



Informatica® Cloud Data Integration  
April 2024

# コンポーネント

Informatica Cloud Data Integration コンポーネント  
April 2024

© 著作権 Informatica LLC 2019, 2024

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（電子的複写、写真複写、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

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

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

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

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、[infa\\_documentation@informatica.com](mailto: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: 2024-06-17

# 目次

<b>序文</b> .....	7
Informatica のリソース.....	7
Informatica マニュアル.....	7
Informatica Intelligent Cloud Services Web サイト.....	7
Informatica Intelligent Cloud Services コミュニティ.....	7
Informatica Intelligent Cloud Services マーケットプレイス.....	8
データ統合のコネクタのドキュメント.....	8
Informatica ナレッジベース.....	8
Informatica Intelligent Cloud Services Trust Center.....	8
Informatica グローバルカスタマサポート.....	8
<b>第 1 章 : コンポーネント</b> .....	9
<b>第 2 章 : API コレクション</b> .....	11
API コレクションの作成.....	12
マッピングスキーマの表示.....	12
REST API 要求の同期.....	12
<b>第 3 章 : ビジネスサービス</b> .....	13
ビジネスサービスの定義.....	13
<b>第 4 章 : ファイルリスナ</b> .....	15
ファイル取り込みタスクのファイルリスナ.....	15
ファイルイベントの信頼性.....	16
ファイルリスナジョブの回復機能.....	17
タスクフローのファイルリスナ.....	17
B2B Gateway パートナーフローでのファイルリスナ.....	18
ファイルリスナの動作の違い.....	19
ファイルリスナの設定.....	19
サーバーソースタイプ用のファイルリスナの設定.....	21
コネクタソースタイプ用のファイルリスナの設定.....	22
パラメータの追加.....	25
手動によるファイルリスナの開始および停止.....	28
ファイルリスナの開始および停止.....	28
<b>第 5 章 : 固定長ファイル形式</b> .....	29
固定長ファイル形式の作成.....	29
<b>第 6 章 : 階層マッパー</b> .....	33
文.....	34

文のタイプ.....	35
文のプロパティ.....	37
階層マッパーの作成.....	38
業界標準メッセージのスキーマのマッピング.....	39
<b>第 7 章 : 階層スキーマ.....</b>	<b>40</b>
サンプルファイルまたはスキーマファイルの選択.....	40
階層スキーマの作成.....	41
<b>第 8 章 : 業種別データサービスカスタマイズ.....</b>	<b>42</b>
メッセージ定義.....	42
メッセージの構造.....	42
グローバル設定および位置設定.....	43
業界標準メッセージのカスタマイズ.....	43
メッセージの構造への要素の追加.....	44
要素のプロパティの編集.....	45
列挙の追加.....	45
メッセージ構造からの要素の削除.....	45
メッセージのプロパティ.....	45
HIPAA メッセージプロパティ.....	45
HL7 メッセージのプロパティ.....	47
<b>第 9 章 : インテリジェント構造モデル.....</b>	<b>50</b>
マッピングでのインテリジェント構造モデルの使用.....	51
詳細モードのマッピングでのインテリジェント構造モデルの使用.....	51
Data Engineering マッピングでのインテリジェント構造モデルの使用.....	52
B2B Gateway のインバウンドパートナーフローでのインテリジェント構造モデルの使用.....	52
インテリジェント構造検出のプロセス.....	53
インテリジェント構造モデルに対する入力.....	53
出力グループ定義.....	55
繰り返しグループ.....	55
プライマリキーおよび外部キー.....	57
データドリフト.....	58
未割り当てデータ.....	59
最大レコードサイズ.....	59
インテリジェント構造モデルの作成.....	59
インテリジェント構造モデルのエクスポート.....	61
インテリジェント構造モデル例.....	61
ユースケース.....	62
インテリジェント構造モデルのトラブルシューティング.....	64
<b>第 10 章 : インテリジェント構造モデルの改良.....</b>	<b>66</b>
インテリジェント構造モデルのビュー.....	67

ビジュアルモデルの操作. . . . .	68
リレーショナル出力の表示. . . . .	69
リレーショナル出力の操作. . . . .	70
構造内のノードの検索. . . . .	71
ノードデータの表示とアクションの実行. . . . .	71
ノードの編集. . . . .	72
繰り返しグループの操作. . . . .	74
モデル内のデータへのドキュメント識別子の追加. . . . .	75
フィールド名へのプレフィックスとサフィックスの追加. . . . .	75
複数ノードでのアクションの実行. . . . .	76
新しいサンプルでの既存のモデルのエンリッチ化. . . . .	77
Microsoft Excel 入力 of 構造の編集. . . . .	78
テーブルの転置. . . . .	79
テーブルと名前と値のペアの切り替え. . . . .	79
テーブルヘッダーの定義. . . . .	80
出力のテスト. . . . .	80
<b>第 11 章: マップレット. . . . .</b>	<b>81</b>
アクティブなマップレットとパッシブなマップレット. . . . .	81
マップレットの入力と出力. . . . .	82
マップレットの入力. . . . .	82
マップレットの出力. . . . .	82
トランスフォーメーションの名前. . . . .	83
マップレットのパラメータ. . . . .	83
マップレットの作成. . . . .	84
トランスフォーメーションパレットのフィルタリング. . . . .	85
マップレットの編集. . . . .	86
依存関係に影響する変更. . . . .	86
マップレットの同期. . . . .	86
データ分類. . . . .	87
データ分類の追加. . . . .	87
マップレットの検証. . . . .	87
PowerCenter マップレット. . . . .	88
アクティブおよびパッシブな PowerCenter マップレット. . . . .	88
マップレット内のストアードプロシージャ. . . . .	89
マップレットの PowerCenter XML ファイル. . . . .	89
データ統合タスクでの PowerCenter マップレット. . . . .	90
PowerCenter マップレットの設定. . . . .	91
<b>第 12 章: 保存済みクエリ. . . . .</b>	<b>92</b>
保存済みクエリ構文. . . . .	92
同期タスクでの保存済みクエリの使用. . . . .	94
SQL トランスフォーメーションでの保存済みクエリの使用. . . . .	94

保存済みクエリの設定.....	94
<b>第 13 章 : 共有シーケンス.....</b>	<b>96</b>
共有シーケンスのプロパティ.....	96
予約値の数.....	97
共有シーケンスの作成.....	98
共有シーケンスの使用.....	98
共有シーケンスのリセット.....	99
<b>第 14 章 : ユーザー定義関数.....</b>	<b>100</b>
ユーザー定義関数の作成.....	100
ユーザー定義関数の全般プロパティ.....	101
ユーザー定義関数の引数.....	101
ユーザー定義関数の式.....	102
ユーザー定義関数の編集および削除.....	102
ユーザー定義関数を使用した式の作成.....	102
ユーザー定義関数のパラメータ化.....	103
データ分類.....	104
データ分類の追加.....	104
<b>索引.....</b>	<b>106</b>

# 序文

コンポーネントを使用すると、Data Integration で作成できる再利用可能なアセットについて学ぶことができます。コンポーネントを作成して、トランスフォーメーション、マッピング、およびタスクに追加する方法をご覧ください。

## 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 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](mailto:infa_documentation@informatica.com)) までご連絡ください。

### Informatica Intelligent Cloud Services Web サイト

Informatica Intelligent Cloud Services Web サイト (<http://www.informatica.com/cloud>) にアクセスできます。このサイトには、Informatica Cloud 統合サービスに関する情報が含まれます。

### Informatica Intelligent Cloud Services コミュニティ

Informatica Intelligent Cloud Services コミュニティを使用して、技術的な問題について議論し、解決します。また、技術的なヒント、マニュアルの更新情報、FAQ（よくある質問）への答えを得ることもできます。

次の Informatica Intelligent Cloud Services コミュニティにアクセスします。

<https://network.informatica.com/community/informatica-network/products/cloud-integration>

開発者は、次の Cloud 開発者コミュニティで詳細情報を確認したり、ヒントを共有したりできます。

<https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers>

## Informatica Intelligent Cloud Services マーケットプレイス

Informatica マーケットプレイスにアクセスすると、データ統合コネクタ、テンプレート、およびマップレットを試用したり購入したりできます。

<https://marketplace.informatica.com/>

## データ統合のコネクタのドキュメント

データ統合のコネクタのドキュメントには、マニュアルポータルからアクセスできます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

## Informatica 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](mailto:KB_Feedback@informatica.com)) です。

## Informatica Intelligent Cloud Services Trust Center

Informatica Intelligent Cloud Services Trust Center は、Informatica のセキュリティポリシーおよびリアルタイムでのシステムの可用性について情報を提供します。

Trust Center (<https://www.informatica.com/trust-center.html>) にアクセスします。

Informatica Intelligent Cloud Services Trust Center にサブスクライブして、アップグレード、メンテナンス、およびインシデントの通知を受信します。[Informatica Intelligent Cloud Services Status](#) ページには、すべての Informatica Cloud 製品の実稼働ステータスが表示されます。メンテナンスの更新はすべてこのページに送信され、停止中は最新の情報が表示されます。更新と停止の通知がされるようにするには、Informatica Intelligent Cloud Services の 1 つのコンポーネントまたはすべてのコンポーネントについて更新の受信をサブスクライブします。すべてのコンポーネントにサブスクライブするのが、更新を逃さないようにするための最良の方法です。

サブスクライブするには、[Informatica Intelligent Cloud Services Status](#) ページで **【サブスクライブして更新】** をクリックします。電子メール、SMS テキストメッセージ、Webhook、RSS フィード、またはこの 4 つの任意に組み合わせとして送信される通知を受信するという選択ができます。

## Informatica グローバルカスタマサポート

グローバルサポートセンターには、Informatica Network または電話でお問い合わせください。

Informatica Network でオンラインサポートリソースを検索するには、Informatica Intelligent Cloud Services のヘルプメニューで **【サポートにお問い合わせください】** をクリックして、**Cloud Support** ページに移動します。**Cloud Support** ページには、システムステータス情報とコミュニティディスカッションが記載されています。追加のリソースを検索する場合や電子メールで Informatica グローバルカスタマサポートに問い合わせる場合は、Informatica Network にログインし、**【サポートが必要な場合】** をクリックしてください。

Informatica グローバルカスタマサポートの電話番号は、Informatica の Web サイト <https://www.informatica.com/services-and-training/support-services/contact-us.html> に掲載されていません。



# 第 1 章

## コンポーネント

コンポーネントはマッピングとタスクをサポートするアセットです。

次のコンポーネントを使用できます。

### API コレクション

機械学習トランスフォーメーションで使用する REST API 要求を作成します。

API コレクションの詳細については、[第 2 章, 「API コレクション」 \(ページ 11\)](#)を参照してください。

### ビジネスサービス

Web サービストランスフォーメーションで使用するビジネスサービスを定義します。

ビジネスサービスの詳細については、[第 3 章, 「ビジネスサービス」 \(ページ 13\)](#)を参照してください。

### ファイルリスナ

特定の場所にあるファイルをリスニングするファイルリスナを作成し、その場所にファイルが到着したとき、およびその場所でファイルが更新または削除されたときにサブスクリバに通知します。

ファイルリスナの詳細については、[第 4 章, 「ファイルリスナ」 \(ページ 15\)](#)を参照してください。

### 固定長ファイル形式

固定長フラットファイルのソースとターゲットに使用できる再利用可能な形式を設定して、可読性を向上します。

固定長ファイル形式の詳細については、[第 5 章, 「固定長ファイル形式」 \(ページ 29\)](#)を参照してください。

### 階層マッパー

階層スキーマ (業界標準メッセージのスキーマなど) を別の階層スキーマにマッピングします。

階層マッパーの詳細については、[第 6 章, 「階層マッパー」 \(ページ 33\)](#)を参照してください。

### 階層スキーマ

階層トランスフォーメーションが含まれるマッピングで使用する XML スキーマファイル、XML サンプルファイル、または JSON サンプルファイルをアップロードします。

階層スキーマの詳細については、[第 7 章, 「階層スキーマ」 \(ページ 40\)](#)を参照してください。

### 業種別データサービスカスタマイザ

業界標準メッセージをカスタマイズし、カスタムデータサービスとしてデータサービスリポジトリにパブリッシュします。

業種別データサービスカスタマイザの詳細については、[第 8 章, 「業種別データサービスカスタマイザ」 \(ページ 42\)](#)を参照してください。

## インテリジェント構造モデル

次のような場合にモデルを作成またはエクスポートします。

- モデルを作成し、そのモデルを構造パーサートランスフォーマーまたは階層ビルダトランスフォーマーで使用します。
- モデルを作成し、そのモデルを詳細モードで使用します。
- モデルをエクスポートし、Data Engineering Integration マッピングで使用する場合。
- モデルを作成し、それを使用して B2B Gateway でパートナーメッセージを処理する場合。

インテリジェント構造モデルの詳細については、[第 9 章, 「インテリジェント構造モデル」 \(ページ 50\)](#)を参照してください。

## マップレット

次のいずれかの方法で、マップレットトランスフォーマーで使用できる再利用可能なトランスフォーマーロジックを定義します。

- データ統合でマップレットを作成します。
- PowerCenter マップレット XML ファイルまたは SAP マップレットをアップロードして、そのトランスフォーマーロジックを同期タスクやマッピングで拡張します。

マップレットの詳細については、[第 11 章, 「マップレット」 \(ページ 81\)](#)を参照してください。

## 保存済みクエリ

同期タスクのソースとして使用できる再利用可能なカスタムクエリを作成します。

保存済みクエリの詳細については、[第 12 章, 「保存済みクエリ」 \(ページ 92\)](#)を参照してください。

## 共有シーケンス

再利用可能なシーケンスを定義し、複数のシーケンスジェネレータートランスフォーマーで使用します。

共有シーケンスの詳細については、[第 13 章, 「共有シーケンス」 \(ページ 96\)](#)を参照してください。

## ユーザー定義関数

トランスフォーマー式や、他のユーザー定義関数で使用できる再利用可能な関数を作成します。

ユーザー定義関数の詳細については、[第 14 章, 「ユーザー定義関数」 \(ページ 100\)](#)を参照してください。

## Visio テンプレート

Visio でトランスフォーマーロジックを設計して、テンプレートをアップロードし、マッピングタスクで使用するパラメータを設定します。

Visio テンプレートの詳細については、「マッピング」を参照してください。

**[新しいアセット]** ページでコンポーネントを作成します。

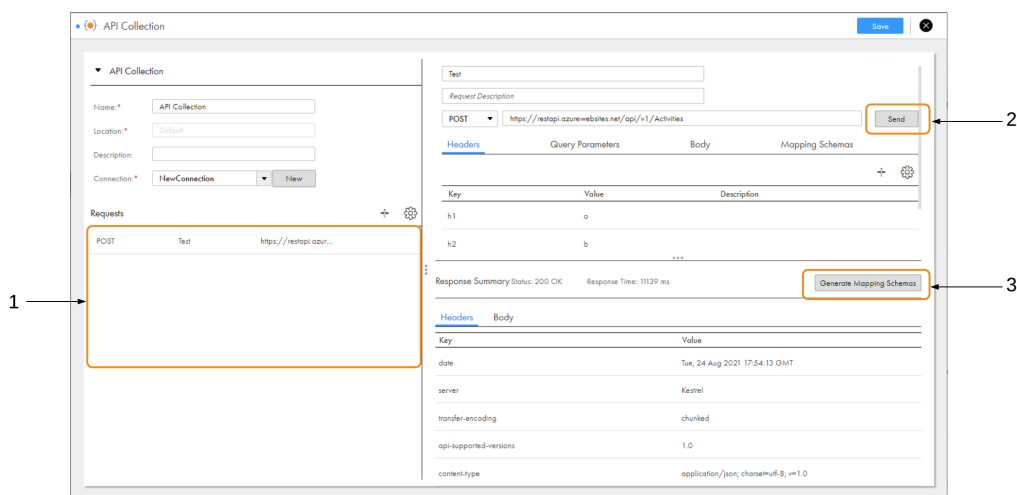
## 第 2 章

# API コレクション

機械学習トランスフォーメーションで使用する REST API 要求を格納する API コレクションを作成します。要求では、REST V3 接続を使用する必要があります。

API コレクションに新しい要求を追加するときは、要求の詳細を入力し、API をテストし、要求と応答を使用してマッピングスキーマを生成します。

次の画像は、API コレクションを示しています。



1. コレクションで利用可能な REST API 要求を確認します。
2. 要求をテストします。
3. 要求と応答に基づいてマッピングスキーマを生成します。

API コレクションは、トランスフォーメーションに対し次のマッピングスキーマを提供します。

- 要求スキーマ。トランスフォーメーションの受信フィールドを REST API の要求フィールドにマッピングするために使用されます。
- 応答スキーマ。REST API からの応答フィールドをトランスフォーメーションの出力フィールドにマッピングするために使用されます。

## API コレクションの作成

API コレクションを作成し、デフォルトの REST V3 接続を選択して要求をテストします。API コレクションをトランスフォーメーションに追加する際は、別の接続を使用できます。

要求を作成するには、API コレクションに新しい要求を追加し、要求の詳細を設定します。パスのパラメータをエンドポイント URL に追加します。次に、要求を送信してテストします。

API コレクションの要求が応答を受信した後は、マッピングスキーマを生成できます。API コレクションは、要求ソースを使用して要求スキーマを生成し、応答ソースを使用して応答スキーマを生成します。

