインは、接続オブジェクトに定義されている資格情報ではなく、クライアントのユーザー名とパスワードを使用して対応するデータベースにログインします。
メタデータアクセスプロパティ: 接続文字列	データベースからメタデータにアクセスするために使用する接続文字列。以下の接続文字列を使用します。 jdbc:informatica:oracle://<host_name>:<port>;SID=<database name> Oracle database through Oracle Connection Manage を介して Oracle データベースに接続するには、次の接続文字列を使用します。 jdbc:informatica:oracle:TNSNamesFile=<fully qualified path to the tnsnames.ora file>;TNSServerName=<TNS server name>;

プロパティ	説明
AdvancedJDBCSecurityOptions	<p>セキュアデータベースへのメタデータアクセスのためのデータベースパラメータ。Informatica は、AdvancedJDBCSecurityOptions フィールドの値を機密データとして扱い、パラメータ文字列を暗号化して格納します。</p> <p>セキュアデータベースに接続するには、以下のパラメータを含めます。</p> <ul style="list-style-type: none"> - EncryptionMethod。必須。ネットワーク上で送信される際にデータが暗号化されるかどうかを示します。このパラメータは SSL に設定する必要があります。 - ValidateServerCertificate。オプション。データベースサーバーによって送信される証明書を Informatica が検証するかどうかを示します。 このパラメータを True に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証します。HostNameInCertificate パラメータを指定すると、Informatica は証明書内のホスト名も検証します。 このパラメータを false に設定すると、Informatica はデータベースサーバーによって送信される証明書を検証しません。指定するトラストストア情報がすべて無視されます。 - HostNameInCertificate。オプション。セキュアデータベースをホストするマシンのホスト名。ホスト名を指定すると、Informatica は接続文字列に含められたそのホスト名を SSL 証明書内のホスト名と照らして検証します。 - cryptoProtocolVersion。オプション。Oracle インスタンスに対して TLS を有効にする場合は、次のように cryptoProtocolVersion パラメータを設定します。 cryptoProtocolVersion=TLSv<version number> 例: cryptoProtocolVersion=TLSv1.2 注: バージョン番号はサーバーに設定した TLS バージョンと同じにする必要があります。 - TrustStore。必須。トラストストアファイルのパスおよびファイル名です。 注: SSL または TLS を設定してファイル名のみを指定する場合は、トラストストアファイルを Informatica のインストールディレクトリにコピーする必要があります。接続をテストしメタデータをインポートするには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica client installation directory>/clients/DeveloperClient マッピングを実行するには、トラストストアファイルを次のディレクトリにコピーします。 <Informatica server installation directory>/tomcat/bin - TrustStorePassword。必須。セキュアデータベースに対するトラストストアファイルのパスワード。 - KeyStore。必須。キーストアファイルのパスとファイル名。 - KeyStorePassword。必須。セキュアデータベースに対するキーストアファイルのパスワード。 <p>注: セキュア JDBC パラメータは接続文字列に自動的に追加されます。セキュア JDBC パラメータを接続文字列に直接指定する場合は、AdvancedJDBCSecurityOptions フィールドにはパラメータを入力しないでください。</p>
データアクセスプロパティ: 接続文字列	<p>以下の接続文字列を使用します。</p> <p><database name>.world</p>
コードページ	<p>ソースデータベースからの読み取りや、ターゲットデータベースまたはファイルへの書き込みに使用されるコードページ。</p>

プロパティ	説明
環境 SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、データベースに接続するたびに、接続環境 SQL を実行します。
トランザクション SQL	データベースに接続する場合に、データベース環境を設定する SQL コマンド。データ統合サービスは、トランザクション環境 SQL を各トランザクションの開始時に実行します。
リトライ期限	このプロパティは、将来使用するために予約されています。
パラレルモードを有効にする	データを一括モードでテーブルにロードする場合の並列処理を可能にします。デフォルトでは、このオプションはクリアされています。
SQL 識別子文字	データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。 データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。 データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。
大文字と小文字が混在する識別子をサポート	データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を【SQL 識別子文字】プロパティに対して選択された文字で囲みます。 【SQL 識別子文字】プロパティが [なし] に設定されている場合、【大文字と小文字が混在する識別子をサポート】プロパティは無効になっています。

Salesforce 接続のプロパティ

Salesforce 接続を使用して Salesforce オブジェクトに接続します。Salesforce 接続はアプリケーション接続タイプです。Salesforce 接続は、Administrator ツールまたは Developer ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Salesforce 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。

プロパティ	説明
場所	接続を作成する Informatica ドメイン。
タイプ	接続タイプ。標準接続タイプまたは OAuth 接続タイプを選択できます。
ユーザ名	標準接続タイプに適用されます。Salesforce のユーザー名。
ユーザーパスワード	<p>標準接続タイプに適用されます。Salesforce のユーザー名に対するパスワード。</p> <p>所属する組織の信頼済みネットワークの外部で Salesforce にアクセスするには、セキュリティトークンをパスワードに追加して、API またはデスクトップクライアントにログインする必要があります。</p> <p>セキュリティトークンを受信またはリセットするには、Salesforce にログインして、[設定] > [個人情報] > [セキュリティトークンのリセット] の順にクリックします。</p> <p>パスワードでは、大文字と小文字が区別されます。</p>
サービスの URL	<p>アクセスする Salesforce サービスの URL。</p> <p>以下に例を示します。</p> <p><code>https://login.salesforce.com/services/Soap/u/54.0</code></p> <p>テストまたは開発環境で、Salesforce Sandbox テスト環境にアクセスできます。Salesforce Sandbox の詳細については、Salesforce のマニュアルを参照してください。</p>
リフレッシュトークン	OAuth 接続タイプに適用されます。Salesforce の更新トークン。
コンシューマキー	OAuth 接続タイプに適用されます。リフレッシュトークンの生成に必要な Salesforce から取得したコンシューマキー。コンシューマキーの生成方法の詳細については、Salesforce のマニュアルを参照してください。
コンシューマシークレット	OAuth 接続タイプに適用されます。リフレッシュトークンの生成に必要な Salesforce から取得したコンシューマシークレット。コンシューマシークレットの生成方法の詳細については、Salesforce のマニュアルを参照してください。

Salesforce Marketing Cloud 接続のプロパティ

Salesforce Marketing Cloud 接続を使用して Salesforce Marketing Cloud オブジェクトに接続します。Salesforce Marketing Cloud 接続は、Administrator ツールまたは Developer tool で作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Salesforce Marketing Cloud 接続のプロパティを示します。

接続プロパティ	説明
名前	Salesforce Marketing Cloud 接続の名前。
ID	データ統合サービスは ID を使用して接続を識別します。

接続プロパティ	説明
説明	オプション。接続の説明。
場所	接続を作成する Informatica ドメイン。
タイプ	接続タイプ。[Salesforce Marketing Cloud] を選択します。
Salesforce Marketing Cloud の URL	<p>データ統合サービスが Salesforce Marketing Cloud WSDL への接続に使用する URL。</p> <p>次に、OAuth 1.0 URL の URL の例を示します。 https://webservice.s7.exacttarget.com/etframework.wsdl</p> <p>次に、OAuth 2.0 URL の URL の例を示します。 <a href="https://<SUBDOMAIN>.soap.marketingcloudapis.com/etframework.wsdl">https://<SUBDOMAIN>.soap.marketingcloudapis.com/etframework.wsdl</p> <p>Salesforce Marketing Cloud で OAuth 1.0 のサポートが終了する前に、OAuth 2.0 にアップグレードすることをお勧めします。</p>
ユーザー名	Salesforce Marketing Cloud アカウントのユーザー名。
パスワード	Salesforce Marketing Cloud アカウントのパスワード。
ClientId	有効なアクセストークンを生成するために必要な Salesforce Marketing Cloud のクライアント ID。
ClientSecret	有効なアクセストークンを生成するために必要な Salesforce Marketing Cloud のクライアントシークレット。
ロギングの有効化	ロギングを有効にすると、タスクのセッションログを参照できます。
UTC オフセット	Secure Agent は、UTC オフセットの接続プロパティを使用して、UTC オフセットタイムゾーンの Salesforce Marketing Cloud との間でデータの読み書きを行います。
バッチサイズ	<p>Secure Agent がバッチでターゲットに書き込む行数。</p> <p>データを挿入または更新し、コンタクトキーを指定するときに、指定したコンタクト ID に関連付けられているデータが、1 つのバッチで Salesforce Marketing Cloud に挿入または更新されます。</p> <p>Salesforce Marketing Cloud にデータを更新/挿入する場合は、コンタクトキーを指定しないようにします。</p>
複数の BU の有効化	Salesforce Marketing Cloud アカウントに複数のビジネスユニットがある場合は、このオプションを選択します。Salesforce Marketing Cloud 接続を使用して、すべてのビジネスユニットのデータにアクセスできます。