サーバーレスランタイム環境で API コレクションを使用し、REST API を認証するように TLS を設定する場合は、詳細について管理者ヘルプを参照してください。非サーバーレスランタイム環境用に TLS を設定するには、Informatica グローバルカスタマサポートにお問い合わせください。

## マッピングスキーマの表示

マッピングスキーマは、**[マッピングスキーマ]** タブで表示できます。トランスフォーメーションは、マッピングスキーマを使用して REST API と対話します。

マッピングスキーマは空の構造を無視します。例えば、次の応答のアドレスオブジェクトは、応答スキーマでは無視されます。

```
{
  "ID":1,
  "address":{}
}
```

**注:** 応答スキーマには、親フィールドと同じ名前のネストされた子フィールドを含めることはできません。ネストされた子フィールドが親フィールドと名前を共有している場合、メタデータは機械学習トランスフォーメーションの応答フィールドに正しくプロパゲートされません。

## REST API 要求の同期

REST API 要求を同期して、機械学習トランスフォーメーションの要求名とマッピングスキーマを更新できます。

1. REST API 要求を使用する機械学習トランスフォーメーションを開きます。
2. **[モデル]** タブで、**[同期]** をクリックします。
3. トランスフォーメーションロジックでエラーを修正します。

エンドポイント URL やヘッダーパラメータなどの要求の詳細を更新するには、API コレクションに新しい要求を作成し、その新しい要求を機械学習トランスフォーメーションで使用します。

## 第 3 章

# ビジネスサービス

ビジネスサービスは、操作が設定されている Web サービスです。Mapping Designer でビジネスサービスを定義し、操作を Web サービストランスフォーメーションに追加します。

ビジネスサービスを定義し、プロジェクトフォルダに保存します。ビジネスサービス定義は、複数のマッピングで使用できます。

## ビジネスサービスの定義

ビジネスサービスを定義するには、次の手順を実行します。

1. **【新規】** > **【コンポーネント】** > **【ビジネスサービス】** をクリックし、**【作成】** をクリックします。
2. ビジネスサービスの名前を入力し、保存する場所を選択します。
3. 使用する接続を選択するか、新しい接続を作成します。
4. 必要に応じて、タスクを実行するたびに接続メタデータを更新する場合は動的更新を有効にします。
5. Web サービスから使用する操作を選択します。
6. 必要に応じて、要求と応答の choice 要素と派生型要素を指定するように操作を設定します。

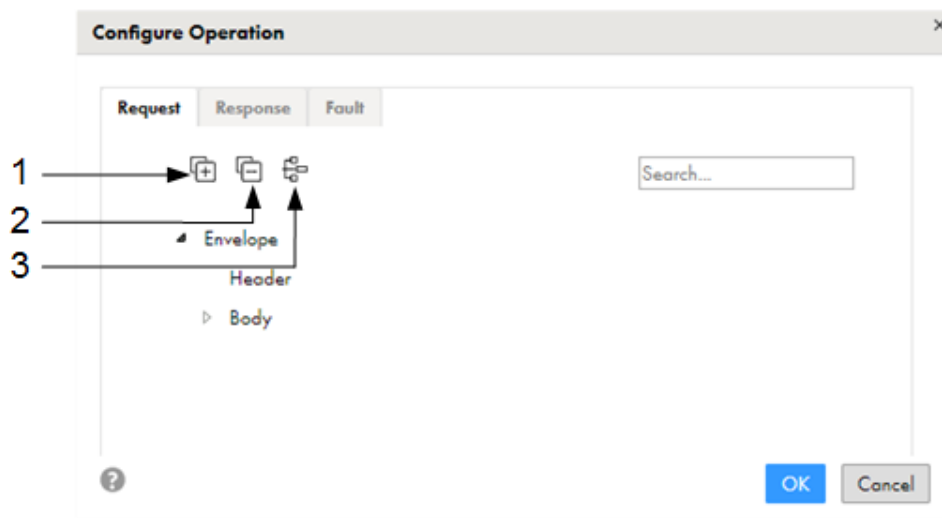
操作コンポーネントに choice 要素または complexType 要素が含まれていて、抽象属性が true である場合は、操作マッピングを設定するときに 1 つ以上の要素または派生型を選択する必要があります。

必要に応じて、抽象属性が false である complexType 要素の場合は、complexType 要素に派生型を選択することもできます。

- a. 設定する操作の **【設定】** をクリックします。
- b. **【操作の設定】** ウィンドウで、**【要求】**、**【応答】**、**【障害】** のいずれかのタブをクリックし、設定する必要があるノードに移動します。

**注:** WSDL で anyAttribute 要素を使用している場合、要求または応答に要素は表示されません。

上部のアイコンをクリックして、設定する必要があるノードに移動できます。



1. すべてのノードを展開。
  2. すべてのノードを縮小。
  3. 設定する必要があるノードのみを表示。このアイコンを選択してノードが展開されない場合、choice 要素または派生型を選択する必要はありません。
- c. choice 要素または派生型を選択します。  
関連するタブをクリックして、操作の要求と応答を設定してください。
- d. 設定した操作を保存します。
7. 必要に応じて、さらに操作を追加し、操作を設定します。
8. ビジネスサービスを保存します。  
設定していない必須 choice 要素と派生型がある場合は、ビジネスサービスを保存できません。

## 第 4 章

# ファイルリスナ

ファイルリスナは、定義した場所にあるファイルをリスニングします。タスクフロー、ファイル取り込みタスク、および B2B Gateway パートナーフローによって、ファイルリスナを使用して特定のフォルダを監視し、ファイルイベントの発生時にコールバック API を介して通知を受信します。監視対象のフォルダに新しいファイルが到着した場合や監視対象のフォルダ内のファイルが更新または削除された場合にファイルイベントが発生します。

特定のフォルダおよびファイルパターンをリスニングするファイルリスナを定義します。次のアセットに対して、通知をトリガするファイルイベントを定義できます。

- タスクフロー
- ファイル取り込みタスク
- B2B Gateway パートナー

**注:** ファイルリスナをアセットに割り当てるには、ファイルリスナの読み取り権限とファイルリスナの実行権限が必要です。

例えば、新しいファイルが到着した場合、または監視対象の場所にあるファイルが更新または削除された場合に、ファイル取り込みタスクに通知するかどうかを定義できます。次に、ファイル取り込みタスクにファイルリスナを割り当てます。ファイルリスナは、ソースフォルダをリスニングし、ファイルの到着イベント、更新イベント、または削除イベントの発生時にファイル取り込みタスクに通知します。ソースは、サーバーイベントまたはコネクタとして設定できます。次に、ファイル取り込みタスクが実行され、ソースフォルダからファイルが取得されます。

ファイルリスナは、フォルダのリスニングを開始するとき、フォルダのリスニングを終了するとき、フォルダのリスニング中にエラーが発生したときに、ファイルの作成者であるユーザーに通知を送信します。

ファイルリスナは、ファイルリスナの開始時にジョブを作成し、Monitor のファイル転送ログページにジョブインスタンスを一覧表示します。ファイルリスナは、ファイルイベントの発生時にジョブを更新します。ファイルリスナジョブの詳細は、ファイルリスナジョブのプロパティに表示されます。

タスクフロー、ファイル取り込みタスク、または B2B Gateway パートナーによって使用されていないファイルリスナは、非アクティブであり、定義済みフォルダをリスニングしません。

## ファイル取り込みタスクのファイルリスナ

ファイルリスナをソースとして使用して、ファイル取り込みタスクでファイルモニタをスケジュールできます。

次のソースタイプを使用したファイル取り込みタスクでは、ファイルリスナから通知を受け取ったときに実行するタスクをスケジュール設定できます。

- ローカルフォルダ

- Advanced FTP V2
- Advanced SFTP V2
- Advanced FTPS V2
- Amazon S3 V2
- Microsoft Azure Data Lake Store Gen2
- Microsoft Azure Data Lake Store V3
- Microsoft Azure Blob Storage V3
- Microsoft Fabric OneLake
- Google Cloud Storage V2
- Hadoop Distributed File Storage (HDFS) V2

**注:** ファイル取り込みタスクの設定に関する詳細については、一括取り込みのヘルプを参照してください。

## ファイルイベントの信頼性

ファイル取り込みタスクでファイルリスナをソースとして使用している場合、新しいファイルが到着したとき、既存のファイルが更新されたとき、またはファイルが削除されたときに、ファイルリスナの構成に基づいてファイルイベントが作成されます。ファイルイベントは、ファイル取り込みタスクに渡されます。このセクションでは、ファイルリスナとファイル取り込みタスクの間でこれらのファイルイベントを処理する際の信頼性の側面について説明します。

ファイルリスナは、次の条件に基づいてイベントを処理します。

- Secure Agent が実行されていないか、一時的なネットワークの中断があり、ファイルイベントがファイル取り込みタスクに到達しない場合、ファイルリスナは各ファイルのイベントをキューに登録し、それを次のファイル取り込みジョブの通知に含めます。つまり、ファイル取り込みタスクは、各ファイルに関する通知を少なくとも 1 回受信します。これにより、ファイルリスナとファイル取り込みタスクの間で少なくとも 1 回の信頼性が保証されます。

**注:** 処理されないファイルイベントは、7 日間キューに残ります。

- 複数のイベントが発生した場合、ファイルリスナは、各ファイルの最後のイベントのみをファイル取り込みタスクに通知します。
- ファイルリスナキューにあるファイルイベントは、次のいずれかの方法でファイル取り込みタスクに到達します。
  - ファイル取り込みジョブが完了すると、一括取り込みサービスがファイルリスナにプル要求を行い、キューに登録されたイベントを確認します。イベントが見つかると、サービスはそれらを処理するために新しい取り込みジョブをトリガします。プル要求は、同じ取り込みタスクによって実行される別の同時実行ジョブにすでに割り当てられているファイルの処理をトリガしないため、常に 1 つの取り込みジョブのみがファイルを処理します。
  - プロアクティブなプル要求によってイベントが取得されない場合、例えば、Secure Agent が実行されていない状態で一括取り込みサービスが要求を行った場合、ファイルリスナは各ファイルの最後のイベントをキューに登録し、それを次のファイル取り込みジョブの通知に含めます。
  - ファイル取り込みタスクを手動で実行して、失敗したイベントを取得することもできます。
- ファイルイベントの処理が失敗すると、ファイル取り込みタスクは失敗したイベントの処理を再試行します。失敗したイベントの再試行は、自動的に 1 回行われ、その後のファイルリスナ通知中に行われます。



- ファイル取り込みタスクは、ステータスが成功または重複のファイルイベントを自動的に再処理しません。転送ログなどを使用した場合など、エラーが原因でターゲットに正常に転送されなかったファイルについては、手動で特定する必要があります。この問題を解決するには、ファイルを移動するか、手動で変更して、ファイルリスナがファイルを取得できるようにします。例えば、ファイルの最終変更時刻が変更された場合、ファイルリスナは、内容が変更されていない場合でも、ファイルが更新されたと見なします。

#### 例

ファイルリスナは、ターゲットに転送する 15 件のファイルイベントを持つファイル取り込みタスクのソースです。バッチサイズは 5 です。ファイル取り込みタスクがトリガされて完了すると、ファイルイベントは次のステータスになります。

- 1 番目のバッチ（ファイル 1 から 5）の 5 件のイベント: 成功
- 2 番目のバッチ（ファイル 6 から 10）の 5 件のイベント: 失敗
- 3 番目のバッチ（ファイル 11 から 15）の 5 件のイベント: 未処理

ファイル取り込みタスクは、一度に 5 件の失敗したイベントと未処理のイベントの処理を自動的に再試行します。ファイル取り込みタスクが完了すると、ファイルイベントは次のステータスになります。

- 1 番目のバッチ（ファイル 6 から 10）の 5 件のイベント: 成功
- 2 番目のバッチ（ファイル 11 から 15）の 5 件のイベント: 失敗

ファイル取り込みタスクは、一度に 5 件の失敗したイベントの処理を自動的に再試行します。ファイル取り込みタスクが完了すると、2 番目のバッチ（ファイル 11 から 15）の 5 件のイベントが失敗します。

ファイル取り込みタスクを手動で実行して、保留中の 5 件のイベントを取得できます。ファイル取り込みタスクを手動で実行しない場合、ファイルリスナは次のファイル取り込みジョブの通知に、失敗したイベントを含めます。

## ファイルリスナジョブの回復機能

ファイルリスナを実行すると、対応するジョブが作成されます。

ファイルリスナジョブは、次の条件に基づいて Secure Agent の可用性を処理します。

- 新しいバージョンの一括取り込みアプリケーションがリリースされると、ファイルリスナは停止し、新しいバージョンで実行を再開します。
- Secure Agent が使用できない状態になって 20 分以内にバックアップされた場合、ファイルリスナは再起動後に同じ Secure Agent 上で実行を再開します。
- Secure Agent が 20 分間使用できない場合、ファイルリスナは Secure Agent グループ内で使用可能な実行中の Secure Agent 上で再起動します。この動作は、ファイルリスナが終了時刻に達していない場合に適用されます。
- Secure Agent グループ内で Secure Agent が 20 分以上使用できない場合、ファイルリスナは応答しない状態のままになります。200 分後、ステータスは **【停止】** に変わります。

## タスクフローのファイルリスナ

ファイルリスナが接続をリスニングするように設定されている場合、タスクフローでファイルリスナを使用できます。

次下のユースケースのタスクフローでファイルリスナを使用できます。

### ファイルリスナ経由でタスクフローを呼び出す手順

コネクタソースタイプを使用して、ファイルリスナ経由でタスクフローを呼び出すことができます。

タスクフロー内で、バインディングタイプとして**イベント**を定義し、イベントソースとしてファイルリスナを選択します。タスクフローをパブリッシュするときに、タスクフローは其中で定義されているファイルリスナにサブスクライブします。ファイルイベントが発生すると、ファイルリスナはタスクフローを呼び出します。

例えば、フォルダ上の新しいファイルをリスンするようにファイルリスナを設定した場合、ファイルリスナは指定したフォルダに新しいファイルが到着するたびに、関連するタスクフローを呼び出します。

### ファイルイベント経由でタスクフロー実行を調整する手順

タスクフローの File Watch タスクステップを使用して、ファイルイベント経由でタスクフロー実行を調整できます。

File Watch タスクステップをタスクフローに追加して、定義された場所にあるファイルをリスニングし、ファイルイベントを監視できます。File Watch タスクステップでは、コネクタソースタイプを使用して既存のファイルリスナを選択できます。ファイルイベントを使用して、タスクフロー実行を調整できます。

例えば、ファイルが特定の場所に到着するのを待ってから、次のステップでファイルを使用できます。

## B2B Gateway パートナーフローでのファイルリスナ

ファイルリスナを使用して、B2B Gateway パートナーのインバウンドフローとアウトバウンドフローをトリガできます。

パートナーを設定する際は、インバウンドフローとアウトバウンドフローをトリガするためのファイルリスナを選択します。パートナーを保存する際、パートナーはファイルリスナにサブスクライブします。ファイルリスナは、ファイルリスナで定義されたルールに従ってフローをトリガします。

例えば、フォルダへの新しいファイルの到着をリスニングするようにファイルリスナを設定してから、インバウンドフローにファイルリスナを使用するようにパートナーを設定します。指定したフォルダにパートナーがファイルを置くと、ファイルリスナはインバウンドフローをトリガします。

## ファイルリスナの動作の違い

ファイル取り込みタスクと B2B Gateway パートナーフローおよびタスクフローの File Watch タスクステップでファイルリスナが使用される場合、特定の異なる動作が発生します。

次の表で、ファイルリスナの動作の違いを、使用される場所に基づいて説明します。

動作	ファイル取り込みタスクと B2B Gateway パートナーフローで使用されるファイルリスナ	タスクフローの File Watch タスクステップで使用されるファイルリスナ
ライフサイクル	ファイルリスナは、ファイルイベントが最初に発生するまで、または設定された終了時刻まで実行されます。	ファイルリスナは、ファイルイベントが最初に発生するまで、またはタイムアウトが発生するまで実行されます。ファイルイベントが発生しない場合、デフォルトでは、タスクフローは 5 分間、または File Watch Task ステップの Time Out 入力フィールドで定義されたオーバーライドされた値の期間だけ待ちます。タスクフローが待機する最大期間は 7 日で、その後タイムアウトになります。
実行の終了時刻	実行の最後の終了時刻は、設定された終了日時のタイムゾーンの午後 11 時 55 分で、翌日に延長することはできません。	ファイルリスナは、ファイルイベントまたはタイムアウトが発生するまで実行され、終了日やタイムゾーンに依存しません。
スナップショット	<ul style="list-style-type: none"> <li>- すべてのファイルリスナインスタンスが同じスナップショットを共有します。</li> <li>- スナップショットが削除されることはありません。</li> <li>- 実行されたファイルリスナは、Monitor の【ファイル転送ログ】タブに、ファイルリスナ名をインスタンス名として一覧表示されます。</li> </ul> 例えば、ファイルリスナ名が FL_Delete の場合、Monitor の【ファイル転送ログ】タブで確認が必要なインスタンス名は FL_Delete になります。	<ul style="list-style-type: none"> <li>- 各ファイルリスナインスタンスは、独自のスナップショットを維持します。</li> <li>- スナップショットは、ジョブが完了した直後に削除されます。</li> <li>- 実行された各ファイルリスナは、Monitor の【ファイル転送ログ】タブに一覧表示されます。</li> </ul> 出力フィールドに表示される monitorJobId をファイルリスナ名に付加して、Monitor の【ファイル転送ログ】タブでインスタンス名を見つけます。 例えば、monitorJobId が 7500 でファイルリスナの名前が FL_Arrive の場合、Monitor の【ファイル転送ログ】タブで確認が必要なインスタンス名は FL_Arrive-7500 になります。
起動と停止	データ統合から、またはファイル転送 REST API リソースを使用して、ファイルリスナインスタンスを起動および停止できます。	File Watch タスクステップを含むタスクフローを実行すると、関連するファイルリスナインスタンスが起動します。ファイルリスナインスタンスは、ファイルイベントまたはタイムアウトが最初に発生したときに停止します。ファイルリスナインスタンスを手動で起動または停止しても、タスクフローは影響を受けません。