SAP 接続のプロパティ

SAP テーブルまたは SAP BW オブジェクトにアクセスするには、SAP 接続を使用します。SAP 接続は、エンタープライズアプリケーション接続の一種です。SAP 接続は、Administrator ツールまたは Developer ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、SAP 接続のプロパティを示します。

プロパティ	説明
ユーザー名	必須。アクセス先の SAP ソースシステムのユーザー名。
パスワード	必須。ユーザー名に対するパスワード。
接続タイプ	<p>必須。作成する接続のタイプ。</p> <p>次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - アプリケーション。特定の SAP アプリケーションサーバーに接続する際にアプリケーション接続を作成します。 - 負荷分散。SAP 負荷分散を使用する場合は、負荷分散接続を作成します。 <p>デフォルトは [アプリケーション] です。</p> <p>選択した接続タイプに応じて、対応する接続プロパティフィールドが【接続の詳細】ダイアログボックスで使用可能になります。Developer tool では、特定の接続タイプで使えない接続プロパティフィールドが灰色で表示されます。</p>
ホスト名	<p>SAP アプリケーション接続を作成する場合は必須。</p> <p>接続先の SAP サーバーのホスト名または IP アドレス。</p>
システム番号	<p>SAP アプリケーション接続を作成する場合は必須。</p> <p>SAP システム番号。</p>
メッセージホスト名	<p>SAP 負荷分散接続を作成する場合は必須。</p> <p>SAP メッセージサーバーのホスト名。</p>
R3 名/SysID	<p>SAP 負荷分散接続を作成する場合は必須。</p> <p>SAP システムの名前。</p>
グループ	<p>SAP 負荷分散接続を作成する場合は必須。</p> <p>SAP アプリケーションサーバーのグループ名。</p>
クライアント	必須。SAP クライアント番号。
言語	<p>オプション。マッピングおよびワークフローで使用する言語。</p> <p>Developer tool のコードページと互換性のある言語でなければなりません。</p> <p>このオプションを空白のままにすると、Developer tool は、SAP システムのデフォルトの言語を使用します。</p>
トレース	<p>オプション。このオプションは、SAP システムによる JCo 呼び出しを追跡する場合に使用します。SAP では、JCo 呼び出しについての情報をトレースファイルに保存しています。</p> <p>次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> - 0. オフ - 1. フル <p>デフォルトは 0 です。</p> <p>以下のディレクトリからトレースファイルにアクセスできます。</p> <ul style="list-style-type: none"> - Informatica サービスをインストールしたマシン上のディレクトリ<Informatica インストールディレクトリ>/tomcat/bin - Developer tool をインストールしたマシン上のディレクトリ<Informatica インストールディレクトリ>/clients/DeveloperClient

プロパティ	説明
追加パラメータ	オプション。使用するその他の接続パラメータを入力します。 次の形式を使用します。 <parameter name>=<value>
ステージングディレクトリ	ステージファイルが作成される SAP システムのパス。
ソースディレクトリ	ソースファイルが保存されるパス。データ統合サービスに、このパスへのアクセス許可を与える必要があります。
FTP を使用	FTP で SAP にアクセスできるようにします。
FTP ユーザー名	FTP を使用する場合は必須。 FTP サーバーに接続するユーザー名。
FTP パスワード	FTP を使用する場合は必須。 FTP ユーザーのパスワード。
FTP ホスト	FTP を使用する場合は必須。 FTP サーバーのホスト名または IP アドレス。 オプションで、1 から 65535 までのポート番号を指定できます。FTP のデフォルトは 21 です。 次のいずれかの構文でホスト名を指定します。 - hostname:port_number - IP address:port_number ポート番号を指定する場合は、ホストマシン上でそのポート番号に対して FTP を使用可能にしてください。 SFTP を有効にする場合、SFTP サーバ用のホスト名またはポート番号を指定する必要があります SFTP のデフォルトは 22 です。
リトライ期限	FTP ホストへの接続が失敗した場合に、データ統合サービスによって再接続が試行される秒数。 リトライ期限までに FTP ホストに再接続できなかった場合、マッピングまたはワークフローは失敗します。 デフォルトは 0 です。値 0 は、リトライ期限を無期限にすることを意味します。
SFTP の使用	SFTP が SAP にアクセスできるようにします。
パブリックキーファイル名	SFTP が有効で、SFTP サーバーがパブリックキー認証を使用している場合は必須。 パブリックキーファイルのパスおよびファイル名。
プライベートキーファイル名	SFTP が有効で、SFTP サーバーがパブリックキー認証を使用している場合は必須。 プライベートキーファイルのパスおよびファイル名。
プライベートキーファイル名パスワード	SFTP が有効で、SFTP サーバーがパブリックキー認証を使用しており、プライベートキーを暗号化する場合に必須。 プライベートキーファイルを復号化するためのパスワード。

プロパティ	説明
ポート範囲	<p>データ統合サービスが、SAP サーバーからストリーミングモードでデータを読み取る際に使用する必要がある HTTP ポート範囲。</p> <p>最小ポート番号と最大ポート番号をハイフンでつないで入力します。最小ポート番号と最大ポート番号は、10000 - 65535 の範囲内で指定します。組織に応じたポート範囲を指定することもできます。</p> <p>デフォルトは 10000～65535 です。</p>
HTTPS の使用	<p>SAP テーブルからデータを読み取る際はこのオプションを選択して HTTPS ストリーミングを有効にします。</p> <p>デフォルトでは、【HTTPS の使用】チェックボックスは選択されていません。</p> <p>ストリーミングモードでテーブルリーダーマッピング用に HTTPS を設定する方法の詳細については、Informatica Documentation Portal の「HTTPS Configuration for Table Reader Mappings in Streaming Mode for PowerExchange for SAP NetWeaver」の記事を参照してください。</p>
キーストアのファイルパス	<p>HTTPS を使用する場合は必須。</p> <p>プライベートキーまたはパブリックキーのペアと関連する証明書を含むキーストアファイルへのパス。</p>
キーストアのパスワード	<p>HTTPS を使用する場合は必須。</p> <p>キーストアファイルのパスワード</p>
プライベートキーのパスワード	<p>HTTPS を使用する場合は必須。</p> <p>プライベートキーファイルを復号化するためのパスワード。</p>

シーケンシャル接続のプロパティ

ソースデータセットにアクセスするにはシーケンシャル接続を使用します。シーケンシャル接続は Developer ツールで作成します。シーケンシャル接続は、Administrator ツールまたは Developer ツールで管理できません。

シーケンシャルデータソースは、PowerExchange が SEQ のアクセス方式で定義されたデータマップを使用してアクセスできるデータソースです。データ統合サービスは PowerExchange 経由でデータソースに接続します。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、シーケンシャル接続のプロパティを示します。

オプション	説明
場所	<p>シーケンシャルなデータセットに接続する PowerExchange リスナの場所のノード名ノード名は、PowerExchange の構成ファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。</p>
ユーザー名	<p>シーケンシャルデータセットにアクセスする権限のあるユーザー名</p>

オプション	説明
パスワード	<p>指定したユーザー名のパスワードまたは有効な PowerExchange のパスフレーズ。PowerExchange パスフレーズは 9～128 文字の長さで指定し、次の文字を含めることができます。</p> <ul style="list-style-type: none"> - 英字（大文字と小文字） - 数字（0～9） - スペース - 次に示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>注: 先頭に書かれている特殊文字はアポストロフィです。</p> <p>パスフレーズに一重引用符 (')、二重引用符 (")、または通貨記号を含めることはできません。</p> <p>パスフレーズを使用するには、DBMOVE メンバのセキュリティ設定を SECURITY=(1,N) 以上にして PowerExchange リスナを実行します。詳細については、『PowerExchange リファレンスマニュアル』の「SECURITY 文」を参照してください。</p> <p>IBM IRRPHREX 出口に指定できる文字は、PowerExchange パスフレーズに指定できる文字に影響しません。</p> <p>注: 有効な RACF のパスフレーズは最長 100 文字です。PowerExchange は、検証のために RACF にパスフレーズを渡すときに 100 文字を超えるパスフレーズを切り捨てます。</p>
コードページ	<p>必須。シーケンシャルなデータセットの読み取りや書き込みに使用するコードページの名前。通常この値は ISO コードページの名前です (ISO-8859-6 など)。</p>
パススルーセキュリティ有効	<p>接続のパススルーセキュリティを有効にする。</p>
暗号化タイプ	<p>オプション。データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - AES <p>デフォルトは [なし] です。</p> <p>注: [暗号化タイプ] および [レベル] 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。</p> <p>PowerExchange ネットワークへの SSL 認証の実装の詳細については、『PowerExchange リファレンスマニュアル』を参照してください。</p>
[暗号化] レベル	<p>[暗号タイプ] で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。</p> <ul style="list-style-type: none"> - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 <p>[暗号化タイプ] に [AES] を選択しない場合、このオプションは無視されます。デフォルトは 1 です。</p>
ページングサイズ	<p>オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ページングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。</p> <p>最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。</p>

オプション	説明
行として解釈	(オプション) このオプションはページングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ページングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ページングサイズはキロバイトで表示されています。
圧縮	オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。
オフロード処理	<p>オプション。バルクデータの処理をソースマシンからデータ統合サービスのマシンにオフロードするかどうかを制御します。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - AUTO。オフロード処理を使用するかどうかデータ統合サービスによって決定されます。 - はい。オフロード処理を使用します。 - いいえ。オフロード処理は使用しません。 <p>デフォルトは [AUTO] です。</p>
ワークスレッド	オプション。オフロード処理の有効時にデータ統合サービスがバルクデータを処理するために使用するスレッドの数。最適なパフォーマンスを得るためには、データ統合サービスマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。
配列サイズ	(オプション) ワークスレッドのためのストレージアレイのレコード数。このオプションは、[ワークスレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 25~5000 です。デフォルトは 25 です。
書き込みモード	<p>オプション。データ統合サービスが PowerExchange リスナにデータを送信するモード。次のうちいずれかの書き込みモードを選択します。</p> <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。このオプションはエラーリカバリを優先する場合に選択します。ただし、このオプションを使用するとパフォーマンスが落ちる可能性があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションは、エラーが発生したときにターゲットテーブルを再ロードしてもよい場合に使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションはエラー検出も有効にします。このオプションは、CONFIRMWRITEOFF のスピードと CONFIRMWRITEON のデータの完全性を結合します。 <p>デフォルトは [CONFIRMWRITEON] です。</p>

Snowflake 接続プロパティ

Snowflake 接続を設定するときに、接続プロパティを設定する必要があります。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

次の表に、Snowflake 接続プロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および次の特殊文字は使用できません。~`!\$%^&*()-+={[}] \:;'"<, > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。 ID では大文字と小文字が区別されません。ID は 255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。 デフォルト値は接続名です。
説明	オプション。接続の説明。説明は、4,000 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。Snowflake を選択します。
ユーザー名	Snowflake アカウントに接続するユーザー名。
パスワード	Snowflake アカウントに接続するためのパスワード。
アカウント	Snowflake アカウントの名前。
ウェアハウス	Snowflake ウェアハウス名。
ロール	ユーザーに割り当てられた Snowflake ロール。
追加の JDBC URL パラメータ	以下の形式で、1 つ以上の JDBC 接続パラメータを入力します。 <param1>=<value>&<param2>=<value>&<param3>=<value>.... 以下に例を示します。 user=jon&warehouse=mywh&db=mydb&schema=public Okta SSO 認証を介して Snowflake にアクセスするには、次の形式で SAML 2.0 プロトコルを実装する Web ベースの IdP を入力します。 authenticator=https://<Your_Okta_Account_Name>.okta.com 注: Microsoft ADFS はサポートされていません。 Okta 認証の設定の詳細については、次の Web サイトを参照してください: https://docs.snowflake.net/manuals/user-guide/admin-security-fed-auth-configure-snowflake.html#configuring-snowflake-to-use-federated-authentication

Teradata Parallel Transporter 接続のプロパティ

Teradata PT 接続を使用して Teradata テーブルにアクセスします。Teradata PT 接続は、データベースタイプの接続です。Teradata PT 接続は、Administrator ツールまたは Developer tool で作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Teradata PT 接続のプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; ' " < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。Teradata PT を選択します。
ユーザー名	データベースへのアクセスに必要な読み込みおよび書き込みデータベース権限を持つ Teradata データベースユーザー名。
パスワード	Teradata データベースユーザー名のパスワード。
ドライバ名	Teradata JDBC ドライバの名前。
接続文字列	データベースからメタデータにアクセスするために使用する接続文字列。 以下の接続文字列を使用します。 jdbc:teradata://<hostname>/database=<database name>,tmode=ANSI,charset=UTF8

以下の表に、データアクセスのプロパティを示します。

プロパティ	説明
TDPID	Teradata データベースマシンの名前、または IP アドレス。
データベース名	Teradata データベース名。 データベース名を入力しない場合、Teradata PT API はデフォルトのログインデータベース名を使用します。
データコードページ	データベースに関連付けられているコードページ。 Teradata ターゲットへデータを書き込むマッピングを実行する場合、Teradata PT 接続のコードページは Teradata ターゲットのコードページと同じでなければなりません。 デフォルトは UTF-8。

プロパティ	説明
Tenacity	Teradata データベース上で最大数の操作が実行されている場合に、Teradata PT API がログオンの試行を続ける時間数（時間単位）。 ゼロ以外の正の整数である必要があります。デフォルト値は 4 です。
最大セッション数	Teradata PT API が Teradata データベースとの間で確立するセッションの最大数。 ゼロ以外の正の整数である必要があります。デフォルト値は 4 です。
最小セッション数	Teradata PT API ジョブを継続するために必要な Teradata PT API セッションの最大数。 1～【最大セッション数】の値までの正の整数である必要があります。デフォルトは 1 です。
スリープ	Teradata データベース上で最大数の操作が実行されている場合に、Teradata PT API がログオンを再試行する前に休止する時間数（分単位）。 ゼロ以外の正の整数である必要があります。デフォルト値は 6 です。
TDCH にメタデータ JDBC URL を使用	メタデータアクセスプロパティの接続文字列に指定した JDBC URL を、Teradata Connector for Hadoop (TDCH) が使用する必要があることを示しています。 デフォルトは選択されています。 TDCH がマッピングを実行するときに使用する必要のある別の JDBC URL を入力するには、このオプションをクリアします。
TDCH JDBC URL	TDCH が Teradata マッピングを実行するときに使用する必要のある JDBC URL を入力します。次の形式を使用します。 <code>jdbc:teradata://<hostname>/database=<database name>,tmode=ANSI,charset=UTF8</code> このフィールドは、【TDCH にメタデータ JDBC URL を使用】オプションをクリアにした場合にのみ使用できます。
データ暗号化	Windows で、SQL の要求、応答、およびデータの完全なセキュリティ暗号化を有効にします。 デフォルトでは無効になっています。
Sqoop の追加の引数	このプロパティは、Hortonworks または Cloudera クラスタを使用していて、Sqoop を通じて Blaze エンジンまたは Spark エンジンで Teradata マッピングを実行する場合に適用されます。 データを処理するために Sqoop が使用する必要のある引数を入力します。例えば、「--method split.by.amp」と入力します。引数が複数の場合はスペースで区切ります。 指定可能な引数の一覧については、Hortonworks for Teradata Connector および Cloudera Connector Powered by Teradata のドキュメントを参照してください。 注: Hortonworks for Teradata Connector を使用していて、読み取り操作で 2 つ以上のソーステーブルを追加する場合は、--split-by 引数が必要です。Cloudera Connector Powered by Teradata を使用していて、ソーステーブルにプライマリキーが定義されていない場合は、ソース接続で--split-引数が必要です。
認証タイプ	ユーザーを認証する方法。 次のいずれかの認証タイプを選択します。 - ネイティブ。接続で指定した Teradata データベースに対してユーザー名およびパスワードを認証します。 - LDAP。外部 LDAP のディレクトリサービスに対してユーザークレデンシャルを認証します。 デフォルトはネイティブです。

Tableau 接続のプロパティ

Tableau に接続するには、Tableau 接続を使用します。Tableau 接続を作成する場合は、Tableau にアクセスするための情報を入力します。

次の表に、Tableau 接続のプロパティを示します。

プロパティ	説明
名前	Tableau 接続の名前。
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成する Informatica ドメイン。
タイプ	接続タイプ。Tableau を選択します。