## ファイルリスナの設定

ファイルリスナを作成する前に、次の条件を確認します。

- 組織が適切なライセンスを所有している。
- ファイルリスナが Secure Agent 上で実行中である。

ファイルリスナを設定するには、次の手順を実行します。

1. ファイルリスナを作成するには、**[新規]** > **[コンポーネント]** > **[ファイルリスナ]** をクリックし、**[作成]** をクリックします。

ファイルリスナを編集するには、**[エクスプローラ]** ページでファイルリスナに移動します。ファイルリスナをリストする行で、**[アクション]** をクリックし、**[編集]** を選択します。

ランタイム環境、コネクタタイプ、接続、またはリスナルールのプロパティの更新などのファイルリスナを編集する場合、ファイルリスナをリセットする必要があります。警告メッセージが表示されます。続行するには、**[OK]** をクリックします。このアクションにより、以前に追跡を行ったファイルのリストがクリアされます。

2. 次のファイルリスナの詳細を設定します。

パラメータ	説明
ファイルリスナの名前	ファイルリスナの名前。ファイルリスナの名前は組織内で一意である必要があります。ファイルリスナの名前には、英数字、スペース、アンダースコアを含めることができます。名前の先頭には英字またはアンダースコアを指定する必要があります。ファイルリスナの名前の大文字と小文字は区別されません。
場所	ファイルリスナが作成されるプロジェクトディレクトリ。
説明	ファイルリスナの説明（省略可能）。最大長は 1024 文字です。
ステータス	ファイルリスナのステータス（有効または無効）。無効なファイルリスナは、指定したフォルダでファイルをリスニングしません。
ランタイム環境	ファイルリスナの実行に使用される Secure Agent を含むランタイム環境。
ソースタイプ	ファイルリスナがリスニングするソースのタイプ。次のうち 1 つのソースタイプを選択します。 <ul style="list-style-type: none"> <li>- <b>サーバー</b>。ファイルリスナはサーバーをリスニングします。サーバーに設定するパラメータとファイルリスナルールの詳細については、<a href="#">「サーバーソースタイプ用のファイルリスナの設定」(ページ 21)</a>を参照してください。</li> <li>- <b>コネクタ</b>。ファイルリスナは接続をリスニングします。コネクタに設定するパラメータとファイルリスナルールの詳細については、<a href="#">「コネクタソースタイプ用のファイルリスナの設定」(ページ 22)</a>を参照してください。</li> </ul>

3. ファイルリスナを実行するスケジュールを設定します。

パラメータ	説明
実行	ファイルリスナを実行する頻度（毎日、毎週、または毎月）。
開始日	ファイルリスナの実行を開始する日付。
終了日	ファイルリスナの実行を終了する日付。
無期限に繰り返す	ファイルリスナは無期限に実行されます。

パラメータ	説明
開始	ファイルリスナが実行を開始する時刻。
次まで確認	ファイルリスナが実行を終了する時刻。
次ごとに確認	ファイルリスナが、リスニング対象フォルダ内のファイルを確認する頻度（秒単位、分単位、または時間単位）。
タイムゾーン	ファイルリスナを実行する基準となるタイムゾーン。
実行する日数	ファイルリスナを毎週実行することを選択した場合は、ファイルリスナを実行する曜日。
月の日付	ファイルリスナを毎月実行することを選択した場合は、ファイルリスナを実行する月の日付。
曜日	ファイルリスナを毎月実行することを選択した場合は、ファイルリスナを実行する月の曜日。
週	ファイルリスナを毎月実行することを選択した場合は、ファイルリスナを実行する週。例えば、次の第4日曜日にファイルリスナを実行する場合は、 <b>【曜日】</b> に <b>【日曜日】</b> 、 <b>【週】</b> に <b>【第4週】</b> を選択します。

4. **【障害管理】**では、次のオプションを選択できます。
  - **障害時に実行を続行。**ファイルリスナは、一時的なネットワークの中断などの障害が発生した場合でも再試行して実行を続けます。
  - **障害時に通知を送信。**ファイルリスナに障害が発生した場合に通知を受け取ります。ファイルリスナに障害が発生した場合に通知を送信する電子メールアドレスのリストを入力します。電子メールアドレスのリストを区切るには、カンマを使用します。
5. **【保存】**をクリックします。

## サーバーソースタイプ用のファイルリスナの設定

ファイルリスナの詳細で、サーバーとしてソースタイプを選択する際、ファイルリスナはサーバーのイベントをリスニングします。

ファイルリスナは、AS2サーバー、HTTPSサーバー、SFTPサーバーでイベントをリスンするように設定できます。例えば、AS2サーバーがファイルを受信するときに通知を送信するようにファイルリスナを設定できます。ファイルリスナ詳細の設定に関する詳細については、[「ファイルリスナの設定」 \(ページ 19\)](#)を参照してください。

サーバーソースタイプを選択する際は、次のパラメータとファイルリスナルールを設定します。

1. 次のパラメータを設定して、ファイルリスナがリスニングするイベントを定義します。

パラメータ	説明
イベントプロバイダ	ファイルリスナがリスニングするサーバータイプ。ファイルリスナは、AS2 サーバー、HTTPS サーバー、または SFTP サーバーをリスンするように設定できます。
イベントのタイプ	ファイルリスナがリスニングするサーバーイベントのタイプ。AS2 サーバーに定義できるイベントタイプは次のとおりです。 <ul style="list-style-type: none"><li>- AS2 メッセージの受信失敗</li><li>- AS2 メッセージの受信成功</li></ul> HTTPS サーバーに定義できるイベントタイプは次のとおりです。 <ul style="list-style-type: none"><li>- HTTPS のアップロード失敗</li><li>- HTTPS のアップロード成功</li></ul> SFTP サーバーに定義できるイベントタイプは次のとおりです。 <ul style="list-style-type: none"><li>- SFTP のアップロード失敗</li><li>- SFTP のアップロード成功</li></ul> 例えば、イベントタイプが「SFTP のアップロード失敗」の場合、ファイルリスナは、SFTP サーバーでファイルのアップロードが失敗したときに通知を送信します。

2. 次のパラメータを設定して、リスナルールを定義します。

パラメータ	説明
キー	イベントの属性。
タイプ	キー値をフィルタする方法。 <ul style="list-style-type: none"><li>- [次を含む]。値を含むキーをフィルタ処理します。</li><li>- [次の値に等しい]。完全一致を見つけてルールを実行します。</li></ul>
値	キーの値。

## コネクタソースタイプ用のファイルリスナの設定

ファイルリスナの詳細で、コネクタとしてソースタイプを選択する際、ファイルリスナは接続をリスニングします。

ファイルリスナ詳細の設定に関する詳細については、「[ファイルリスナの設定](#)」(ページ 19)を参照してください。

1. 次のパラメータを設定して、接続を定義します。

パラメータ	説明
接続タイプ	ファイルリスナがリスニングする接続のタイプ。
接続	ファイルリスナがリスニングする接続。

2. 次のパラメータを設定して、リスナルールを定義します。

パラメータ	説明
フォルダパス	<p>ファイルリスナがリスニングするフォルダへのパス。</p> <p><b>注:</b> ファイルリスナは、NFS および CIFS をサポートするネットワーク共有上のファイルおよびディレクトリにアクセスできます。</p> <p>ソースファイルシステムへの相対パスを入力できます。相対パスを入力するには、パスをピリオドで開始し、その後にスラッシュを配置します (./)。パスは、接続で指定されたソースディレクトリの相対パスです。</p>
パターンタイプ	<p>ファイルリスナがリスニングするファイル名のパターンを決定します。次のいずれかのパターンを選択します。</p> <ul style="list-style-type: none"> <li>- ワイルドカード</li> <li>- Regex</li> <li>- インジケータファイル</li> </ul>
ファイルパターン	<p>ファイルリスナがリスニングするファイル名パターン。</p> <ul style="list-style-type: none"> <li>- ワイルドカード: ファイル名のワイルドカードパターンを使用します。</li> <li>- Regex: ファイル名パターンと一致する正規表現を使用します。 正規表現のパターンタイプを選択する場合は、次の例を検討します。 <ul style="list-style-type: none"> <li>- 次の構文を使用すると、名前に out、foo、baz を含むファイルを除くすべてのファイルをリスニングします。<code>^(?!.*(?:out baz foo)).*\$ all except</code></li> <li>- 次の構文を使用すると、拡張子が doc、docx、pdf のすべてのファイルをリスニングします。<code>([a-zA-Z0-9\s_\.\\-\/\(\):])+(\.doc .docx .pdf)\$</code></li> <li>- 次の構文を使用すると、名前に out.txt を含むファイルを除くすべてのテキストファイルをリスニングします。<code>^(?!out).\*.txt\$</code></li> </ul> </li> <li>- インジケータファイル。ファイルリスナがリスニングするファイル名を使用します。</li> </ul>
サブフォルダ内のファイルを確認	<p>ファイルリスナが、リスニング対象フォルダにあるサブフォルダ内のファイルを確認するかどうかを示します。</p>
アクション後	<p>ファイルリスナがイベントをリスニングした後にファイルリスナが実行する必要のあるアクションを決定します。</p> <p>ファイルパターンがインジケータファイルである場合にのみ、アクション後に <b>【削除】</b> を選択することができます。デフォルトは <b>【なし】</b> です。</p> <p>次の接続タイプは、[アクション後] のオプションをサポートします。</p> <ul style="list-style-type: none"> <li>- ローカルフォルダ</li> <li>- Advanced FTP V2</li> <li>- Advanced FTPS V2</li> <li>- Advanced SFTP V2</li> <li>- Amazon S3 V2</li> <li>- Microsoft Azure Data Lake Store Gen2</li> <li>- Microsoft Fabric OneLake</li> </ul>
パラメータの追加	<p>式を作成して、フォルダパスのパラメータとして追加します。詳細については、<a href="#">「パラメータの追加」</a> (ページ 25) を参照してください。</p>

パラメータ	説明
ファイル到着時に	<p>ファイルリスナが、登録済みのサービスに通知を送信するタイミングを決定します。</p> <ul style="list-style-type: none"> <li>- 到着時。ファイルリスナがリスニングするフォルダにファイルが到着したときに通知を送信します。</li> <li>- 更新時。ファイルリスナがリスニングするフォルダのファイルが更新されたときに通知を送信します。</li> <li>- 削除時。ファイルリスナがリスニングするフォルダのファイルが削除されたときに通知を送信します。</li> </ul> <p>必要に応じて複数のオプションを選択できます。</p> <p><b>注:</b> <b>【アクション後】</b> が <b>【削除】</b> の場合は、<b>【削除時】</b> オプションを選択しないでください。</p>
ルールを満たした場合はチェックを停止	<p>ファイルリスナは、リスナルルールに合致するとフォルダのリスニングを停止します。例えば、リスニング対象フォルダ内のファイルが削除されたときに通知を送信するようにファイルリスナを設定している場合、フォルダ内で最初にファイル削除イベントが発生するとリスナはフォルダのリスニングを停止します。</p> <p>このオプションが選択されない場合、ファイルリスナはイベントを登録済みのアプリケーションに通知し、後続のイベントを引き続きリスニングします。</p>
ファイルの安定性を確認	<p>ファイルリスナは、登録済みのサービスに通知する前に、ファイル全体がリスニング対象フォルダにコピーされたかどうかを確認します。</p> <p><b>ヒント:</b> サイズの大きいファイルを転送する場合（フォルダへのファイルの書き込み処理に時間がかかる場合）は、このオプションを選択します。</p>
安定性確認間隔	<p>ファイルリスナがファイルの安定性を確認するために待機する時間（秒単位）。例えば、安定時間が 15 秒の場合、ファイルリスナは 15 秒ごとにファイルのステータスを確認します。</p> <p>安定性確認間隔フィールドは、<b>【ファイルの安定性を確認】</b> オプションを選択した場合にのみ有効になります。</p> <p>安定性確認間隔は、10 秒から 300 秒の範囲です。デフォルトは 10 秒です。</p>
最初の実行時にファイルが存在するかどうかを通知	<p>ファイルリスナを初めて実行すると、リスニングするフォルダにファイルが存在する場合に通知が送信されます。このオプションを選択すると、<b>【ファイルリスナが実行中でない場合にイベントを除外】</b> オプションが無効になります。</p>
ファイルリスナが実行中でない場合にイベントを除外	<p>ファイルリスナが実行されていないときに発生するファイルイベントを除外するかどうかを決定します。このオプションを選択すると、<b>【最初の実行時にファイルが存在するかどうかを通知】</b> オプションが無効になります。</p>

3. スケジュールを指定する場合は、次のオプションを検討してください。
  1.
    - スケジュールオプションは、ファイルリスナを毎日、毎週、または毎月実行するように設定できます。
    - スケジュールオプションは、特定のタイムゾーンに基づいて特定の日時（および秒）にファイルリスナを開始し、ジョブインスタンスを繰り返し実行するように設定できます。
  4. **【障害管理】** で **【障害時に実行を続行】** を選択し、タスクがエラーで失敗してもファイルリスナを実行するようにします。**【障害時に通知を送信】** を選択して、タスクが失敗した場合には通知を受け取るようにします。タスクに障害が発生した場合に通知を送信する電子メールアドレスのリストを入力します。



## パラメータの追加

変数を使用して、ファイルリスナが読み書きを行うパラメータを設定できます。


次のタイプの変数のいずれかを使用して、パラメータを設定できます。

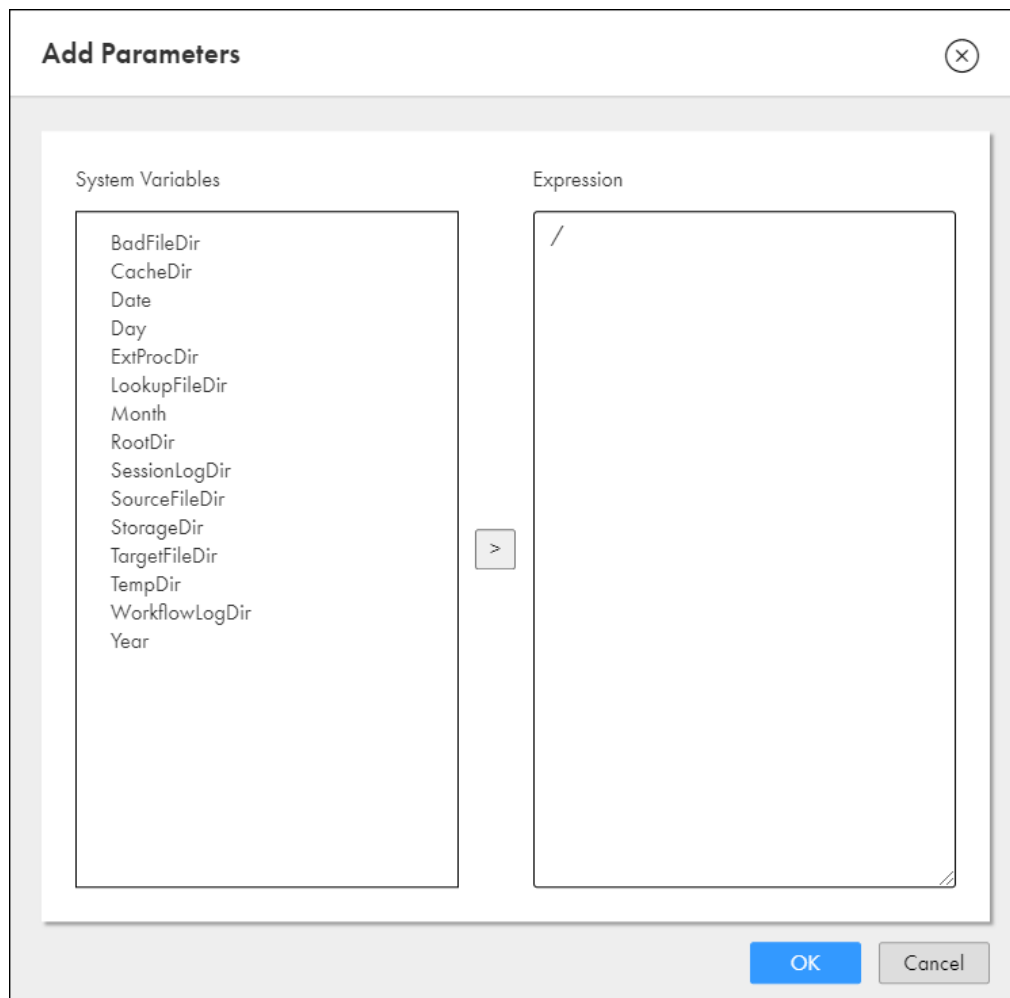
- システム変数
- ユーザー定義変数


**注:** ユーザー定義変数を使用して、ユーザーインターフェースからタスクを実行することはできません。ユーザー定義変数の値は、Mass Ingestion Files REST API のジョブリソースを使用して渡す必要があります。詳細については、REST API のヘルプを参照してください。