次の表は、Tableau に接続するためのプロパティを示しています。

接続プロパティ	説明
Tableau 製品	接続する Tableau 製品の名前。 TDE または TWBX ファイルをパブリッシュするには次のいずれかの Tableau 製品を選択できます。 <ul style="list-style-type: none">- Tableau デスクトップ。データ統合サービスマシンで TDE ファイルを作成します。作成した TDE ファイルを Tableau デスクトップに手動でインポートできます。- Tableau サーバー。生成した TDE または TWBX ファイルを Tableau サーバーにパブリッシュします。- Tableau Online。生成した TDE または TWBX ファイルを Tableau Online にパブリッシュします。
接続 URL	TDE または TWBX ファイルをパブリッシュする Tableau サーバーまたは Tableau Online の URL。URL の形式は次のとおりです: http://<Tableau サーバーまたは Tableau Online のホスト名>:<ポート>
ユーザー名	Tableau サーバーまたは Tableau Online アカウントのユーザー名。
パスワード	Tableau サーバーまたは Tableau Online アカウントのパスワード。
コンテンツ URL	TDE または TWBX ファイルをパブリッシュする Tableau サーバーまたは Tableau Online 上のサイトの名前。 サイト名を入力するには Tableau 管理者にお問い合わせください。

Tableau V3 接続のプロパティ

Tableau V3 接続をセットアップする際には、接続プロパティを設定する必要があります。

次の表に、Tableau V3 接続のプロパティを示します。

プロパティ	説明
名前	
ID	データ統合サービスが接続の識別に使用する文字列。ID では大文字と小文字が区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成してから変更できません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成する Informatica ドメイン。
タイプ	接続タイプ。

次の表は、Tableau に接続するためのプロパティを示しています。

接続プロパティ	説明
Tableau 製品	接続する Tableau 製品の名前。 または TWBX ファイルをパブリッシュするには次のいずれかの Tableau 製品を選択できます。 Tableau デスクトップ。 データ統合サービスマシンでファイルを作成します。作成したファイルを Tableau デスクトップに手動でインポートできます。 Tableau Server。 生成したまたは TWBX ファイルを Tableau サーバーにパブリッシュします。 Tableau Online。 生成した または TWBX ファイルを Tableau Online にパブリッシュします。
接続 URL	.hyper ファイルのパブリッシュ先となる Tableau サーバーまたは Tableau Online の URL。 URL の形式は次のとおりです: http://<Tableau サーバーまたは Tableau Online のホスト名>:<ポート>
ユーザー名	Tableau Server または Tableau Online のユーザー名を入力してください。
パスワード	Tableau サーバーまたは Tableau Online アカウントのパスワード。

接続プロパティ	説明
サイト ID	または TWBX ファイルをパブリッシュする Tableau サーバーまたは Tableau Online 上のサイトの名前。 注: サイト ID を入力するには、Tableau 管理者にお問い合わせください。
スキーマファイルのパス	スキーマファイルパスについて、次のいずれかのオプションを入力します。 <ul style="list-style-type: none"> - .hyper ファイルへの絶対パス。 - .hyper ファイルへのディレクトリパス。 - 空のディレクトリパス。 テンプレートファイル用に指定するパスは、ターゲットファイルのデフォルトパスになります。

Twitter Streaming 接続プロパティ

Twitter Web サイトからほぼリアルタイムでデータにアクセスするには、Twitter ストリーミング接続を使用します。Twitter ストリーミング接続は、ソーシャルメディア企業のストリーミング API への接続です。Twitter ストリーミング接続は、Administrator ツールまたは Developer ツールで作成および管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、Twitter ストリーミング接続の一般的なプロパティを示します。

プロパティ	説明
名前	接続の名前。この名前では、大文字と小文字が区別されず、ドメイン内で一意にする必要があります。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	データ統合サービスが接続の識別に使用する文字列。ID の大文字と小文字は区別されません。255 文字以下で、ドメイン内で一意である必要があります。このプロパティは、接続を作成した後で変更することはできません。デフォルト値は接続名です。
説明	接続の説明。説明は、765 文字を超えることはできません。
場所	接続を作成するドメイン。
タイプ	接続タイプ。Twitter ストリーミングを選択します。

以下の表に、ホースタイプおよび OAuth 認証のプロパティを示します。

プロパティ	説明
ホースタイプ	ストリーミング API のメソッド。以下のいずれかの方式を指定できます。 <ul style="list-style-type: none">- フィルタ。Twitter statuses/filter メソッドは検索条件に一致するパブリックステータスを返します。- サンプル。Twitter statuses/sample メソッドはすべてのパブリックステータスのランダムサンプルを返します。
コンシューマキー	Twitter でアプリケーションを作成するときに取得するコンシューマキー。Twitter はこのキーを使用してアプリケーションを識別します。
コンシューマシークレット	Twitter アプリケーションを作成するときに取得するコンシューマシークレット。Twitter はこのシークレットを使用してコンシューマキーの所有権を確立します。
OAuth の詳細の有無	OAuth を設定するかどうかを示します。次のいずれかの値を選択します。 <ul style="list-style-type: none">- [はい]。アクセストークンおよびシークレットがあることを示します。- [いいえ]。OAuth Utility を起動します。
アクセストークン	OAuth Utility から返されるアクセストークン。Twitter はユーザークレデンシャルの代わりにこのトークンを使用して、保護されたリソースにアクセスします。
アクセスシークレット	OAuth Utility が返すアクセスシークレット。このシークレットがトークンの所有権を確立します。

VSAM 接続のプロパティ

VSAM 接続を使用して VSAM データテーブルにアクセスします。VSAM 接続はフラットファイル接続タイプです。VSAM 接続は Developer ツールで作成します。VSAM 接続は、Administrator ツールまたは Developer ツールで管理できます。

注: 接続プロパティの順序は、表示するツールによって異なる場合があります。

以下の表に、VSAM 接続のプロパティを示します。

オプション	説明
場所	VSAM のデータセットに接続する PowerExchange リスナの場所のノード名ノード名は、PowerExchange の構成ファイル (dbmover.cfg) 内の NODE 文の最初のパラメータで定義されます。
ユーザー名	VSAM データセットに接続する権限のあるユーザー名

オプション	説明
パスワード	<p>指定したユーザーまたは有効な PowerExchange パスフレーズのパスワード。 PowerExchange パスフレーズは 9～128 文字の長さで指定し、次の文字を含めることができます。</p> <ul style="list-style-type: none"> - 英字（大文字と小文字） - 数字（0～9） - スペース - 次に示す特殊文字。 ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>注: 先頭に書かれている特殊文字はアポストロフィです。</p> <p>パスフレーズに一重引用符 (')、二重引用符 (")、または通貨記号を含めることはできません。</p> <p>パスフレーズを使用するには、DBMOVE メンバのセキュリティ設定を SECURITY=(1,N) 以上にして PowerExchange リスナを実行します。詳細については、『PowerExchange リファレンスマニュアル』の「SECURITY 文」を参照してください。</p> <p>IBM IRRPHREX 出口に指定できる文字は、PowerExchange パスフレーズに指定できる文字に影響しません。</p> <p>注: 有効な RACF のパスフレーズは最長 100 文字です。PowerExchange は、検証のために RACF にパスフレーズを渡すときに 100 文字を超えるパスフレーズを切り捨てます。</p>
コードページ	<p>必須。VSAM データセットの読み取りや書き込みに使用するコードページの名前。通常この値は ISO コードページの名前です (ISO-8859-6 など)。</p>
パススルーセキュリティ有効	<p>接続のパススルーセキュリティを有効にする。</p>
暗号化タイプ	<p>オプション。データ統合サービスが使用する暗号化のタイプ。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - なし - AES <p>デフォルトは [なし] です。</p> <p>注: [暗号化タイプ] および [レベル] 接続プロパティを構成する代わりに、Secure Sockets Layer (SSL) 認証を使用することをお勧めします。SSL 認証は複数の Informatica 製品で使用されており、より厳格なセキュリティが提供されます。</p> <p>PowerExchange ネットワークへの SSL 認証の実装の詳細については、『PowerExchange リファレンスマニュアル』を参照してください。</p>
[暗号化] レベル	<p>[暗号化タイプ] で [AES] を選択した場合は、データ統合サービスが使用する暗号化レベルを指定するために、次に挙げるオプションのいずれかを選択してください。</p> <ul style="list-style-type: none"> - 1. 128 ビットの暗号化キーを使用します。 - 2. 192 ビットの暗号化キーを使用します。 - 3. 256 ビットの暗号化キーを使用します。 <p>[暗号化タイプ] に [AES] を選択しない場合、このオプションは無視されます。デフォルトは 1 です。</p>
ページングサイズ	<p>オプション。ソースシステムが PowerExchange リスナに渡すことができるデータ量。ページングサイズは、外部アプリケーション、データベース、またはデータ統合サービスのノードがボトルネックになっている場合に設定します。低い値を使用するとパフォーマンスが向上します。</p> <p>最小値およびデフォルト値は 0 です。値を 0 にするとパフォーマンスが最大になります。</p>