### システム変数を使用したパラメータの追加

システム変数を使用して、タスクウィザードで **[フォルダパス]** にパラメータを追加します。

1.  ファイルリスナタスクウィザードの **[アクション後]** の横にあるアイコンをクリックします。  
**[パラメータの追加]** ウィンドウが表示されます。



2. **[システム変数]** カラムで必要な変数を選択し、 をクリックします。選択したシステム変数が **[式]** カラムに表示されます。この手順を繰り返して、複数のシステム変数を選択します。

**注:** タスク内でシステム変数を使用する場合は、`${systemvariablename}`という形式で指定する必要があります。

以下の表に、システム変数を示します。

システム変数	説明	式
BadFileDir *	リジェクトファイルのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	<code>`\${PMBadFileDir}</code>
CacheFileDir *	キャッシュファイルの場所。	<code>`\${PMCacheDir}</code>
Date **	ISO (yyyy-MM-dd) 形式の現在の日付。	<code>`\${system.date}</code>
Day **	曜日	<code>`\${system.day}</code>
ExtProcDir *	外部プロシージャのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	<code>`\${PMExtProcDir}</code>
LookupFileDir *	ルックアップファイルのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	<code>`\${PMLookupFileDir}</code>
Month **	月 (数値)	<code>`\${system.month}</code>
RootDir	ノードによるルートディレクトリへのアクセス性。これは他のサービスプロセス変数のルートディレクトリです。次の特殊文字は使用できません。 * ? < > "   ,	<code>`\${PMRootDir}</code>
SessionLogDir *	セッションログのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	<code>`\${PMSessionLogDir}</code>

システム変数	説明	式
StorageDir *	実行時ファイルのディレクトリ。ワークフローリカバリファイルは、PowerCenter 統合サービスプロパティで設定された \$PMStorageDir に保存されます。セッションリカバリファイルは、オペレーティングシステムプロファイルで設定された \$PMStorageDir に保存されます。次の特殊文字は使用できません。 * ? < > "   ,	\${PMStorageDir}
TargetFileDir *	ターゲットファイルのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	\${PMTargetFileDir}
TempDir *	一時ファイルのディレクトリ。次の特殊文字は使用できません。 * ? < > "   ,	\${PMTempDir}
WorkflowLogDir *	ワークフローログファイルの場所。	\${PMWorkflowLogDir}
Year **	年	\${system.year}
<p>* 値はデータ統合サーバーから取得されます。</p> <p>** タイムゾーンは Secure Agent のタイムゾーンです。</p>		

**注:** タスクを開始するときにパラメータの置換が発生します。ファイルリスナが起動して、パラメータ値が異なるために以前監視されていた場所とは異なる場所であると評価された場合は、以前監視されていた場所に存在するファイルに関する削除通知が送信されます。通知の削除オプションを無効にするか、**【ファイルリスナが実行中でない場合にイベントを除外】** オプションを有効にすると、通知は送信されません。

3. **【OK】** をクリックします。

式が **【フォルダパス】** フィールドに表示されます。

## ユーザー定義変数を使用したパラメータの追加

ユーザー定義変数を使用して、タスクウィザードで **【フォルダパス】** の値を定義します。

1. **【フォルダパス】** フィールドをクリックして、変数を入力します。変数は \${systemvariablename} という形式で指定する必要があります。

Listener Rules*				
Folder Path	Pattern Type	File Pattern	Check For Files In Sub-Folders	Post Action
/root/\${session}	Wildcard	*.txt	<input type="checkbox"/>	None

2. **【OK】** をクリックします。

式が **【フォルダパス】** に表示されます。

# 手動によるファイルリスナの開始および停止

ファイルリスナは、実行スケジュールに基づいて、定義した頻度で実行されます。事前定義したスケジュールでファイルリスナを実行しない場合は、手動でファイルリスナを開始または停止できます。

ファイルリスナの実行時間としてスケジュール設定した開始時刻から終了時刻までの間、ファイルリスナは定義した場所でファイルをリスニングします。

ファイルリスナを手動で開始および停止する場合は、ファイルリスナの次の動作を考慮します。

- ファイルリスナを、スケジュール済みの実行より前に開始した場合を考えます。手動でトリガされたファイルリスナは、スケジュール済みの実行の開始時刻に実行および停止します。
- ファイルリスナを、スケジュール済みの実行の終了時刻より後に開始した場合を考えます。ファイルリスナは1回実行して停止します。
- ファイルリスナが実行スケジュールに基づいてすでに実行されている場合、ファイルリスナを開始する事は出来ません。
- エージェントのバージョンがアップグレードされると、ファイルリスナは自動的に新しいエージェントのバージョンに送信され、古いバージョンは停止します。
- 一時的なネットワークの中断など、エージェントが失敗した場合、エージェントが再起動すると、ファイルリスナが自動的にエージェントに送信されます。エージェントが失敗したときに送信された Arrive、Update、Delete などのイベントは、エージェントの再起動時にファイルリスナから通知されます。
- 実行中の既存のファイルリスナを停止した場合を考えます。ファイルリスナは、その実行に対してファイルのリスニングを停止します。ファイルリスナが手動で開始されない限り、次のスケジュール時刻に新しい実行が開始されます。
- ファイルリスナが無効になっている場合、タスクに関連付けられていない場合、または関連付けられている Secure Agent がダウンしている場合、ファイルリスナは起動に失敗します。
- 到着、更新、削除などのファイルイベントの数がイベントの最大制限（10,000 ファイル）を超えると、ファイルリスナの起動が失敗します。

## ファイルリスナの開始および停止

**【アクション】** メニューを使用して、ファイルリスナコンポーネントの **【編集】** ページまたは **【参照】** ページからファイルリスナを開始および停止できます。ファイルリスナを手動で開始または停止するには、次の手順を実行します。

1. **【参照】** ページで、1つ以上のファイルリスナを選択し、**【アクション】** メニューをクリックします。すべてのファイルリスナを開始する場合は、**【参照】** ページでファイルリスナアセットをフィルタリングし、**【すべて選択】** をクリックします。  
**【開始】** ボタンと **【停止】** ボタンが有効になります。  
ファイルリスナを開始に失敗した場合は、失敗の詳細を示すメッセージが表示されます。
2. **【開始】** をクリックします。  
ファイルリスナが設定を検証し、ファイルリスナの設定ルールで定義されているファイルのリスニングを開始します。
3. **【停止】** をクリックしてファイルリスナを停止します。すべてのファイルリスナを停止する場合は、**【参照】** ページでファイルリスナアセットをフィルタリングし、**【すべて選択】** をクリックします。ファイルリスナの停止に失敗した場合は、失敗の詳細を示すメッセージが表示されます。

## 第 5 章

# 固定長ファイル形式

固定長フラットファイルの形式の詳細情報を指定する固定長ファイル形式を作成して保存できます。

固定長フラットファイルは、マッピングとマッピングタスクのソースまたはターゲットとして使用できます。ソーストランスフォーメーションまたはターゲットトランスフォーメーションのフラットファイル接続を選択するときに、フラットファイルタイプを指定します。固定長フラットファイルタイプを選択する場合は、固定長フラットファイルのデータに基づいて、使用に最適な固定長ファイル形式を選択します。

複数の固定長ファイル形式を作成できます。例えば、四半期ごとの売上データを含む固定長フラットファイルと、インベントリデータを含む別の固定長フラットファイルに固定長ファイル形式を使用します。

固定長ファイル形式がまだ存在しない場合は、固定長フラットファイルを使用するマッピングまたはマッピングタスクを作成する前に、固定長ファイル形式を作成する必要があります。

固定長ファイル形式を設定するには、カラム数と各カラムの幅、名前、データ型を指定します。また、固定長形式の詳細プロパティを設定することもできます。例えば、NULL 文字を処理する方法を指定できます。また、カラムごとにデフォルトの日付形式を指定できます。

使用しない固定長ファイル形式を削除できます。マッピングまたはマッピングタスクで使用されている固定長ファイル形式は削除できません。

## 固定長ファイル形式の作成

マッピング内のソースまたはターゲットとして固定長フラットファイルを使用できるように、固定長ファイル形式を作成します。

固定長ファイル形式の形式オプションを指定する場合は、サンプルフラットファイルを使用します。サンプルファイル内のデータがページに表示され、データに適した形式オプションの指定に役立ちます。指定した形式オプションによって、サンプルファイルが変更または保存されることはありません。

1. **【新規】** > **【コンポーネント】** > **【固定長ファイル形式】** をクリックしてから、**【作成】** をクリックします。

固定長ファイル形式を編集するには、**【エクスプローラ】** ページで固定長ファイル形式に移動します。固定長ファイル形式が含まれる行で、**【アクション】** をクリックして **【編集】** を選択します。

2. 以下の固定長ファイル形式の詳細を入力します。

プロパティ	説明
設定名	固定長ファイル形式の名前。
場所	固定長ファイル形式の場所。固定長ファイル形式を保存するフォルダを参照するか、デフォルトの場所を使用します。 [Explore (参照)] ページが現在アクティブになっていて、プロジェクトまたはフォルダが選択されている場合、アセットのデフォルトの場所はその選択されているプロジェクトまたはフォルダです。そうでない場合、デフォルトの場所は直近で保存されたアセットの場所です。
説明	固定長ファイル形式の説明。

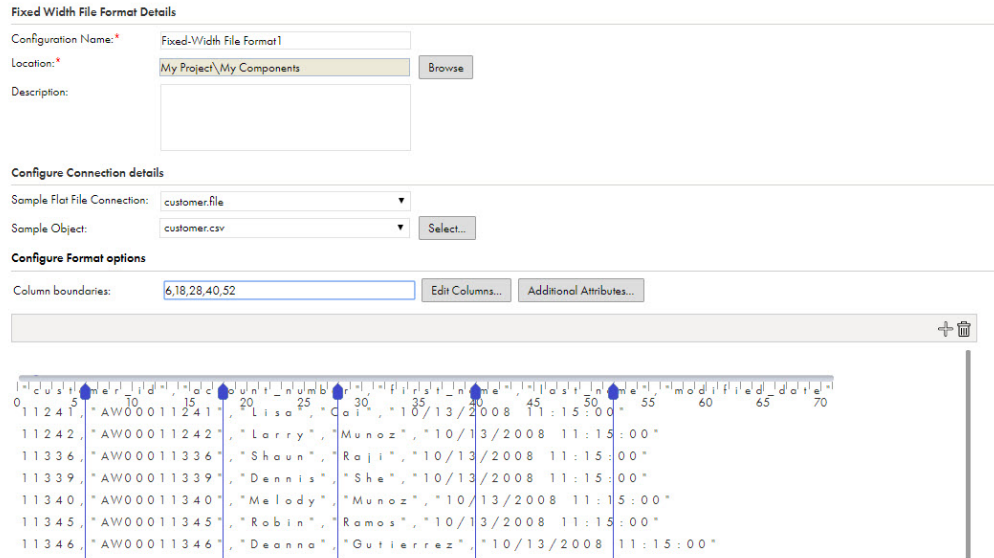
3. サンプルのフラットファイルについて、次の接続の詳細を指定します。

プロパティ	説明
フラットファイルの接続サンプル	固定長形式の指定に使用するサンプルファイルへの接続。
サンプルオブジェクト	形式オプションの指定に使用するサンプルファイル。

4. サンプルファイル内のデータに合うように、カラムの境界を設定します。[カラムの境界] フィールドに境界を入力するか、ルーラーを使用して境界を配置することができます。

- [カラムの境界] フィールドでカラムの境界を設定するには、バイト数を入力して、最初のカラムの幅を指定します。次に、それ以降のカラムの境界を入力します。各値はカンマで区切ります。境界を入力した後に Enter を押し、ルーラーの境界の配置を確認します。
- ルーラーを使用してカラムの境界を設定するには、マウスを使用して境界を最初のカラムのルーラーの適切な場所までドラッグします。ルーラー上の各マークは 1 バイトを表します。新しい境界を追加するには、ルーラーまたは [追加] アイコンをクリックします。必要に応じて、マウスを使用して境界をルーラーの適切な場所までドラッグします。

次の図は、6つのコラムに対して境界が設定された固定長ファイル形式を示しています。



5. カラム名を追加または変更し、データ型を編集するには、**カラムの編集**をクリックしてから、カラム名を入力し、各カラムのデータ型を選択します。
6. 詳細プロパティを指定するには、**[追加属性]**をクリックして、次のプロパティを指定します。

プロパティ	説明
改行コードあり	各行を改行文字で終了します。フラットファイルのソースまたはターゲットオブジェクトで固定長ファイル形式を使用する場合は、有効になっている必要があります。[改行コードあり]はデフォルトで有効になっています。
スキップする行数	スキップする先頭の行数。例えば、空白行またはヘッダー行をスキップする場合です。
カラム終端後にスキップするバイト数	行の最後のカラムと次の行の最初のカラムの間のバイト数。
NULL 文字タイプ	NULL 文字がテキストまたは ASCII コードのどちらかです。
NULL 文字	NULL 値を表す文字。
NULL 文字を繰り返す	単一フィールドの連続した NULL 文字を単一の NULL 値として読み込みます。

プロパティ	説明
後続のスペースを削除	末尾の空白を文字列値から削除します。
デフォルトの日付形式	<p>日付形式がフラットファイル接続で指定されていない場合にカラムで使用する日付形式。デフォルトの日付形式を指定するには、次の情報を入力します。</p> <ul style="list-style-type: none"> <li>- フィールド長を調整するか固定するか。幅を調整するには A を入力し、幅を固定するには F を入力して、その後 2 つのスペースを入力します。</li> <li>- フィールド長 (バイト単位)。</li> <li>- 日付形式。</li> </ul> <p>例えば、幅を 12 バイトの固定長にして日付形式を DD/MM/YYYY にする場合は、次のテキストを入力します。</p> <p>F 12 DD/MM/YYYY</p>

詳細プロパティを保存するには、**[OK]** をクリックします。

7. 固定長ファイル形式を保存するには、**[保存]** をクリックします。



## 第 6 章

# 階層マッパー

階層マッパーは、階層スキーマ（業界標準メッセージのスキーマなど）を別の階層スキーママッピングします。例えば、パーサーデータサービスの XML 出力をターゲット XSD ファイルにマッピングできます。

階層マッパーは、ソーススキーマとターゲットスキーマをロードします。文を使用して、ソーススキーマのソース要素をターゲットスキーマのターゲット要素にリンクします。文を生成するには、ソース要素をターゲット要素にドラッグするか、文を手動で作成します。文では、データを階層入力から階層出力に変換する方法を定義し、条件とフィルタを定義します。

例えば、顧客の請求書データを XML 形式で表すソーススキーマをロードし、それを、XML 形式で顧客の注文を表すターゲットスキーマにマッピングできます。文を使用して、顧客の注文を月ごとにグループ化し、連絡先情報と注文合計をマッピングできます。

データサービストランスフォーメーションで階層マッパーを使用して、階層入力を処理し、XML ドキュメントを別の XML ドキュメントに変換したり、JSON ドキュメントを別の JSON ドキュメントに変換したりするなど、異なる階層構造を使用する階層出力に変換できます。

次の図は、階層マッパーを示しています。

1 → Hierarchical Mapper

2 → Statements

Row	Name	Statement Type	Input	Condition	Skip Miss...	Default	On Fail	Output	Mede	Comments
1	Candidates to Depa...	Group	.		<input checked="" type="checkbox"/>		skip	.	Match or Acid	
2	Candidate to Depar...	Group	Candidate[1]		<input checked="" type="checkbox"/>		skip	Department	Match or Acid	
3	Profile to Profile	Group	Profile		<input checked="" type="checkbox"/>		skip	Employee/Profile	Add	
4	Id to Id	Map	Id		<input checked="" type="checkbox"/>		skip	Id	Match or Acid	
5	FirstName to FullN...	Map	concat(FirstName, "...		<input checked="" type="checkbox"/>		skip	FullName	Match or Acid	
6	Skills to Skills	Group	Skills		<input checked="" type="checkbox"/>		skip	Skills	Match or Acid	

1. 階層マッパー。マッパーには、ソーススキーマとターゲットスキーマが表示されます。ソース要素をターゲット要素にドラッグすると、文を生成できます。
2. 文。文のプロパティを編集するか、新しい文を追加します。

階層マッパーを使用するには、次のタスクを実行します。

1. XML ドキュメントや JSON ドキュメントなどの階層ドキュメントに基づいて階層スキーマを作成します。詳細については、「[第 7 章、「階層スキーマ」 \(ページ 40\)](#)」を参照してください。
2. 階層マッパーを作成して、ソーススキーマをターゲットスキーマにマッピングします。
3. マッピングを作成し、データサービストランスフォーメーションで階層マッパーを選択して、階層入力を処理し、階層出力に変換します。マッピング作成の詳細については、「[マッピング](#)」を参照してください。データサービストランスフォーメーションの詳細については、「[トランスフォーメーション](#)」を参照してください。

## 文

文は、ソーススキーマの階層入力をターゲットスキーマの階層出力にマッピングする方法を定義します。ソース要素をターゲット要素にドラッグすると、階層マッパーによって編集可能な文が生成されます。

ソース要素ごとに文を作成することも、文をネストして他の文に依存させることもできます。文は、子の文のグループの親になることができ、その場合は親の文が実行されるたびに子の文も実行されます。子の文は、親の文の下にインデントされて表示されます。