オプション	説明
行として解釈	オプション。このオプションは、ページングサイズを行数で表示する場合に選択します。このオプションをクリアすると、ページングサイズがキロバイト単位で表示されます。デフォルトではこのオプションが選択されていないため、ページングサイズはキロバイトで表示されています。
圧縮	オプション。このオプションは、ソースデータの圧縮を有効にする場合に選択します。データを圧縮することで、Informatica アプリケーションがネットワークで送信するデータ量を減少させることができます。デフォルトではこのオプションが選択されていないため、圧縮は無効になっています。
オフロード処理	<p>オプション。バルクデータの処理をソースマシンからデータ統合サービスのマシンにオフロードするかどうかを制御します。次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - AUTO。オフロード処理を使用するかどうかデータ統合サービスによって決定されます。 - はい。オフロード処理を使用します。 - いいえ。オフロード処理は使用しません。 <p>デフォルトは [AUTO] です。</p>
ワーカースレッド	オプション。オフロード処理の有効時にデータ統合サービスがバルクデータを処理するために使用するスレッドの数。最適なパフォーマンスを得るためには、データ統合サービスマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。
配列サイズ	オプション。ワーカースレッドのためのストレージアレイのレコード数。このオプションは、 [ワーカースレッド] オプションを 0 より大きな値に設定したときに適用されます。有効な値は 25~5000 です。デフォルトは 25 です。
書き込みモード	<p>オプション。データ統合サービスが PowerExchange リスナにデータを送信する際のモード。次のうちいずれかの書き込みモードを選択します。</p> <ul style="list-style-type: none"> - CONFIRMWRITEON。PowerExchange リスナにデータを送信し、応答を待ってから以降のデータを送信します。このオプションはエラーリカバリを優先する場合に選択します。ただし、このオプションを使用するとパフォーマンスが落ちる可能性があります。 - CONFIRMWRITEOFF。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションは、エラーが発生したときにターゲットテーブルを再ロードしてもよい場合に使用します。 - ASYNCHRONOUSWITHFAULTTOLERANCE。応答を待たずに、PowerExchange リスナにデータを送信します。このオプションはエラー検出も有効にします。このオプションは、CONFIRMWRITEOFF のスピードと CONFIRMWRITEON のデータの完全性を結合します。 <p>デフォルトは [CONFIRMWRITEON] です。</p>

Web サービス接続のプロパティ

Web サービスコンシューマトランスフォーメーションを Web サービスに接続するには、Web サービス接続を使用します。

以下の表に、Web サービス接続のプロパティを示します。

プロパティ	説明
ユーザー名	Web サービスに接続するユーザー名。HTTP 認証または WS-Security を有効にした場合は、ユーザー名を入力します。 Web サービスコンシューマトランスフォーメーションに WS-Security ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して動的ユーザー名を受け取ります。データ統合サービスは、接続で定義されているユーザー名をオーバーライドします。
パスワード	ユーザー名に対するパスワード。HTTP 認証または WS-Security を有効にした場合は、パスワードを入力します。 Web サービスコンシューマトランスフォーメーションに WS-Security ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して動的パスワードを受け取ります。データ統合サービスは、接続で定義されているパスワードをオーバーライドします。
エンドポイント URL	アクセス先の Web サービスの URL。データ統合サービスは、WSDL ファイルで定義されている URL をオーバーライドします。 Web サービスコンシューマトランスフォーメーションにエンドポイント URL ポートが含まれている場合、Web サービスコンシューマトランスフォーメーションは入力ポートを介して URL を動的に受け取ります。データ統合サービスは、接続で定義されている URL をオーバーライドします。
タイムアウト	データ統合サービスが、接続を閉じる前に、Web サービスプロバイダからの応答を待機する秒数。1~10000 秒の間でタイムアウト値を指定します。
HTTP 認証タイプ	HTTP 経由のユーザー認証のタイプ。次のいずれかの値を選択します。 <ul style="list-style-type: none">- なし。認証しません。- 自動: データ統合サービスが Web サービスプロバイダの認証タイプを選択します。- 基本: Web サービスプロバイダのドメインに対するユーザー名およびパスワードの入力が要求されます。データ統合サービスは、認証のためにユーザー名とパスワードを Web サービスプロバイダに送ります。- ダイジェスト: Web サービスプロバイダのドメインに対するユーザー名およびパスワードの入力が要求されます。データ統合サービスは、ユーザー名とパスワードから暗号化されたメッセージダイジェストを生成し、それを Web サービスプロバイダに送ります。プロバイダはユーザー名とパスワードの一時的な値を生成し、それをドメインコントローラの Active Directory に格納します。その値とメッセージダイジェストが比較されます。これらが一致する場合、Web サービスプロバイダはユーザを認証します。- NTLM: ドメイン名、サーバー名、またはデフォルトのユーザー名とパスワードの入力が要求されます。Web サービスプロバイダは、接続先のドメインに基づいてユーザを認証します。Windows Domain Controller からユーザー名とパスワードが取得され、ユーザが入力したユーザー名とパスワードと比較されます。これらが一致する場合、Web サービスプロバイダはユーザを認証します。NTLM 認証では、暗号化されたパスワードがドメインコントローラのアクティブディレクトリに保存されることはありません。

プロパティ	説明
WS Security タイプ	<p>使用する WS-Security のタイプ。次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - なし。データ統合サービスは、生成した SOAP リクエストに Web サービスセキュリティヘッダーを追加しません。 - PasswordText。データ統合サービスは、生成した SOAP リクエストに Web サービスセキュリティヘッダーを追加します。パスワードはクリアテキスト形式で保存されます。 - PasswordDigest。データ統合サービスは、生成した SOAP リクエストに Web サービスセキュリティヘッダーを追加します。パスワードはダイジェスト形式で保存され、ネットワーク上の再生攻撃に対する有効な保護を提供します。データ統合サービスが、ナンス値とタイムスタンプをパスワードに組み合わせます。データ統合サービスでは、そのパスワードに SHA ハッシュを適用して base64 エンコーディングでエンコードし、エンコードしたパスワードを SOAP ヘッダー内で使用します。
トラストサーティフィケーションファイル	<p>Web サービスの SSL 証明書の認証時にデータ統合サービスが使用するトラストサーティフィケーションのバンドルを含むファイル。ファイル名と完全ディレクトリパスを入力します。</p> <p>デフォルトは、<Informatica installation directory>/services/shared/bin/ca-bundle.crt です。</p>
クライアント証明書ファイル名	<p>クライアントの認証時に Web サービスで使用するクライアント証明書。Web サービスがデータ統合サービスを認証する必要がある場合にクライアントサーティフィケーションファイルを指定します。</p>
クライアント証明書のパスワード	<p>クライアントサーティフィケーションのパスワード。Web サービスがデータ統合サービスを認証する必要がある場合にクライアント証明書のパスワードを指定します。</p>
クライアント証明書のタイプ	<p>クライアントサーティフィケーションファイルの形式。次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - PEM: 「.pem」 拡張子のファイルです。 - DER: 「.cer」 または 「.der」 拡張子のファイルです。 <p>Web サービスがデータ統合サービスを認証する必要がある場合にクライアント証明書のタイプを指定します。</p>
プライベートキーファイル名	<p>クライアントサーティフィケーションのプライベートキーファイル。Web サービスがデータ統合サービスを認証する必要がある場合にプライベートキーファイルを指定します。</p>
プライベートキーのパスワード	<p>クライアントサーティフィケーションのプライベートキーのパスワード。Web サービスがデータ統合サービスを認証する必要がある場合にプライベートキーのパスワードを指定します。</p>
プライベートキーのタイプ	<p>プライベートキーのタイプ。サポートされているタイプは PEM です。</p>

データベース接続における識別子のプロパティ

多数のリレーショナルデータベース接続を作成するときには、データベースの識別子のプロパティを設定する必要があります。識別子のプロパティは、データベースにアクセスするためにデータ統合サービスが SQL クエリを生成するときにこのサービスが識別子を区切り文字で囲むかどうかを決定します。

データベース識別子は、データベースオブジェクト名です。テーブル、ビュー、カラム、インデックス、トリガ、プロシージャ、制約、ルールは、識別子を持つことができます。その識別子を使用して、SQL クエリ内でオブジェクトを参照します。データベースは、通常の識別子を持つことも、区切り文字で囲む必要がある区切り記号付き識別子を持つこともできます。

通常の識別子

通常の識別子とは、識別子の形式ルールに従った識別子のことです。通常の識別子は、SQL クエリで使用するときに区切り文字を必要としません。

例えば、以下の SQL 文では通常の識別子 *MYTABLE* と *MYCOLUMN* が使用されています。

```
SELECT * FROM MYTABLE
WHERE MYCOLUMN = 10
```

区切り記号付き識別子

区切り記号付き識別子は識別子の形式ルールに従いません。このため、区切り記号付き識別子は区切り文字で囲む必要があります。

データベースは、以下のタイプの区切り記号付き識別子を使用できます。

予約キーワードを使用する識別子

識別子が予約キーワードを使用する場合、SQL クエリでは区切り文字でその識別子を囲む必要があります。例えば、以下の SQL 文は *ORDER* という名前のテーブルにアクセスします。

```
SELECT * FROM "ORDER"
WHERE MYCOLUMN = 10
```

特殊文字を使用する識別子

識別子が特殊文字を使用する場合、SQL クエリでは区切り文字でその識別子を囲む必要があります。例えば、以下の SQL 文は *MYTABLE\$@* という名前のテーブルにアクセスします。

```
SELECT * FROM "MYTABLE$@"
WHERE MYCOLUMN = 10
```

大文字小文字を区別する識別子

デフォルトでは、IBM DB2、Microsoft SQL Server、および Oracle データベースにおける識別子は、大文字小文字を区別しません。データベースオブジェクト名は大文字で格納されますが、SQL クエリではそれらを参照するのに大文字と小文字のどちらでも使用できます。例えば、以下の SQL 文は *MYTABLE* という名前のテーブルにアクセスします。

```
SELECT * FROM mytable
SELECT * FROM MyTable
SELECT * FROM MYTABLE
```

大文字小文字を区別する識別子を使用するには、SQL クエリでは区切り文字でその識別子を囲む必要があります。例えば、以下の SQL 文は *MyTable* という名前のテーブルにアクセスします。

```
SELECT * FROM "MyTable"
WHERE MYCOLUMN = 10
```

識別子のプロパティ

多数のデータベース接続を作成するときには、データベース識別子のプロパティを設定する必要があります。設定する識別子のプロパティは、データベースが通常の識別子を使用するか、識別子内でキーワードまたは特殊文字を使用するか、それとも大文字小文字を区別する識別子を使用するかによって異なります。

データベース接続で以下の識別子のプロパティを設定します。

SQL 識別子文字

データベースが、SQL クエリで区切り識別子を囲むのに使用する文字のタイプ。使用できる文字は、データベースタイプによって異なります。

データベースで通常識別子が使用される場合、[なし] を選択します。データ統合サービスで SQL クエリが生成される際、識別子の周りに区切り文字は配置されません。

データベースで区切り識別子が使用される場合、文字を選択します。データ統合サービスで SQL クエリが生成される際、区切り識別子はこの文字で囲まれます。

大文字と小文字が混在する識別子をサポート

データベースで大文字と小文字を区別する識別子が使用される場合、有効にします。有効にした場合、データ統合サービスは、すべての識別子を **[SQL 識別子文字]** プロパティに対して選択された文字で囲みます。

Informatica クライアントツールでは、大文字と小文字を正しく使用して識別子を参照する必要があります。例えば、データベース接続を作成する際、大文字と小文字を正しく使用してデータベースユーザー名を入力する必要があります。

[SQL 識別子文字] プロパティが **[なし]** に設定されている場合、**[大文字と小文字が混在する識別子をサポート]** プロパティは無効になっています。

例: データベースが通常の識別子を使用する

この例では、データベースが通常の識別子を使用しています。どの識別子にも予約キーワードや特殊文字は含まれていません。データベースは、大文字小文字を区別しない識別子を使用します。

データベース接続で、**[SQL 識別子文字]** プロパティを **[(なし)]** に設定します。**[SQL 識別子文字]** が **[なし]** に設定されている場合、**[大文字と小文字が混在する識別子をサポート]** プロパティは無効になっています。

データ統合サービスで SQL クエリを生成する場合は、識別子を区切り文字で囲みません。

例: データベースが識別子内でキーワードまたは特殊文字を使用する

この例では、データベースは一部の識別子内でキーワードまたは特殊文字を使用しています。データベースは、大文字小文字を区別しない識別子を使用します。

データベース接続で、識別子のプロパティを以下のように設定します。

1. **[SQL 識別子文字]** プロパティを、データベースが区切り記号付き識別子に使用する文字に設定します。

この例は、プロパティを `"""` (引用符) に設定します。

2. **[大文字と小文字が混在する識別子をサポート]** プロパティをクリアします。

データ統合サービスは、SQL クエリを生成するときに、予約キーワードを使用する識別子または特殊文字を使用する識別子を、選択した文字で囲みます。例えば、データ統合サービスは以下のようなクエリを生成します。

```
SELECT * FROM "MYTABLE$@" /* identifier with special characters enclosed within delimited
                           character */
WHERE MYCOLUMN = 10 /* regular identifier not enclosed within delimited character */
```

例: 大文字小文字を区別する識別子をデータベースが使用する

この例では、大文字小文字を区別する識別子をデータベースが使用しています。このデータベースは、いくつかの識別子でキーワードまたは特殊文字を使う可能性もあれば、これらを使わない可能性もあります。

データベース接続で、識別子のプロパティを以下のように設定します。

1. **[SQL 識別子文字]** プロパティを、データベースが区切り記号付き識別子に使用する文字に設定します。

この例は、プロパティを `"""` (引用符) に設定します。

2. **[大文字と小文字が混在する識別子をサポート]** プロパティを選択します。

データ統合サービスは、SQL クエリを生成するときに、すべての識別子を選択した文字で囲みます。例えば、データ統合サービスは以下のようなクエリを生成します。

```
SELECT * FROM "MyTable"      /* case-sensitive identifier enclosed within delimited character */
WHERE "MYCOLUMN" = 10        /* regular identifier enclosed within delimited character */
```


索引

数字

10 進型のデータ型
概要 [223](#)

A

Adabas 接続
プロパティ [264](#)
Amazon Redshift 接続
プロパティ [266](#)
array
1 次元 [226](#)
インデックス [226](#)
形式 [226](#)
次元 [226](#)
多次元 [226](#)
複合データ型 [226](#)
例 [226](#)
Avro データ型
トランスフォーメーションデータ型 [230](#)

B

bigint
計算での使用 [219](#)
高精度処理 [219](#)
式の定数 [220](#)
フラットファイルへの書き込み [221](#)
Blaze エンジン
接続プロパティ [286](#)

C

Cassandra 接続
プロパティ [273](#)
CI/CD
infacmd [200](#), [202](#)
REST API [200](#), [202](#)
アプリケーションのデプロイ [202](#)
アプリケーションの再デプロイ [202](#)
オブジェクトのテスト [206](#)
オブジェクトのデプロイ [203](#), [206](#), [208](#)
オブジェクトのデプロイメント [203](#)
オブジェクトの割り当て [204](#), [207](#)
チームベース開発 [204](#)
開発者のコラボレーション [204](#)
概要 [199](#)
競合の解決 [204](#)
競合の検出 [204](#)
継続的統合 [203](#)
継続的統合の例 [205](#)
継続的配信 [205](#)

CI/CD (続く)
継続的配信の例 [208](#)
Confluent Kafka 接続
Confluent Kafka Broker のプロパティ [276](#)
infacmd を使用した作成 [277](#)
全般プロパティ [276](#)

D

Databricks 接続プロパティ [277](#)
Date/Time データ型
概要 [221](#)
DDL の生成
DDL の生成エラー [74](#)
DDL の生成と実行 [72](#)
decimal
高精度処理 [219](#), [223](#)
Decimal
データ型の変換 [255](#)
精度の損失 [224](#)
Developer tool
ワークスペースディレクトリ [18](#)
double
高精度処理 [223](#)

E

Excel
フラットファイルデータオブジェクトの編集 [112](#)
フラットファイルの設定 [112](#)
フラットファイルへのコピー [112](#)

G

Google BigQuery 接続
プロパティ [281](#)
Google PubSub
接続プロパティ [285](#)
Greenplum 接続
プロパティ [279](#)