文を作成するには、ソース要素をターゲット要素にドラッグします。階層マッパーは、文のタイプ、入力、および出力を識別し、文を生成します。また、文を手動で追加して設定することもできます。

## 文のタイプ

文の各タイプは、ソーススキーマとターゲットスキーマの間でマッピングされる要素のタイプを表します。Router 文の場合、文のタイプは、要素をマッピングするために実行するアクションを表します。

次のタイプの文を定義できます。

### マップ

Map 文は、単純なソース要素を単純なターゲット要素にリンクします。

### グループ

Group 文は、複合ソース要素を複合ターゲット要素にリンクします。

### 繰り返しグループ

Repeating Group 文は、繰り返し複合ソース要素を繰り返し複合ターゲット要素にリンクします。

### ルーター

Router 文は、条件を使用して各ソース要素を評価し、入力値に基づいて処理します。

### オプション

Option 文は、Router 文で条件を定義します。条件が true と評価された場合、階層マッパーは Option 文を実行します。

### デフォルト

Default 文は、Option 文がどれも適用されない場合に Router 文で実行する Default 文を定義します。

## Map 文

Map 文は、単純なソース要素を単純なターゲット要素にリンクします。入力は単一の値か定数値である必要があります。Map 文は、Group 文、Repeating Group 文、または Option 文の子にすることができます。

Map 文は、ソースの name 要素とターゲットの name 要素など、2つの要素間の単純なマッピングにすることも、式を使用する複雑なマッピングにすることもできます。例えば、ソーススキーマの Id、FirstName、および LastName 要素を、ターゲットスキーマの Name 属性にマッピングするとします。入力式 `concat(Profile/Id, "-", Profile/FirstName, "-", Profile/LastName)` と出力式 `@Name` を使用して、Map 文を設定できます。

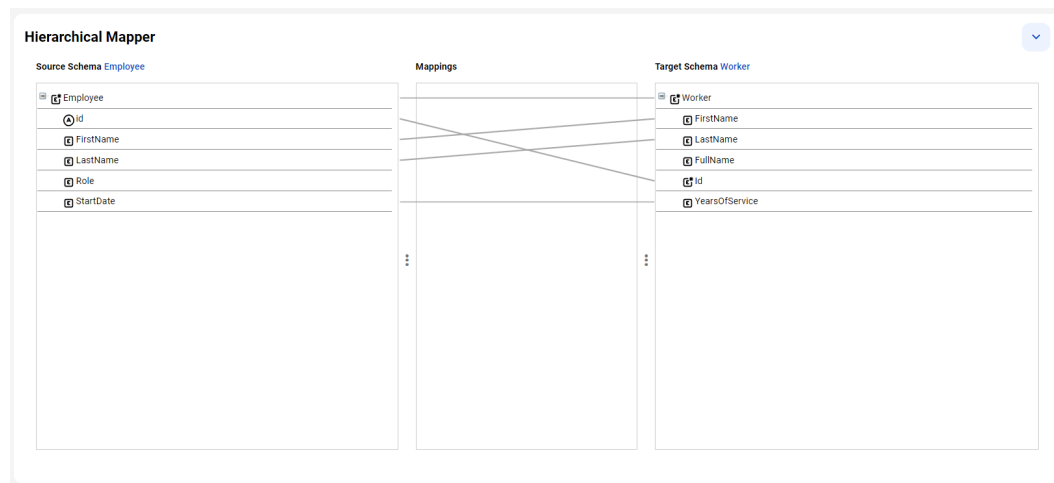
## Group 文

Group 文は、複合ソース要素を複合ターゲット要素にリンクします。他のタイプの文を Group 文の子としてネストできます。

Group 文を使用すると、すべての子の文が同時に合格または不合格になる共通のコンテキストを提供できます。また、Group 文を使用して、他の文を整理したり簡略化したりすることもできます。

例えば、ソーススキーマの Employee 親要素をターゲットスキーマの Worker 親要素にマッピングし、両方の親要素に FirstName や LastName などネストになった子要素が含まれているとします。Group 文を作成して Employee 親要素を Worker 親要素にマッピングし、ネストになった Map 文を作成して Employee 親要素の子要素を Worker 親要素の子要素にマッピングできます。

次の図は、階層マッパーで要素がどのようにマッピングされるかを示しています。



次の図は、Group 文とネストになった Map 文を示しています。

Row	Name	Statement Type	Input	Condition	Skip Missin...	Default	On Fail	Output	Mode	Comments
1	Employee to Worker	Group	.		<input checked="" type="checkbox"/>		Propagate	.	Match or A...	
2	@id to Id	Map	@id		<input checked="" type="checkbox"/>		Propagate	Id	Match or A...	
3	FirstName to FirstName	Map	FirstName		<input checked="" type="checkbox"/>		skip	FirstName	Match or A...	
4	LastName to LastName	Map	LastName		<input checked="" type="checkbox"/>		skip	LastName	Match or A...	
5	StartDate to YearsOfService	Map	year-from-date(C...		<input checked="" type="checkbox"/>		skip	YearsOfService	Match or A...	

## 繰り返しグループ文

Repeating Group 文は、繰り返し複合ソース要素を繰り返し複合ターゲット要素にリンクします。Repeating Group 文は、ソーススキーマにソース要素が登場するたびに実行されます。

Repeating Group 文には、反復した Map 文が含まれています。Repeating Group への入力、要素または値のシーケンスとして評価される式です。

## Router 文、Option 文、Default 文

Router 文は、条件を使用して各ソース要素を評価し、入力値に基づいて処理します。

Router 文は Option 文を 1 つ以上含み、Default 文を 1 つ含むことができます。階層マッパーは、ソース要素を処理するときに、入力値と一致する条件を持つ Option 文のみを実行します。どの条件も入力値と一致しない場合、階層マッパーは Default 文を実行します。

例えば、組織の従業員を雇用しようとしていて、そのスキルを従業員プロファイルの報酬スコアにマッピングするとします。Router 文を Option 文と共に使用して、候補者がスキル A またはスキル B のどちらを持っているかを評価し、適切な報酬スコアを割り当てることができます。候補者がどちらのスキルも持っていない場合は、Default 文が実行されて報酬スコアが割り当てられます。

次の図は、Router 文とネストになった Option 文および Default 文を示しています。

Row	Name	Statement Type	Input	Condition	Skip Missing ...	Default	On Fail	Output	Mode	Comments
8	Skills to Profile	Router	Skills		<input checked="" type="checkbox"/>		skip	.	Match or Add	
9	Skill is A	Option	Skill	normalize-s...	<input type="checkbox"/>		skip	CompensationScore	Match or Add	
10		Map	10		<input type="checkbox"/>		skip	.	Match or Add	
11	Skill is B	Option	Skill	normalize-s...	<input type="checkbox"/>		skip	CompensationScore	Match or Add	
12	default	default			<input type="checkbox"/>		skip	CompensationScore	Match or Add	

## 文のプロパティ

階層マッパーは、文の生成時に文のプロパティを自動的に設定しますが、ユーザーが文のプロパティを編集したり、文を手動で構成したりすることもできます。

行を追加または削除するには、【設定】ボタンを使用します。文字列を使用して文を検索することもできます。文の行を選択すると、対応するマッピングが【階層マッパー】領域で強調表示されます。同様に、階層マッパーで要素を選択すると、対応する文が【文】領域で強調表示されます。

以下の表に、文のプロパティを示します。

プロパティ	説明
行	文を識別する行番号。行番号は自動的に生成されます。
名前	文を識別するための名前。この名前はいつでも変更でき、文の名前は一意である必要はありません。文の名前を使用すると、セッションログで文を見つけやすくなります。
文のタイプ	ソース要素がソーススキーマに出現した場合の文の実行方法を定義する型。
入力	ソーススキーマの入力要素を定義する XPath または XPath で構成される式。式は、要素または値として評価できます。
条件	ソース要素をターゲット要素にマップする条件を定義する XPath または XPath で構成される式。 例: <code>normalize-space(string(.)) = 'A'</code>
欠落している入力をスキップ	入力式がソーススキーマの要素と一致しない場合にマッピング文をスキップするかどうかを決定します。次のいずれかのオプションを選択します。 - 有効: ソース要素が存在しない場合、階層マッパーはエラーなしで文をスキップします。 - 無効: ソース要素が存在しない場合、文は失敗します。
デフォルト	デフォルト値は、階層マッパーがソース要素を見つけられない場合のターゲット要素の値を設定します。
失敗時	文が失敗した場合の対処方法を決定します。次のいずれかのオプションを選択します。 - 文のスキップ。文が失敗した場合、文をスキップします。 - 反復のスキップ。文が失敗し、それがグループの一部である場合は、文のすべての反復をスキップします。 - プロパゲート。文が失敗すると、親の文も失敗します。
出力	ターゲットスキーマの出力要素を定義する XPath または XPath で構成される式。

プロパティ	説明
モード	階層マッパーが出力要素をターゲットスキーマに追加するか、出力式の値をターゲットスキーマの既存の要素と一致させるかを決定します。次のいずれかのオプションを選択します。 <ul style="list-style-type: none"> <li>- 追加。ターゲットスキーマに要素を作成します。その要素が複数出現する要素ではなく、ターゲットスキーマにすでに存在する場合、文は失敗します。</li> <li>- 一致。出力式の値をターゲットスキーマの既存の要素と照合します。要素が存在しない場合、文は失敗します。</li> <li>- 一致または追加。出力式の値がターゲットスキーマに存在する場合、階層マッパーは出力をターゲット要素と照合します。要素がターゲットスキーマに存在しない場合、階層マッパーはターゲットスキーマに要素を作成します。</li> </ul>
コメント	文についてのコメント。

## 階層マッパーの作成

階層マッパーを作成して、階層スキーマ（業界標準メッセージのスキーマなど）を別の階層スキーマにマッピングします。

1. **【新規】** > **【コンポーネント】** > **【階層マッパー】** をクリックし、**【作成】** をクリックします。
2. **【階層マッパー】** デザイナの **【全般】** ページでプロパティを設定します。

次の表は、全般プロパティの説明です。

プロパティ	説明
名前	階層マッパーの名前。
説明	階層マッパーの説明。
プロジェクト	階層マッパーを保存するプロジェクトフォルダ。

3. **Next** をクリックします
4. **【階層マッパー】** ページで、**【ソーススキーマの選択】** をクリックします。
  - a. **【ソーススキーマの選択】** ダイアログボックスで、ソースタイプを選択します。
  - b. **【参照】** をクリックして、デザイナーにロードするソーススキーマを参照し、**【選択】** をクリックします。
5. **【ターゲットスキーマの選択】** をクリックします。
  - a. **【ターゲットスキーマの選択】** ダイアログボックスで、ターゲットタイプを選択します。
  - b. **【参照】** をクリックして、デザイナーにロードするターゲットスキーマを参照し、**【選択】** をクリックします。
6. マッピング文を生成するには、以下のいずれかのオプションを使用します。
  - ソース要素をターゲット要素にドラッグすると、文が自動的に生成されます。
  - 文のリストまで下にスクロールし、**【追加】** ボタンをクリックして新しい文を追加し、文のプロパティを設定します。
7. 文を設定したら、**【保存】** をクリックします。

# 業界標準メッセージのスキーマのマッピング

業界標準メッセージのスキーマをマッピングし、メッセージを別の階層構造に変換するには、データサービスリポジトリからデータサービスをダウンロードし、データ統合でスキーマを階層スキーマとしてアップロードします。その後、階層マッパーで階層スキーマを使用し、マッピングで階層マッパーを使用できます。

1. Administrator で、データサービスをダウンロードして解凍します。
  - a. **【データサービスリポジトリ】** ページを開きます。
  - b. データサービスを含む行を選択し、**【ダウンロード】** アイコンをクリックして、ZIP ファイルをローカルマシンに保存します。
  - c. ファイルを一意的な名前のフォルダに解凍します。
2. データ統合で、ダウンロードしたデータサービスをベースとして階層スキーマを作成します。
  - a. **【新規】 > 【コンポーネント】 > 【階層スキーマ】** をクリックします。
  - b. **【新規階層スキーマ】** ページで、階層スキーマの名前と場所を指定するか、デフォルト値を使用します。
  - c. **【アップロード】** をクリックします。
  - d. **【スキーマ/サンプルファイルをアップロード】** ダイアログボックスで、**【ファイルの選択】** をクリックして、データサービスのスキーマを含む XSD ファイルを選択し、**【OK】** をクリックしてアップロードします。
  - e. 階層スキーマを保存するには、**【保存】** をクリックします。
3. 階層スキーマをソーススキーマとして使用して、階層マッパーを作成します。
  - a. **【新規】 > 【コンポーネント】 > 【階層マッパー】** をクリックします。
  - b. **階層マッパーデザイナー**で、全般プロパティを設定し、ソーススキーマとして作成した階層スキーマを選択します。次に、それをターゲット階層スキーマにマッピングします。  
階層マッパーの作成の詳細については、[「階層マッパーの作成」 \(ページ 38\)](#)を参照してください。
4. 階層マッパーを使用するには、以下のマッピングを次の順序で作成して実行します。
  1. データサービストランスフォーメーションを使用して受信した業界標準メッセージを解析し、XML ファイルを生成するパーサーデータサービスを呼び出すマッピング。
  2. データサービストランスフォーメーションを使用して、生成された XML ファイルに対して階層マッパーを実行し、階層マッパーでターゲットスキーマとして定義された XML ファイルを出力するマッピング。

## 第 7 章

# 階層スキーマ

階層スキーマは、予想されるデータの階層を定義するアセットです。スキーマは、データ統合でアップロードするサンプルファイルまたはスキーマファイルをベースとしています。

階層スキーマは、階層ビルダトランスフォーメーションまたは階層パーサートランスフォーメーションで使用できます。階層スキーマは、スタンドアロンアセットとして作成することも、トランスフォーメーション内で作成することもできます。スキーマをスタンドアロンアセットとして作成する場合でも、トランスフォーメーション内で作成する場合でも、任意のトランスフォーメーションに関連付けることができます。

階層スキーマを作成した後、それを編集してルートまたはスキーマ定義を変更できます。階層スキーマがトランスフォーメーションで使用されている場合、そのスキーマを編集または削除することはできません。

## サンプルファイルまたはスキーマファイルの選択

階層スキーマを作成するときは、サンプルファイルまたはスキーマファイルをベースとして作成します。精度を高めるには、可能な限りサンプルファイルではなくスキーマファイルを使用します。

### スキーマファイル

スキーマファイルを使用する場合は、次のルールとガイドライを考慮してください。

- スキーマに再帰要素を含めることはできません。
- 各スキーマファイルには、最大 10,000 個の要素を含めることができます。10,000 を超える要素を含むスキーマを使用するには、スキーマを複数のファイルに分割します。
- 他のスキーマを参照するスキーマを使用するには、各ファイルを個別にアップロードするか、ZIP ファイルにまとめてアップロードします。ZIP ファイルに XSD ファイル以外のファイルが含まれている場合、データ統合では追加のファイルが無視され、XSD ファイルのみを使用して階層スキーマが生成されます。
- 同じ名前のスキーマファイルを複数アップロードする必要がある場合は、ZIP ファイルでアップロードします。それらを個別にアップロードすると、データ統合によって重複ファイルの名前が変更されます。

### サンプルファイル

サンプルファイルを使用する場合は、すべての可能なフィールド、値とデータ型のすべての順序、およびマッピングが処理するデータの長さを表す値など、処理する予定のデータを表していることを確認してください。



# 階層スキーマの作成

階層スキーマを作成するには、全般プロパティを定義し、サンプルファイルまたはスキーマファイルをアップロードします。

1. **【新規】** > **【コンポーネント】** > **【階層スキーマ】** をクリックします。
2. **【新規階層スキーマ】** ページで、階層スキーマの名前と場所を指定するか、デフォルト値を使用します。
3. 必要に応じて、階層スキーマの説明を入力します。
4. **【アップロード】** をクリックして、階層スキーマのベースとなるファイルを選択します。
5. **【スキーマ/サンプルファイルをアップロード】** ダイアログボックスで、**【ファイルの選択】** をクリックしてファイルを参照し、**【OK】** をクリックしてアップロードします。
6. 他の XSD ファイルを参照する XSD ファイルをアップロードすると、データ統合に参照ファイルのリストが表示されます。**【アップロード】** をクリックして、各参照ファイルをアップロードします。
7. 使用可能なルート要素が複数ある XSD ファイルをアップロードした場合は、ドロップダウンメニューから1つのルート要素を選択します。
8. 階層スキーマを保存するには、**【OK】** をクリックします。

## 第 8 章

# 業種別データサービスカスタマイザ

業種別データサービスカスタマイザを作成して業界標準メッセージをカスタマイズし、カスタムデータサービスとしてデータサービスリポジトリにパブリッシュします。

カスタマイザには、メッセージング標準に基づく、追加設定不要のメッセージタイプが用意されています。メッセージタイプを選択し、要素を追加、編集、および削除して、メッセージ構造をカスタマイズします。メッセージを保存して後で編集を続けることや、メッセージをデータサービスとしてパブリッシュして、データサービストランスフォーメーションで使用できるようにすることができます。パブリッシュ済みのデータサービスを編集することはできません。ただし、カスタマイザを編集し、別の名前を使用してメッセージを別のデータサービスとして再パブリッシュすることは可能です。

## メッセージ定義




メッセージ定義にはメッセージ構造が含まれており、メッセージ構造内の各要素には編集可能なプロパティが含まれています。プロパティは、要素とメッセージ構造内のプロパティの位置によって異なります。