H

HBase 接続
MapR-DB プロパティ [295](#)
プロパティ [292](#)
HDFS 接続
プロパティ [293](#)
Hive プッシュダウン
接続プロパティ [286](#)
Hive 接続
プロパティ [295](#)

HTTP 接続
プロパティ [299](#)

I

IBM DB2 for i5/OS 接続
プロパティ [304](#)
IBM DB2 for z/OS 接続
プロパティ [308](#)
IBM DB2 接続
プロパティ [301](#)
IMS 接続
プロパティ [311](#)
Informatica Data Services
概要 [17](#)
Informatica Developer
概要 [15](#)
起動 [18](#)
検索 [45](#)
設定 [23](#)
Informatica Marketplace
説明 [22](#)

J

JD Edwards EnterpriseOne 接続
プロパティ [317](#)
JDBC V2 接続
プロパティ [315](#)
JDBC 接続
プロパティ [313](#)
JSON データ型
トランスフォーメーションデータ型 [232](#)

K

Kafka 接続
infacmd を使用した作成 [321](#)
Kafka Broker のプロパティ [319](#)
全般プロパティ [319](#)
Kudu 接続
プロパティ [321](#)

L

LDAP 接続
プロパティ [322](#)

M

map
複合データ型 [227](#)
キーと値のペア [227](#)
形式 [227](#)
例 [227](#)
Microsoft Azure Data Lake Storage Gen1 接続
プロパティ [325](#)
Microsoft Azure Data Lake Storage Gen2 接続
プロパティ [326](#)
Microsoft Azure SQL Data Warehouse 接続
プロパティ [327](#)
MS SQL Server 接続
プロパティ [329](#)

N

NaN
説明 [220](#)
Netezza 接続
プロパティ [333](#)

O

OData 接続
プロパティ [281](#), [283](#), [284](#), [334](#), [351](#)
ODBC 接続
プロパティ [335](#)
Oracle 接続
プロパティ [336](#)
ORC ファイルのデータ型
トランスフォーメーションデータ型 [232](#)

P

Parquet データ型
トランスフォーメーションデータ型 [233](#)

Q

QNAN
1.への変換#QNAN [220](#)

S

Salesforce Marketing Cloud
接続プロパティ [340](#)
SAP 接続
プロパティ [341](#)
Snowflake 接続
プロパティ [346](#)
Spark HDFS ステージングディレクトリ
Hadoop 接続プロパティ [286](#)
Spark イベントログディレクトリ
Hadoop 接続プロパティ [286](#)
Spark エンジン
接続プロパティ [286](#)
Spark デプロイモード
Hadoop 接続プロパティ [286](#)
Spark 実行パラメータ
Hadoop 接続プロパティ [286](#)
SQL
環境 SQL の設定 [59](#)
環境 SQL の入力に関するガイドライン [60](#)
SQL ヒント
Developer ツールでの入力 [77](#)
struct
複合データ型 [228](#)
スキーマ [228](#)
形式 [228](#)
名前と型のペア [228](#)
例 [228](#)

T

Teradata Parallel Transporter 接続
プロパティ [348](#)

Twitter ストリーミング接続
プロパティ [352](#)

V

VSAM 接続
プロパティ [353](#)

W

Web サービス
構成 [142](#)
構成の作成 [142](#)
構成プロパティ [135](#)

Web 接続
プロパティ [299](#)

WSDL データオブジェクト
インポート [89](#)
概要ビュー [90](#)
作成 [90](#)
詳細ビュー [90](#)
スキーマビュー [89](#)
同期 [91](#)

あ

アプリケーション
アプリケーションの検証 [149](#)
アプリケーションの作成 [149](#)
データ統合サービスへのデプロイ [152](#)
ファイルへのエクスポート [154](#)
ファイルへのデプロイ [154](#)
プロパティ [150](#)
マッピングのデプロイメントのプロパティ [150](#)
ランタイムアプリケーションへの接続 [164](#)
ランタイムオブジェクトの表示 [156](#)
概要 [148](#)
完全 [149](#)
更新 [156](#)
差分 [150, 167](#)
再デプロイ [165](#)
作成 [149, 159](#)
作成、デプロイ、更新の方法 [158](#)
置換 [156](#)
アプリケーションアーカイブファイル [154](#)
アプリケーションの更新
オブジェクト [187](#)
アプリケーションパッチ
オブジェクトの影響 [181, 182, 185](#)
ガイドライン [189](#)
タイプ [175](#)
デプロイ [185, 187](#)
デプロイメント戦略 [185](#)
ルール [189](#)
依存関係
リモート [168](#)
間接 [168](#)
直接 [168](#)
影響 [181, 182, 185](#)
影響を受けるオブジェクト
検索 [184](#)
関連オブジェクト [175](#)
継承 [168, 175](#)
最も関係の近い関連オブジェクト [175](#)
作成 [187](#)

アプリケーションパッチ (続く)

削除 [183](#)
実行時 [168](#)
数 [185](#)
設計時 [168](#)
選択したオブジェクト [175](#)
直接、間接、およびリモートの依存関係 [179](#)
直接および間接の依存関係 [178](#)
直接依存関係 [177](#)
追加 [182](#)
範囲 [185](#)
変更 [182](#)
アマゾン S3 接続
プロパティ [269](#)

い

インポート
XML ファイル [211](#)
アプリケーションアーカイブ [163](#)
依存オブジェクト [210](#)
オブジェクト [214](#)
概要 [209](#)

え

エクスポート
XML ファイル [211](#)
アプリケーションアーカイブファイルへ [161, 162](#)
依存オブジェクト [210](#)
オブジェクト [212](#)
概要 [209](#)
エラーメッセージ
グループ化 [146](#)
制限 [147](#)

お

オブジェクト
コピー [30](#)
バージョン履歴 [43](#)
ロックされたオブジェクト [37](#)
オブジェクトの依存関係
オブジェクトの依存関係の表示 [144](#)
オブジェクトの依存関係の表示
オブジェクトの依存関係 [144](#)
オブジェクトの依存関係のフィルタリング [145](#)
オブジェクトの依存関係のフィルタリング
オブジェクトの依存関係 [145](#)
[オブジェクトの依存関係] ビュー
データの表示 [144](#)
オブジェクトのチェックアウトとチェックイン [42](#)
オブジェクトのチェックインとチェックアウト [42](#)

か

外部結合のサポート
カスタマイズデータオブジェクト [81](#)
カスタマイズデータオブジェクト
[個別に選択] の使用 [78](#)
カスタム SQL クエリ [75](#)
カスタムクエリの作成 [87](#)
キーの作成 [71](#)
キーリレーション [67](#)

カスタマイズデータオブジェクト (続く)

キーリレーションの作成 [71](#)

個別に選択 [78](#)

作成 [69](#)

詳細クエリ [75](#)

説明 [66](#)

ソースフィルタの入力 [78](#)

ソート済みポート [79](#)

ソート済みポートの使用 [79](#)

単純クエリ [75](#)

デフォルトクエリ [75](#)

トラブルシューティング [96](#)

マッピング実行前/実行後 SQL コマンドの追加 [87](#)

マッピング実行前/実行後 SQL コマンド [86](#)

ユーザー定義ジョイン [80](#)

ユーザー定義ジョインの入力 [81](#)

予約語ファイル [75](#)

リレーショナルデータオブジェクトの追加 [70](#)

リレーショナルリソースの追加 [70](#)

書き込みプロパティ [68](#)

カスタム SQL クエリ

カスタマイズデータオブジェクト [75](#)

作成 [87](#)

カスタムクエリ

Informatica ジョイン構文 [82](#)

Normal ジョイン構文 [82](#)

外部結合のサポート [81](#)

右外部結合構文 [86](#)

レフトアウタージョイン構文 [84](#)

監視

説明 [147](#)

き

キーリレーション

カスタマイズデータオブジェクト [67](#)

カスタマイズデータオブジェクトの作成 [71](#)

リレーショナルデータオブジェクト [64](#)

リレーショナルデータオブジェクト間に作成 [65](#)

く

クエリビュー

ヒントの設定 [77](#)

区切り記号付き識別子

データベース接続 [358](#)

区切りプロパティ

フラットファイルデータオブジェクト [101](#)

け

検証

エラーメッセージのグループ化 [146](#)

エラーメッセージの制限 [147](#)

プレファレンスの設定 [146](#)

こ

構成

トラブルシューティング [143](#)

高精度

Bigint データ型 [219](#)

Decimal データ型 [219](#)

固定幅プロパティ

フラットファイルデータオブジェクト [102](#)

コピー

オブジェクト [30](#)

説明 [30](#)

個別に選択

カスタマイズデータオブジェクト [78](#)

カスタマイズデータオブジェクトでの使用 [78](#)

し

識別子

区切り記号付き [358](#)

通常の [358](#)

識別リレーション

説明 [128](#)

シーケンシャル接続

プロパティ [344](#)

自己結合

カスタム SQL クエリ [75](#)

ジョイン構文

Informatica 構文 [82](#)

Normal ジョイン構文 [82](#)

カスタマイズデータオブジェクト [82](#)

右外部結合構文 [86](#)

レフトアウタージョイン構文 [84](#)

証明書

証明書の管理 [92](#)

証明書のプロパティ [92](#)

信頼されていない証明書 [92](#)

信頼されていない証明書の追加 [92](#)

せ

整数

計算での使用 [219](#)

式の定数 [220](#)

フラットファイルへの書き込み [221](#)

そ

ソースフィルタ

入力 [78](#)

ソート済みポート

カスタマイズデータオブジェクト [79](#)

カスタマイズデータオブジェクトでの使用 [79](#)

属性

リレーション [128](#)

ち

チートシート

説明 [21](#)

チームベース開発

トラブルシューティング [43](#)

つ

通常の識別子

データベース接続 [358](#)

て

データ型

Bigint [219](#)
DB2 for i5/OS [239](#)
DB2 for z/OS [239](#)
decimal [223](#)
double [223](#)
IBM DB2 [238](#)
Integer [219](#)
Microsoft SQL Server [242](#)
ODBC [247](#)
Oracle [248](#)
SAP HANA [251](#)
XML [253](#)
暗黙の変換 [255](#)
概要 [217](#)
トランスフォーメーション [218](#)
バイナリ [221](#)
Date/Time [221](#)
非リレーショナル [245](#)
ポート対ポートのデータ変換 [255](#)
文字列 [225](#)
複合ファイル [229](#)

データタイプ

JDBC [241](#)

データ統合サービス

選択 [24](#)

データの論理ビュー

開発 [119](#)

概要 [117](#)

データビューア

構成 [135](#), [139](#)

構成の作成 [140](#)

構成のトラブルシューティング [143](#)

構成プロパティ [135](#)

データベース

環境 SQL [59](#)

データベース接続

識別子のプロパティ [357](#)

データベースのヒント

Developer ツールでの入力 [77](#)

データ型リファレンス

フラットファイル [237](#)

デフォルトの SQL クエリ

表示 [87](#)

デプロイメント

アプリケーションアーカイブファイルへ [161](#), [162](#)

アプリケーションの更新 [156](#)

アプリケーションの置換 [156](#)

データ統合サービスへ [161](#), [162](#)

マッピングプロパティ [150](#)

概要 [148](#)

と

同期

カスタマイズデータオブジェクト [93](#)

物理データオブジェクト [93](#)

ドメイン

オブジェクトとプロパティの検索 [46](#)

説明 [25](#)

検索 [45](#)

追加 [23](#)

トラブルシューティング

バージョンニング [43](#)

トランザクション環境 SQL

設定 [59](#), [60](#)

トランスフォーメーションデータ型

リスト [218](#)

は

バージョンニング

トラブルシューティング [43](#)

倍精度浮動小数点数型のデータ型

概要 [223](#)

バイナリデータタイプ

概要 [221](#)

パフォーマンスのチューニング

Web サービス設定 [142](#)

データビューア構成の作成 [140](#)

データビューアの構成 [139](#)

マッピング構成 [141](#)

マッピング構成の作成 [142](#)

ひ

非識別リレーション

説明 [128](#)

ビジネス用語

ホットキーのカスタマイズ [48](#)

ルックアップ [47](#)

ビュー

[接続エクスプローラ] ビュー [52](#)

説明 [19](#)

非リレーショナルデータ操作

読み取り、書き込み、およびルックアップトランスフォーメーションの作成 [89](#)

ヒント

クエリビュー [77](#)

ふ

フォーマットのプロパティ

フラットファイルデータオブジェクト [100](#)

フォルダ

説明 [29](#)

フォルダー

作成 [29](#)

複合データ型

array [226](#)

map [227](#)

struct [228](#)

概要 [225](#)

プライマリキー

カスタマイズデータオブジェクトの作成 [71](#)

リレーショナルデータオブジェクトでの作成 [65](#)

フラットファイルデータオブジェクト

Excel での編集 [112](#)

Excel でのメタデータの設定 [112](#)

空として作成 [113](#)

カラムのプロパティ [98](#)

既存のフラットファイルからの作成 [114](#)

区切りプロパティ [101](#)

固定幅プロパティ [102](#)

詳細プロパティ [99](#)

制御ファイルからの作成 [116](#)

説明 [97](#)

全般プロパティ [98](#)

フォーマットのプロパティ [100](#)

フラットファイルデータオブジェクト (続く)

メタデータのコピー [112](#)

ランタイム書き込みプロパティ [105](#)

ランタイム読み取りプロパティ [103](#)

プロジェクト

共有 [25](#)

権限 [26](#)

権限の割り当て [28](#)

作成 [25](#)

説明 [25](#)

フィルタリング [26](#)

プロジェクトの権限

親オブジェクトのアクセスの許可 [28](#)

外部オブジェクトの権限 [27](#)

書き込み権限 [26](#)

権限の付与 [26](#)

セキュリティの詳細の表示 [28](#)

読み取り権限 [26](#)

割り当て [28](#)

依存オブジェクトインスタンス [27](#)

ブロックチェーン

接続プロパティ [272](#)

プロパティ [281](#), [283](#), [284](#), [334](#), [351](#)

ま

マッピング

構成 [135](#), [141](#)

構成のトラブルシューティング [143](#)

構成プロパティ [135](#)

デプロイメントのプロパティ [150](#)

構成の作成 [142](#)

マッピング実行前/実行後 SQL コマンド

カスタマイズデータオブジェクト [86](#)

リレーショナルデータオブジェクトへの追加 [87](#)

め

メタデータアクセスサービス

選択 [24](#)

メッセージング接続

Confluent Kafka 接続 [275](#)

Kafka 接続 [318](#)

も

文字列データタイプ

概要 [225](#)

モデルリポジトリ

オブジェクト [34](#)

オブジェクトとプロパティの検索 [46](#)

オブジェクトのチェックアウトとチェックイン [42](#)

接続 [43](#)

説明 [33](#)

チームベース開発 [38](#)

バージョンされたオブジェクト [39](#)

バージョン管理された [38](#)

バージョン管理されていない [38](#)

追加 [23](#)

モデルリポジトリサービス

更新 [44](#)

ゆ

ユーザー定義ジョイン

Informatica 構文 [82](#)

Normal ジョイン構文 [82](#)

外部結合のサポート [81](#)

カスタマイズデータオブジェクト [80](#)

入力 [81](#)

右外部結合構文 [86](#)

レフトアウタージョイン構文 [84](#)

よ

ようこそページ

説明 [21](#)

予約語ファイル

カスタマイズデータオブジェクト [75](#)

作成 [76](#)

ら

ランタイムアプリケーション

オブジェクトとプロパティの検索 [46](#)

ランタイム書き込みプロパティ

フラットファイルデータオブジェクト [105](#)

ランタイム読み取りプロパティ

フラットファイルデータオブジェクト [103](#)

り

リソースパラメータ

マッピングでのデプロイ [154](#)

リレーショナル接続

カスタマイズデータオブジェクトへの追加 [70](#)

リレーショナルデータオブジェクト

インポート [64](#)

カスタマイズデータオブジェクトへの追加 [70](#)

キーの作成 [65](#)

キーリレーション [64](#)

キーリレーションの作成 [65](#)

トラブルシューティング [96](#)

ワイルドカード検索 [64](#)

説明 [62](#)

る

ルックアップ

ビジネス用語集デスクトップ [47](#)

れ

例

array [226](#)

map [227](#)

struct [228](#)

ろ

ローカルワークスペースディレクトリ

設定 [18](#)

ログ

説明 [146](#)

論理データオブジェクト

説明 [127](#)

属性のリレーション [128](#)

プロパティ [127](#)

例 [118](#)

作成 [128](#)

論理データオブジェクトマッピング

書き込みマッピング [132](#)

タイプ [131](#)

読み取りマッピング [132](#)

作成 [132](#)

論理データオブジェクトモデル

インポート [120](#)

論理データオブジェクトモデル (続く)

例 [118](#)

作成 [120](#)

説明 [119](#)

わ

ワークスペースディレクトリ

設定 [18](#)

ワークベンチ

説明 [19](#)