メッセージ構造は、メッセージング標準に従って定義されます。メッセージの構造とプロパティの詳細については、適切なメッセージング標準を参照してください。例えば、HIPAA メッセージの構造とプロパティの詳細については、HIPAA のドキュメントを参照してください。

## メッセージの構造

メッセージは、各構造要素にアイコンでラベルが付いた階層構造を使用します。

次の表に、メッセージ構造内のアイコンに関する説明を示します。

アイコン	構造要素	説明
	セグメントグループ	繰り返し可能なセグメントのグループ。
	セグメント	データのシーケンス要素。
	コンポジットまたはフィールド	データ要素の構造。

アイコン	構造要素	説明
	コンポーネント	データ要素のコンテナ。
	データ要素	単純なデータ項目。

メッセージング標準では、構造要素に異なる名前が使用され、構造要素がどのようにネストされるかが定義されます。一部の標準では、要素にセグメントやコンポジットではなく、レコードやフィールドなどの名前が付けられています。セグメントにコンポジットとデータ要素が含まれる場合や、コンポジットにデータ要素が含まれる場合があります。一部の標準では、オプションや代替といった追加の構成要素が定義されています。

## グローバル設定および位置設定

構造要素には、編集可能なグローバル設定と位置設定があります。グローバル設定は要素のすべてのインスタンスに適用され、位置設定は要素の選択済みのインスタンスに適用されます。

例えば、グローバル設定を使用して、要素に10個の列挙値のいずれかを含めることができるようにすることができます。また、位置設定を使用して、要素を2つの値のサブセットに制限するようにすることができます。

## 業界標準メッセージのカスタマイズ

業界標準メッセージをカスタマイズして、メッセージ構造内の要素を追加、編集、および削除し、メッセージをデータサービスとしてパブリッシュします。

1. データ統合で、**[新規]** > **[コンポーネント]** > **[業種別データサービスカスタマイザ]** をクリックします。
2. **[作成]** をクリックします。
3. 全般プロパティを設定します。  
次の表は、全般プロパティの説明です。

プロパティ	説明
名前	メッセージの名前。
場所	メッセージの場所。メッセージを保存するフォルダを参照するか、デフォルトの場所を使用します。
説明	メッセージの説明。
メッセージタイプ	メッセージング標準に基づいた、追加設定不要のメッセージタイプ。

4. メッセージ構造を設定し、要素のプロパティを編集します。
  - 要素を追加するには、[「メッセージの構造への要素の追加」](#) (ページ 44) を参照してください。

- 要素のプロパティを編集するには、「[要素のプロパティの編集](#)」(ページ 45)を参照してください。
  - 列挙を追加するには、「[列挙の追加](#)」(ページ 45)を参照してください。
  - 要素を削除するには、「[メッセージ構造からの要素の削除](#)」(ページ 45)を参照してください。
5. **【保存】** をクリックします。
  6. メッセージをデータサービスとしてパブリッシュするには、**【パブリッシュ】** を選択します。
  7. データサービスのプロパティを設定します。
- 次の表に、データサービスのプロパティを示します。

プロパティ	説明
サービス名	データサービスの名前。
ランタイム環境	実行中の Secure Agent を含むランタイム環境。
パブリッシュオプション	<p>使用できるようにするためにデータサービスをパブリッシュする方法を定義します。以下のオプションを1つ以上選択します。</p> <ul style="list-style-type: none"> <li>- パーサー。データサービスを使用して、パートナーからメッセージを受信します。</li> <li>- シリアライザ。データサービスを使用して、パートナーにメッセージを送信します。</li> <li>- 制限されたシリアライザ。データサービスを使用して、検証メッセージなどのメッセージをパートナーに送信します。HIPAA メッセージに適用されます。</li> </ul> <p>カスタマイザは、選択したオプションごとにデータサービスをパブリッシュします。</p>

8. **【パブリッシュ】** をクリックします。

メッセージをデータサービスとしてパブリッシュすると、Administrator のデータサービスリポジトリでデータサービスを表示したり、データサービストランスフォーメーションで使用したりできるようになります。

## メッセージの構造への要素の追加

カスタマイザを使用して、メッセージ構造に要素を追加します。要素を追加すると、カスタマイザはメッセージング標準に基づいてネストされた要素をその要素に取り扱います。

1. メッセージ構造で要素を選択します。
2. **【追加】** ボタンをクリックして、次のオプションのいずれかを選択します。
  - **新規**。要素リストの最後に要素を追加します。
  - **間に挿入**。選択した要素の間に要素を挿入します。
3. 次のいずれかのタスクを実行します。
  - 追加する既存の要素を選択します。
  - 要素 ID を指定して新しい要素を作成します。
4. それぞれのタスクで、**【選択】** または **【作成】** をクリックします。

## 要素のプロパティの編集

カスタマイザを使用して、メッセージ構造内の要素のプロパティを編集します。

1. メッセージ構造で要素を選択します。  
【プロパティ】パネルに要素のプロパティが表示されます。
2. 【プロパティ】パネルでプロパティ値をクリックし、新しい値を入力します。
3. 必要に応じて、プロパティにマウスカーソルを合わせ、【デフォルト値の復元】を選択して変更を元に戻します。

## 列挙の追加

要素に列挙を追加して、許可されている値のテーブルを作成します。

1. メッセージ構造で要素を選択します。  
【プロパティ】パネルに要素のプロパティが表示されます。
2. 【プロパティ】パネルで、【列挙】行にカーソルを合わせ、【列挙の更新】アイコンを選択します。
3. 列挙テーブルの名前を指定します。
4. 【追加】アイコンをクリックします。
5. 列挙の値と説明を入力します。
6. 【更新】をクリックします。

## メッセージ構造からの要素の削除

カスタマイザを使用して、メッセージ構造から要素を削除します。

1. メッセージ構造で要素を選択します。
2. 【削除】ボタンをクリックします。
3. 【OK】をクリックします。

# メッセージのプロパティ

メッセージング標準により、メッセージ構造で設定できる要素とプロパティを定義します。このセクションでは、各標準のプロパティについて説明します。

## HIPAA メッセージプロパティ

HIPAA は、医療制度の管理上および財政的トランザクションの業界標準です。HIPAA は X12 標準に準拠しています。

HIPAA メッセージでは、トランザクションセット、ループ、セグメント、コンポジット、およびデータ要素を使用します。

## トランザクションセット

以下の表に、トランザクションセットのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	メッセージの識別子。
ライブラリのタイプ	グローバル	メッセージング標準。HIPAA に設定します。
ライブラリのバージョン	グローバル	メッセージング標準のバージョン。
業界 ID	グローバル	業界識別子。
標準名	グローバル	メッセージ名。
機能グループ	グローバル	メッセージが属するグループ。

## ループ

以下の表に、ループプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	ループ識別子
繰り返し	位置	ループの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。

## セグメント

次の表に、セグメントのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	セグメント識別子。
標準名	グローバル	セグメント名。
標準修飾子	グローバル	セグメントを修飾するデータ要素のインデックス。
使用率	位置	必須、オプション、条件付き、または使用されていない。
繰り返し	位置	セグメントの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。
業界名	位置	セグメントの業界標準の分かりやすい名前。

## コンポジット

以下の表に、コンポジットプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	コンポジットプロパティ。
標準名	グローバル	コンポジット名。
使用率	位置	必須、オプション、条件付き、または使用されていない。
繰り返し	位置	コンポジットの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。

## データ要素

次の表に、データ要素のプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	データ要素識別子。
標準名	グローバル	データ要素名。
タイプ	グローバル	データ要素タイプ。
最小長	グローバル	データの最小長。
最大長	グローバル	データの最大長。
使用率	位置	必須、オプション、条件付き、または使用されていない。
繰り返し	位置	データ要素の繰り返し回数（例えば、1 または 2）。
有効なコード	位置	有効なコードのリスト。

## HL7 メッセージのプロパティ

HL7 メッセージング標準は、医療サービス業界で使用されている Health Level Seven バージョン 2.x メッセージング標準を実装しています。HL7 標準は、世界中の病院医療情報システムに採用されています。

HL7 メッセージは、セグメントグループ、セグメント、フィールド、コンポーネント、およびタイプを使用します。

### メッセージ

次の表に、メッセージのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	メッセージの識別子。
ライブラリのタイプ	グローバル	メッセージング標準。HL7 に設定します。

プロパティ	設定タイプ	説明
ライブラリのバージョン	グローバル	メッセージング標準のバージョン。
説明	グローバル	メッセージの説明。
カテゴリ	グローバル	メッセージが属するカテゴリ（患者照会など）。

## セグメントグループ

次の表に、セグメントグループのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	セグメントグループの識別子。
使用方法	位置	必須またはオプション。
繰り返し	位置	セグメントの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。

## セグメント

次の表に、セグメントのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	セグメント識別子。
説明	グローバル	セグメントの説明。
使用方法	位置	必須またはオプション。
繰り返し	位置	セグメントの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。

## フィールド

次の表に、フィールドのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	フィールドの識別子。識別子は、通常、親要素の ID の後にインデックス番号が続く形式となります。
説明	グローバル	フィールドの説明。
タイプ	グローバル	フィールドのタイプ。
最大長	グローバル	データの最大長。
列挙	グローバル	許可されるデータ値のリスト。



プロパティ	設定タイプ	説明
使用方法	位置	必須またはオプション。
繰り返し	位置	セグメントの繰り返し回数（例えば、1 または 2）。無制限の場合は、>1 と入力します。

## コンポーネント

次の表に、コンポーネントのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	コンポーネントの識別子。識別子は、通常、親要素の ID の後にインデックス番号が続く形式となります。
説明	グローバル	コンポーネントの説明。
タイプ	グローバル	コンポーネントのタイプ。
使用方法	グローバル	必須またはオプション。

## タイプ

次の表に、タイプのプロパティを示します。

プロパティ	設定タイプ	説明
ID	グローバル	タイプの識別子。
説明	グローバル	タイプの説明。

## 第 9 章

# インテリジェント構造モデル

CLAIRE<sup>®</sup>インテリジェント構造モデルは、インテリジェント構造検出が、モデルでの実行時の解析が想定されるデータを表す入力に基づいて作成するアセットです。

インテリジェント構造検出は、モデルに対して指定された入力の基になるパターンと構造を判別し、出力グループのトランスフォーメーション、解析、および生成に使用されるモデルを作成します。

構造がほとんどまたはまったくない、複雑で長いファイルは、解析が困難です。インテリジェント構造検出は、入力データを自動的に解読し、構造化されていないファイルからパターン、繰り返し、関係性、データの種類の検出します。

インテリジェント構造でデータの構造が検出された後に、構造を改良してテストを行ってから、そのデータを保存またはエクスポートできます。インテリジェント構造を保存またはエクスポートすると、インテリジェント構造検出は、.amodel ファイルにインテリジェント構造モデルを作成します。

インテリジェント構造検出は、期待される出力データを表すモデルを作成します。マッピング内のインテリジェント構造モデルを使用することで、非構造化データ、半構造化データ、または構造化データを解析できます。

モデルは次の入力タイプから作成することができます。

- テキストファイル（CSV ファイルなどの区切りファイルやテキスト階層を含む複雑なファイルを含む）
- 機械生成されたファイル（Web ログやクリックストリームなど）
- JSON ファイル
- XML ファイル
- ORC ファイル
- Avro ファイル
- Parquet ファイル
- Microsoft Excel ファイル
- PDF フォームフィールド内のデータ
- Microsoft Word テーブル内のデータ
- XSD ファイル
- COBOL コピーブック

# マッピングでのインテリジェント構造モデルの使用

マッピングでインテリジェント構造モデルを使用するには、構造パーサートランスフォーメーションまたは階層ビルダートランスフォーメーションをマッピングに追加します。

インテリジェント構造モデルは、構造パーサートランスフォーメーションに必須であり、階層ビルダートランスフォーメーションではオプションです。

トランスフォーメーションを設定する場合は、インテリジェント構造モデルを選択または作成し、トランスフォーメーションで受け取ることが想定される入力のタイプと、ダウンストリームトランスフォーメーションに渡す出力を選択します。モデルは、リレーショナル、XML、JSON、JSON 行、または Hadoop 形式の出力を生成します。

**注:** 出力に DATE 型のデータが含まれる場合、インテリジェント構造検出はマッピング内のデータを、Date または DateTime ではなく String として表します。

マッピング内の構造パーサートランスフォーメーションで使用するモデルには、最大 12,000 個のフィールドを含めることができます。12,000 個を超えるフィールドを含むモデルをロードすると、構造パーサートランスフォーメーションが失敗する場合があります。

**ヒント:** JSON、XML、または XSD ベースではないモデルを作成し、モデルにネストされた繰り返しグループが含まれている場合は、出力データを正規化することでモデル内のフィールドの数を減らすことができます。インテリジェント構造検出は、ネストされた繰り返しグループを含む JSON、XML、および XSD ベースのモデルの入力データをデフォルトで正規化します。

構造パーサートランスフォーメーションと階層ビルダートランスフォーメーションの詳細については、「トランスフォーメーション」を参照してください。

## 詳細モードのマッピングでのインテリジェント構造モデルの使用

詳細モードのマッピングでインテリジェント構造モデルを使用するには、ソーストランスフォーメーションを設定してソースのデータ構造を検出するか、マッピングの下流に構造パーサートランスフォーメーションを追加します。

モデルは階層出力を生成します。詳細モードで使用するモデルには、最大 8,000 個のフィールドを含めることができます。

ソーストランスフォーメーションを設定するときにインテリジェント構造モデルを使用するには、ソーストランスフォーメーションを設定するときに既存のモデルを選択するか、新しいモデルを作成します。Amazon S3 V2 または Microsoft Azure Data Lake Gen2 接続でインテリジェント構造モデルを使用できます。次の表に、インテリジェント構造モデルを使用するようにソースプロパティを設定する場合に実行するアクションを示します。

プロパティ	アクション
接続	Amazon S3 V2 または Microsoft Azure Data Lake Gen2 接続を選択します。
ソースタイプ	<b>[単一オブジェクト]</b> を選択します。

プロパティ	アクション
オブジェクト	<p>ソースオブジェクトとしてファイルまたはフォルダを選択します。</p> <ul style="list-style-type: none"> <li>- ファイルを選択するには、<b>【詳細】</b> 領域でソースタイプとして <b>【ファイル】</b> を選択してから、インテリジェント構造検出でサポートされているタイプのファイルを選択します。</li> <li>- フォルダを選択するには、<b>【詳細】</b> 領域でソースタイプとして <b>【ディレクトリ】</b> を選択し、1つ以上の Avro、Parquet、または ORC ファイルを含むフォルダを選択します。</li> </ul>
形式	<b>【構造の検出】</b> を選択します。
インテリジェント構造モデル	<p>モデルをトランスフォーメーションに関連付けるには、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"> <li>- <b>選択</b>。既存のモデルを選択します。</li> <li>- <b>新規</b>。新しいモデルを作成します。モデルを作成するには、<b>【新規を設計】</b> を選択します。選択したサンプル入力に基づいてモデルを生成するには、インテリジェント構造検出に <b>【サンプルファイルから自動生成】</b> を選択します。</li> </ul>

詳細については、「トランスフォーメーション」を参照してください。

## Data Engineering マッピングでのインテリジェント構造モデルの使用

Data Engineering マッピングでインテリジェント構造モデルを使用するには、モデルをデータオブジェクトに追加します。モデルは HTYPE 出力を生成します。

インテリジェント構造モデルをデータオブジェクトに追加し、それらを Data Engineering のマッピングに統合できます。データオブジェクトのインテリジェント構造モデルを使用するには、まずデータ統合からローカルドライブにエクスポートします。

複雑なファイルデータオブジェクト、Amazon S3 データオブジェクト、または Microsoft Azure Blob データオブジェクトにインテリジェント構造を追加するには、Informatica Developer を使用します。データオブジェクトは Data Engineering マッピングに追加し、Spark エンジンでデータを処理できます。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。

## B2B Gateway のインバウンドパートナーフローでのインテリジェント構造モデルの使用

B2B Gateway のインバウンドパートナーフローでインテリジェント構造モデルを使用するには、パートナーを作成するときにインテリジェント構造を選択します。

インテリジェント構造モデルを使用して、受信する Excel、TXT、および CSV ファイルを受け取ることができます。インテリジェント構造検出は、ファイルを B2B Gateway のドキュメントストア上の CSV インタフェースファイルに書き込みます。

詳細については、B2B Gateway のヘルプを参照してください。

# インテリジェント構造検出のプロセス

インテリジェント構造モデルは、インテリジェント構造検出を使用して作成できます。

入力ファイルの指定後インテリジェント構造検出は、繰り返し発生するデータパターンの基礎的な部分を判別し、データフィールドとその関係を表す構造を作成します。ログファイル、クリックストリーム、顧客の Web アクセス、エラーテキストファイル、その他業界標準に準拠しないインターネットデータ、センターデータ、デバイスデータなど、ファイルの構造がきわめて複雑で、検出に時間とコストがかかる場合でも、データをすばやくモデル化できます。

次の図は、インテリジェント構造検出が、根底にあるデータパターンを解読し、そのデータパターンのモデルを作成するプロセスを示しています。



モデルが Hadoop ファイルソースからの入力に基づいている場合、インテリジェント構造検出は、Hadoop スキーマのタイプと構造に基づいてモデルで使用するデータ型を決定します。インテリジェント構造検出は、モデル内にマップ型のデータのキーと値のペアの構造の配列を作成します。Enum 型のノードは文字列として表されます。

## インテリジェント構造モデルに対する入力

実行時にモデルでの使用が想定される入力に基づいて、インテリジェント構造モデルの基礎となる入力は、サンプルファイル、XSD スキーマ、Avro スキーマ、または COBOL コピーブックを使用することができます。

入力ファイルのサイズは最大 1 MB です。入力ファイルには、最大 30,000 個の単純なフィールドを含めることができます。ファイルに 30,000 を超える単純なフィールドが含まれる場合、インテリジェント構造検出は、グループとポートのないモデルを作成します。階層内のレベル数に制限はありません。

最適な解析結果を得るため、モデルの作成時に指定する入力は、実行時にモデルが受け取ると予想されるすべてのデータ要素を網羅するように幅広くしてください。入力の範囲が狭すぎる場合、解析の出力に識別できないデータが含まれます。入力に行が含まれる場合は、少なくとも 3 行のデータが含まれている必要があります。

簡易的な入力を使用してモデルを生成します。例えば、入力データにテーブルがある場合は、多数のデータ行ではなく、いくつかのサンプル行を含むテーブルを使用します。データの繰り返しグループを含む JSON 入力ファイルを使用する場合は、繰り返しの回数を制限してください。

モデルが実行時の入力データと一致しない場合、または入力データと部分的にしか一致しない場合は、大量の未確認データが存在するかデータ損失が発生している可能性があります。ただし、一部のバリエーションは引き続き解析されます。

グループ名とフィールド名の各組み合わせの長さが 80 文字を超えていないことを確認してください。例えば、フィールドが属するグループの名前が group である場合、フィールド名に 75 文字を超える名前を使用できません。グループ名とフィールド名の組み合わせが 80 文字を超えていると、モデルを使用するマッピングの実行に失敗します。

## XML または JSON サンプルファイル全体からの構造の検出

デフォルトでは、インテリジェント構造検出は、サンプルファイルの最初の部分に基づいてデータの構造を検出します。XML または JSON ファイルに基づいてモデルを作成する場合、最大 30 MB のファイル全体に基づいてデータの構造を検出するように選択できます。ファイルの最初の部分が、実行時にモデルを使用すると想定されるすべての入力を表していない場合は、このオプションを使用します。

インテリジェント構造検出がファイル全体に基づいてデータの構造を検出する場合、検出プロセスが完了するまでに数分かかる場合があります。

## Microsoft Excel ファイルからの構造の検出

次の種類の Microsoft Excel ファイルに基づいてモデルを作成できます: xla、xlam、xls、xlsm、xlsx、xlt、xltm、および xlsx。

## ORC ファイルの使用

モデルを使用して、データ統合のフラットファイル接続を介して ORC ファイルを読み取ることができます。このモデルを ORC のストリーミングに使用することはできません。

## モデルでの複数のサンプルファイルの使用

JSON、XML、ORC、AVRO、または PARQUET サンプルファイルに基づいてモデルを作成した後に、追加のサンプルファイルを使用して、新しいサンプルに存在するフィールドで構造をエンリッチ化できます。追加のファイルは、モデルの基礎となるファイルのタイプと同一のファイルタイプである必要があります。

## マルチファイル XSD スキーマの使用

モデル入力として複数の XSD ファイルを含む XSD スキーマを使用する場合は、次のガイドラインを考慮してください。

- スキーマファイルは圧縮する必要があります。
- XSD ファイルがディレクトリ構造に存在する場合、構造を保持するには、親ディレクトリを圧縮する必要があります。

## 大規模な XSD スキーマからの構造の検出

XSD スキーマに基づいてモデルを作成する場合、デフォルトでは、インテリジェント構造検出は最大 1.5 MB のサイズのスキーマからデータの構造を検出できます。より大きなファイルを使用するには、次のいずれかのアクションを実行します。

- 最大 30 MB のスキーマ全体に基づいてデータの構造を検出するように選択します。
- スキーマファイルを圧縮し、モデルの入力としてその zip ファイルを選択します。

これらのアクションのいずれかを実行せずに大規模なスキーマを使用すると、インテリジェント構造検出は入力を XML として扱い、部分的なデータに基づいて構造を検出します。

## XSD ベースのモデルでの XML サンプルファイルの使用

構造パーサートランスフォーメーションで使用する XSD ベースのモデルを作成する場合に、XML サンプルファイルをモデルに添付できます。モデル内のグループの名前と内容は、インテリジェント構造モデルページに表示されます。モデルを構造パーサートランスフォーメーションに関連付ける場合は、この情報を使用して、ターゲットに接続するグループを指定します。サンプルファイルをモデルに添付した場合でも、モデルの構造に影響を及ぼしたり、モデルの構造が変更されたりすることはありません。

## JSON でエンコードされた Avro メッセージの解析

Avro スキーマに基づくモデルを使用して、JSON でエンコードされた Avro メッセージを解析できます。

## XSD スキーマの文字エンコード

モデル入力として使用する XSD スキーマでは、UTF-8 または UTF-16 文字エンコードを使用できます。

# 出力グループ定義

XML ファイルまたは XSD ファイルをベースとしてインテリジェント構造モデルを作成する場合は、インテリジェント構造検出で出力グループを定義する方法を選択できます。

入力ファイルの構造を確認する前に、**[設定]** タブで次のいずれかのオプションを選択できます。

## デフォルトの出力グループ

インテリジェント構造検出は、正規化または非正規化された出力グループを生成できます。入力ファイルで使用されている場合、インテリジェント構造検出は XSD コンポジタをベースとしたグループ（選択グループなど）も生成します。

## 階層リレーションの正規化

この方法は、繰り返し要素ごとに出力グループを生成します。正規化された階層リレーションを使用すると、通常はデフォルトの出力グループを使用するよりも高速になり、メモリ使用量が少なくなります。

## エンティティリレーション

この方法は、各繰り返し要素および複合データ型の出力グループを生成します。エンティティリレーションを使用すると、複合データ型の最もコンパクトな表現が作成されます。

インテリジェント構造検出は、エンティティリレーションを生成するときに、以下のルールを使用します。

- インテリジェント構造検出は、すべての出力グループに<group name>\_PK という名前のプライマリキーを追加します。
- インテリジェント構造検出は、次のグローバル要素の出力グループを生成します。
  - ルート要素
  - 再帰参照の一部である要素
  - インジケータ maxOccurs = "unbounded" の繰り返し要素
- インテリジェント構造検出は、グローバル要素を参照する各出力グループのグローバル要素に基づいて、出力グループに外部キーを追加します。外部キーは、<parent group name>\_FK という名前を使用します。グローバル要素を参照する親グループが複数ある場合、グローバル要素の出力グループには複数の外部キーがあります。
- インテリジェント構造検出は、すべての複合型の出力グループを生成します。出力グループの名前は、複合型の名前です。
- 複合型が親要素と 1 対 1 の関係にある場合、インテリジェント構造検出は、<parent element name>\_<type name>\_FK という名前の外部キーを複合タイプの出力グループに追加します。
- 複合型とその親要素の間に 1 対多または多対多の関係がある場合、インテリジェント構造検出は<group name 1>\_<group name 2>というブリッジテーブルを生成します。ブリッジテーブルには、プライマリキー、最初の出力グループへの外部キー、および 2 番目の出力グループへの外部キーのみが含まれます。

どの出力グループも、最初のフィールドは生成されたプライマリキーで、その後外部キーが続き、その後スキーマに存在するフィールドが続きます。

# 繰り返しグループ

インテリジェント構造検出は、行または一連のフィールドをグループとして繰り返す、テーブルや配列などの入力データの繰り返しグループを作成します。

JSON、XML、または XSD ファイルに基づくモデルに、ネストされた繰り返しグループ、つまり繰り返しグループ内に繰り返しグループが含まれている場合、インテリジェント構造検出は、ネストされた各繰り返しグル

ープを独自の出力グループに割り当てます。他の入力タイプに基づくモデルの場合は、ネストされた繰り返しグループをそれらの出力グループに手動で割り当てることができます。モデル内のデータを正規化することにより、すべてのネストされた繰り返しグループを出力グループに割り当てることができます。または、個々のネストされた繰り返しグループを、昇格することで出力グループに割り当てることができます。ネストされた繰り返しグループを出力グループに割り当てると、モデル内のポートの数が減ります。

例えば、次の JSON 入力に基づいてモデルを作成するとします。

```
{
  "CompanyID": 210850,
  "Name": "Tollers Shipping",
  "Address": "701 Natash Ave.",
  "City": "San Diego",
  "Department": [
    {
      "Domain": "Logistics",
      "Identifier": "21973b77",
      "Employees": [
        {
          "name": "Sujitha Naarana",
          "employeeID": "210029"
        },
        {
          "name": "Dwain Lord",
          "employeeID": "34t001"
        }
      ]
    },
    {
      "Domain": "Accounting",
      "Identifier": "301ad177",
      "Employees": [
        {
          "name": "LeTroy Prince",
          "employeeID": "31910a"
        }
      ]
    }
  ]
}
```

次の画像は、インテリジェント構造検出が検出した構造を示しています。

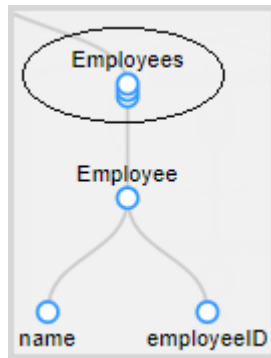
A	B	C	D	E			
210850	CompanyID	Name	Tollers Shippi...	701 Natash Ave.	San Diego	group_documentId	9066744a-546c-463c-a3e6-6298...
8a776972-2523-4482-a00e-2cc45...	Department_PK	Domain	Logistics	21973b77	Identifier	Department_documentId	9066744a-546c-463c-a3e6-6298...
386b8228-3be5-44e8-8a11-c347...	Department_PK	Domain	Accounting	301ad177	Identifier	Department_documentId	9066744a-546c-463c-a3e6-6298...
Sujitha Naarana	name	employeeID	210029	8a776972-2523-4482-a00e-2cc45...	Department_FK		
Dwain Lord	name	employeeID	34t001	8a776972-2523-4482-a00e-2cc45...	Department_FK		
LeTroy Prince	name	employeeID	31910a	386b8228-3be5-44e8-8a11-c347...	Department_FK		

構造には、ネストされた繰り返しグループが含まれています。トップレベルの繰り返しグループは **Department** です。各 **Department** グループには、**Employee** グループノードを含む、**Employees** 繰り返しグループがあります。**Employee** グループノードには、**name** と **employeeID** という子ノードがあります。

繰り返しグループは、親ノードの3つの円が重なるようにビジュアルモデルに表示されます。



次の図は、**Employees** 繰り返しグループを示しています。



## プライマリキーおよび外部キー

ネストされた繰り返しグループを含むモデルの場合、インテリジェント構造検出は、プライマリキーと外部キーを使用して、繰り返しグループとその子ノード間の関係を識別します。

JSON、XML、または XSD ファイルに基づくモデルに、ネストされた繰り返しグループが含まれている場合、インテリジェント構造検出は、出力データを正規化し、ネストされた各繰り返しグループを自身の出力グループに割り当てます。他の入力タイプに基づくモデルの場合は、ネストされた繰り返しグループをそれらの出力グループに手動で割り当てることができます。

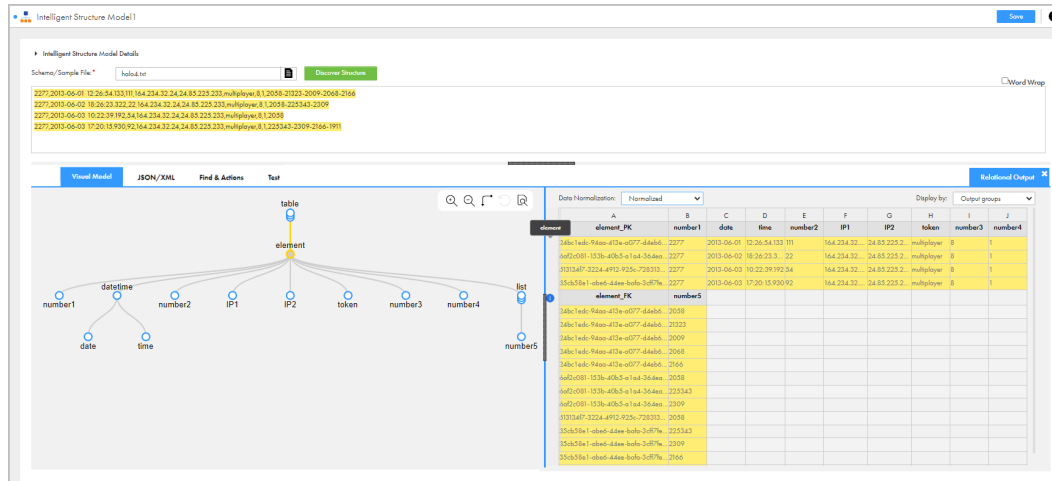
ネストされた繰り返しグループがその出力グループに割り当てられると、インテリジェント構造検出は、親グループにプライマリキーを追加し、子グループに外部キーを追加します。

次の図は、インテリジェント構造検出が CSV 入力ファイルから検出した構造を示しています。

A	B	C	D	E	F	G	H	I	J
number1	date	time	number2	IP1	IP2	token	number3	number4	number5
2277	2013-06-01	12:26:54.133	111	164.234.32	24.85.225.2	multiplayer	B	1	2058
2277	2013-06-01	12:26:54.133	111	164.234.32	24.85.225.2	multiplayer	B	1	23323
2277	2013-06-01	12:26:54.133	111	164.234.32	24.85.225.2	multiplayer	B	1	2009
2277	2013-06-01	12:26:54.133	111	164.234.32	24.85.225.2	multiplayer	B	1	2068
2277	2013-06-01	12:26:54.133	111	164.234.32	24.85.225.2	multiplayer	B	1	2166
2277	2013-06-01	18:26:23.3	12	164.234.32	24.85.225.2	multiplayer	B	1	2058
2277	2013-06-02	18:26:23.3	12	164.234.32	24.85.225.2	multiplayer	B	1	223343
2277	2013-06-02	18:26:23.3	12	164.234.32	24.85.225.2	multiplayer	B	1	2309
2277	2013-06-03	10:22:39.192	54	164.234.32	24.85.225.2	multiplayer	B	1	2058
2277	2013-06-03	17:20:15.930	92	164.234.32	24.85.225.2	multiplayer	B	1	223343
2277	2013-06-03	17:20:15.930	92	164.234.32	24.85.225.2	multiplayer	B	1	2309
2277	2013-06-03	17:20:15.930	92	164.234.32	24.85.225.2	multiplayer	B	1	2166
2277	2013-06-03	17:20:15.930	92	164.234.32	24.85.225.2	multiplayer	B	1	1911

このモデルでは、**list** グループが **element** 出力グループの一部となっています。データ正規化モードが非正規化され、ネストされた繰り返しグループである **list** は、別の出力グループに割り当てられません。

次の画像は、データ正規化モードを正規化に変更した後の同じモデルを示しています。



インテリジェント構造検出は、2つの異なる出力グループ、**element** 出力グループと **list** 出力グループを生成します。グループの左側にあるヒントアイコンにカーソルを合わせると、グループ名を表示できます。

インテリジェント構造検出は、プライマリキー **element\_PK** を親の **element** 出力グループに追加し、外部キー **element\_FK** をネストされた出力グループ **list** に追加しました。

別のノードをレコード ID として定義することでプライマリキーとして選択できます。レコード ID を変更すると、インテリジェント構造検出は対応する外部キーをネストされたグループに作成します。

## データドリフト

インテリジェント構造モデルは、次の例のように、特定の場合にデータドリフトに対応します。この例では、モデルの作成に使用されるサンプルデータに次のテキストが含まれています。

```
05967|2014-09-19|04:49:50.476|51.88.6.206|custid=83834785|cntry=Tanzania|city=Mtwango|
movie={b1027374-6eec-4568-8af6-6c037d828c66|"Touch of Evil"}|paid=true
01357|2014-11-13|18:07:57.441|88.2.218.236|custid=41834772|movie={01924cd3-87f4-4492-b26c-268342e87eaf|"The
Good, the Bad and the Ugly"}|paid=true
00873|2014-06-14|09:16:14.522|134.254.152.84|custid=58770178|movie={cd381236-53bd-4119-
b2ce-315dae932782|"Donnie Darko"}|paid=true
02112|2015-01-29|20:40:37.210|105.107.203.34|custid=49774177|cntry=Colombia|city=Palmito|movie={ba1c48ed-
d9ac-4bc-b5ed-cf3afbb61f04|"Lagaan: Once Upon a Time in India"}|paid=false
00408|2014-06-24|03:44:33.612|172.149.175.30|custid=29613035|cntry=Iran|city=Bastak|movie={3d022c51-f87f-487a-
bc7f-1b9e5d138791|"The Shining"}|paid=false
03568|2015-01-07|11:36:50.52|82.81.202.22|custid=27515249|cntry=Philippines|city=Magallanes|
movie={ad3ae2b4-496e-4f79-a6dd-202ec932e0ae|"Inglourious Basterds"}|paid=true
```

モデルが解析する入力データには、次のテキストが含まれています。

```
0448|2015-04-07|01:50:5.35|27.248.247.174|custid=613068|cntry=Iran|city=SarÄ•b|movie={50fb37b-621-484e-
a565-2b5c1cbdc43|"Network"}|paid=false|ua=Mozilla/5.0 (Windows NT 5.1)
02780|2014-12-28|08:14:58.685|17.2.236.233|custid=731|cntry=Greece|city=NÄ•a RÄ•3da|
movie={1876aea0-3cb5-4c7a-22f-d33f233210|"Full Metal Jacket"}|paid=true|ua=Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_10_1)
03353|2015-04-20|21:02:40.532|143.48.11.171|custid=83736441|cntry=Russia|city=Mozhaysk|movie={67272f85-
bfc-418a-82ea-a7c4ae6b028a|"Gangs of Wasseypur"}|paid=true|ua=Mozilla/5.0 (iPad; CPU OS 5_1 like Mac OS X)
04073|2014-10-25|15:33:03.442|87.235.48.100|custid=861028|cntry=Indonesia|city=Lamalera|
movie={4a511f3-6367-4017-874e-50a46f5ea567|"Shutter Island"}|paid=false|ua=Mozilla/5.0 (X11; Linux x86_64)
02170|2015-02-1|23:36:40.271|25.14.204.46|custid=1240203|cntry=Albania|city=LukovÄ•|
movie={2047efa-22c6-431c-87d4-ca73af1034|"The Grapes of Wrath"}|paid=false|ua=Mozilla/5.0 (Windows NT 6.1)
```

入力データには、モデルで定義されていない追加のデータが含まれています。ただし、モデルはこの差異があっても解析は可能です。

## 未割り当てデータ

インテリジェント構造検出は、ファイルの構造を検出するときに、ファイルにインテリジェント構造モデルと一致しないレコードがある場合、`[未割り当てデータ]` というフィールドを作成することがあります。

インテリジェント構造検出は、以下のシナリオを含むさまざまな理由で `[未割り当てデータ]` フィールドを作成します。

- CSV ファイルやログファイルなどの区切りファイルには、想定以上に多くの要素が含まれており、インテリジェント構造検出はデータドリフトを解析できません。
- JSON ファイル内のレコードがインテリジェント構造モデルに含まれていないか、最大レコードサイズを超えています。

## 最大レコードサイズ

インテリジェント構造検出は、JSON サンプルファイルの構造を検出するときに、最大レコードサイズを使用して繰り返しレコードを識別します。レコードが最大レコードサイズより大きい場合、インテリジェント構造検出はそのレコードを `[未割り当てデータ]` フィールドに割り当てます。

デフォルトの最大レコードサイズは 640,000 バイトです。最大レコードサイズを大きくして、`[未割り当てデータ]` フィールドを使用しないようにすることができます。

最大レコードサイズを編集するには、Administrator を使用して、データ統合サーバープロパティの JVM オプションを設定します。次の構文を使用して、最大レコードサイズを定義します。

```
-DISD_MAX_RECORD_SIZE=<size in bytes>
```

例えば、2 MB の最大レコードサイズを定義するには、JVMOption1 プロパティに次の値を入力します。

```
-DISD_MAX_RECORD_SIZE=2000000
```

**注:** 最大レコードサイズを大きくすると検出プロセスのメモリ消費量が増えることに注意してください。このため、必要に応じて次のアクションのいずれかまたは両方を実行する必要があります。

- データ統合サーバープロパティの最大 JVM ヒープサイズを増やします。JVM ヒープサイズを増やすには、JVM プロパティの 1 つを `-Xmx<heap size in megabytes>` に設定します。
- Secure Agent マシンのメモリを増やします。Secure Agent が Informatica Cloud ホステッドエージェントで実行されている場合は、Informatica グローバルカスタマサポートにお問い合わせください。

データ統合サーバープロパティと Secure Agent の設定の詳細については、Administrator ヘルプを参照してください。

## インテリジェント構造モデルの作成

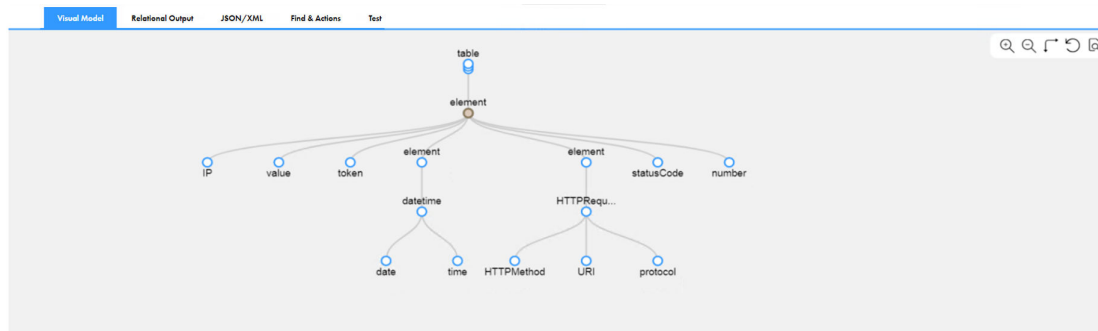
モデルでの実行時の解析が想定されるデータを表した入力に基づいて、インテリジェント構造モデルを作成します。

1. **[新規]** > **[コンポーネント]** > **[インテリジェント構造モデル]** をクリックし、**[作成]** をクリックします。

2. **[インテリジェント構造モデル]** ページで、インテリジェント構造モデルの名前を入力します。  
この名前に含めることができるのは英数字およびアンダースコア文字のみです。
3. モデルを保存するプロジェクトおよびフォルダに移動するか、デフォルトの場所を使用します。  
名前や場所は、インテリジェント構造モデルを保存した後 **[エクスプローラ]** ページで変更できます。
4. 使用している入力の種類に応じて、次のいずれかの手順を実行します。
  - JSON サンプルファイルを使用するには、まず、モデルのベースをファイルサンプリングとファイル全体のどちらにするかを選択します。ファイルを選択し、**[構造の検出]** をクリックします。
  - XML サンプルファイルを使用するには、まず、モデルのベースをファイルサンプリングとファイル全体のどちらにするかを選択します。ファイルを選択し、出力グループの定義方法を選択して、**[構造の検出]** をクリックします。
  - Avro スキーマファイルまたはその他のタイプのサンプルファイルを使用するには、ファイルを選択して **[構造の検出]** をクリックします。
  - XSD スキーマファイルを使用するには、まず、ファイルサンプリングをベースとしてモデルを作成するか、スキーマが 1.5 MB を超える場合はスキーマ全体に基づいてモデルを作成するかを選択します。ファイルを選択し、スキーマルートが選択されていることを確認してから、出力グループの定義方法を選択し、**[構造の検出]** をクリックします。構造パーサトランスフォーメーションでモデルを使用する場合は、**[XML サンプルのアップロード]** をクリックして、モデルに添付する XML サンプルファイルを選択します。
  - COBOL コピーブックを使用するには、コピーブックを選択します。必要に応じて、実行時に使用する **[ファイル編成]** と **[コードページ]** の値を変更します。**[構造の検出]** をクリックします。

入力タイプの詳細については、[「インテリジェント構造モデルに対する入力」 \(ページ 53\)](#)を参照してください。

**[構造の検出]** をクリックすると、インテリジェント構造検出によって入力内のデータが解読され、データで表現されたパターンが検出されます。次の図に、**[ビジュアルモデル]** タブに表示される、検出された構造の例を示します。



インテリジェント構造検出により、一意の名前を持つノードが作成されます。インテリジェント構造検出は、同じ型のデータのインスタンスを検出し、ノード名に数字のサフィックスを追加します。例えば、入力に 2 つのテーブルのタイムスタンプが含まれている場合、インテリジェント構造検出はそれらに **timestamp1** と **timestamp2** という名前を付けます。

5. Avro、ORC、または Parquet ファイルをベースとしてモデルを作成する場合は、インテリジェント構造検出が、データ要素とファイルスキーマの要素の両方を検出します。デフォルトでは、インテリジェント構造検出は、スキーマにのみ表示される要素をモデルから除外します。スキーマ要素を出力に追加するには、それらをモデルの構造に含めます。詳細については、[「複数ノードでのアクションの実行」 \(ページ 76\)](#)を参照してください。
6. Excel ワークシート用に作成するモデルの場合、インテリジェント構造検出は、シートのインデックスと名前を使用してメタデータノードを作成します。デフォルトでは、インテリジェント構造検出は、これら

のノードをモデルの構造から除外します。ノードを出力に追加するには、それらを構造に含めます。詳細については、「[Microsoft Excel 入力の構造の編集](#)」(ページ 78)を参照してください。

7. 本番環境でモデルを使用するとき出力が要件を満たすように、構造を改良できます。詳細については、[第 10 章, 「インテリジェント構造モデルの改良」](#) (ページ 66)を参照してください。
8. **【保存】** をクリックします。  
インテリジェント構造検出でインテリジェント構造モデルが生成され、選択した場所にそのモデルが保存されます。

## インテリジェント構造モデルのエクスポート

インテリジェント構造モデルを Data Engineering マッピングで使用する前に、これをエクスポートする必要があります。モデルをローカルドライブにエクスポートした後で、そのモデルを Data Engineering データオブジェクトに統合できます。

インテリジェント構造モデルをエクスポートするには、組織に適切なライセンスが必要です。

1. **【エクスプローラ】** ページで、インテリジェント構造モデルを含むプロジェクトとフォルダに移動します。  
**【エクスプローラ】** ページには、フォルダ内のすべてのアセットが表示されます。
2. 関連するインテリジェント構造モデルを含む行をクリックして選択します。**【アクション】** メニューで、**【編集】** を選択します。**【インテリジェント構造の詳細】** パネルで、**【編集】** をクリックします。  
インテリジェント構造モデルは、別のページに表示されます。
3. **【インテリジェント構造】** ページで、**【ビジュアルモデル】** タブの右上隅にあるアイコンメニューを見つけ、**【モデルのエクスポート】** アイコンをクリックします。  
**注:** **【モデルのエクスポート】** アイコンは、組織に適切なライセンスがある場合のみ使用できます。

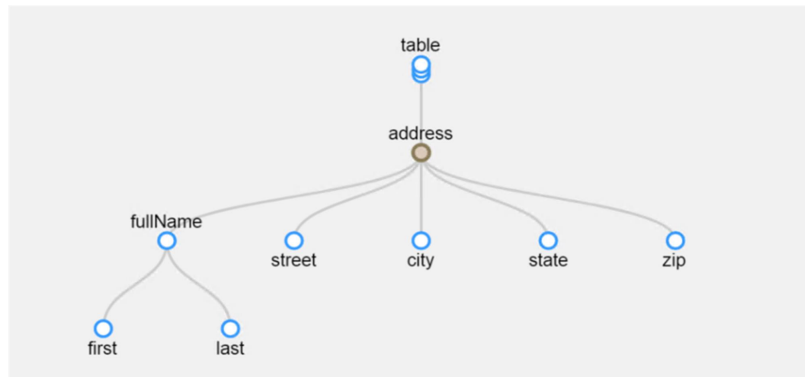
## インテリジェント構造モデル例

インテリジェント構造検出は、指定された入力データの構造に基づいてインテリジェント構造モデルを作成します。

一例として、次のコンテンツを含む、CSV 入力ファイル用のモデルの作成例を示します。

```
first,last,street,city,state,zip
Carrine,Stone,17 Torrence Street,Livingston,PA,10173
Poona,Tillkup,52 Perez Avenue,Livingston,PA,10256
Tasha,Herrera,158 Shiraz Boulevard,Kensington,WA,33823
John,Washington,22A Zangville Drive,Tucson,AZ,20198
Jane Hochuli 4483 Central Street Suite 30 Phoenix PA 38721
```

次の図は、インテリジェント構造検出が入力ファイルに基づいて検出する構造を示しています。



インテリジェント構造検出によって、**first**、**last**、**street**、**city**、**state**、**zip** などの入力ファイル内のフィールドを表すノードが作成されたことがわかります。

構造は、データフィールド自体を表すだけでなく、フィールド間の関係性も示しています。例えば、インテリジェント構造検出は、データ Carrine,Stone が人物の名と姓を表していると認識しました。ノード **first** とノード **last** は、ノード **fullName** の下でグループ化され、互いの関係性を表しています。

インテリジェント構造検出は、データ全体がアドレスを示していることも認識しました。データは親ノードの **address** の下でグループ化されます。

ノードは、出力の一部であるフィールドを表しています。関連するノードは1つの出力グループにグループ化されます。出力グループには、1つ以上のノードを含めることができます。

## ユースケース

ストリーミングメディアサービス会社のオペレーショングループで働いているとします。所属するチームが求めるのは、サーバーファームからの Web ログを処理して業務分析を行い、保守上の問題を特定することです。

バックエンドシステムは、サーバーアクセスとサーバーファーム内のシステム負荷に関するデータを収集します。チームが求めるのは、過去数週間に最大のサーバー負荷を引き起こした操作を特定することです。その後、監査目的でデータを保存することも必要です。

データアナリストがデータを処理する前に、データを解析する必要があります。ただし、ログは半構造化された状態であり、サーバーのアップグレード後はログファイルの構造がわずかに変わり、一部の情報については形式が変わる可能性もあります。標準のトランスフォーメーションでは、これによってデータ損失やログ処理の失敗が発生する場合があります。

入力データにヘッダーが含まれている場合、インテリジェント構造検出はさまざまな場所へのデータドリフトをサポートします。入力データにヘッダーが含まれていない場合、インテリジェント構造検出は入力データの最後の追加のデータを識別します。

最初のログファイルの構造は次のとおりです。

```
05967|2014-09-19|04:49:50.476|51.88.6.206|custid=83834785|cntry=Tanzania|city=Mtwango|
movie={b1027374-6eec-4568-8af6-6c037d828c66|"Touch of Evil"}|paid=true
01357|2014-11-13|18:07:57.441|88.2.218.236|custid=41834772|movie={01924cd3-87f4-4492-b26c-268342e87eaf|"The
Good, the Bad and the Ugly"}|paid=true
00873|2014-06-14|09:16:14.522|134.254.152.84|custid=58770178|movie={cd381236-53bd-4119-
b2ce-315dae932782|"Donnie Darko"}|paid=true
02112|2015-01-29|20:40:37.210|105.107.203.34|custid=49774177|cntry=Colombia|city=Palmito|movie={ba1c48ed-
d9ac-4bcb-be5d-cf3afb61f04|"Lagaan: Once Upon a Time in India"}|paid=false
00408|2014-06-24|03:44:33.612|172.149.175.30|custid=29613035|cntry=Iran|city=Bastak|movie={3d022c51-f87f-487a-
bc7f-1b9e5d138791|"The Shining"}|paid=false
```

```
03568|2015-01-07|11:36:50.52|82.81.202.22|custid=27515249|cntry=Philippines|city=Magallanes|
movie={ad3ae2b4-496e-4f79-a6dd-202ec932e0ae|"Inglourious Basterds"}|paid=true
```

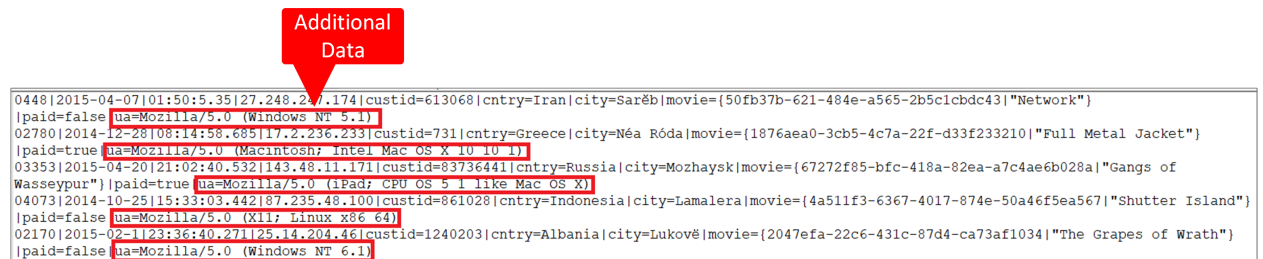
サーバーのアップグレード後に、一部のログファイルは次のような構造になります。

```
0448|2015-04-07|01:50:5.35|27.248.247.174|custid=613068|cntry=Iran|city=SarÄb|movie={50fb37b-621-484e-
a565-2b5c1cbdc43|"Network"}|paid=false|ua=Mozilla/5.0 (Windows NT 5.1)
02780|2014-12-28|08:14:58.685|17.2.236.233|custid=731|cntry=Greece|city=NÄ@a RÃ³da|
movie={1876aea0-3cb5-4c7a-22f-d33f233210|"Full Metal Jacket"}|paid=true|ua=Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_10_1)
03353|2015-04-20|21:02:40.532|143.48.11.171|custid=83736441|cntry=Russia|city=Mozhaysk|movie={67272f85-
bfc-418a-82ea-a7c4ae6b028a|"Gangs of Wasseypur"}|paid=true|ua=Mozilla/5.0 (iPad; CPU OS 5_1 like Mac OS X)
04073|2014-10-25|15:33:03.442|87.235.48.100|custid=861028|cntry=Indonesia|city=Lamalera|
movie={4a511f3-6367-4017-874e-50a46f5ea567|"Shutter Island"}|paid=false|ua=Mozilla/5.0 (X11; Linux x86_64)
02170|2015-02-1|23:36:40.271|25.14.204.46|custid=1240203|cntry=Albania|city=LukovÄ|
movie={2047efa-22c6-431c-87d4-ca73af1034|"The Grapes of Wrath"}|paid=false|ua=Mozilla/5.0 (Windows NT 6.1)

02170|2015-02-1|23:36:40.271|25.14.204.46|custid=1240203|cntry=Albania|city=LukovÄ|
movie={2047efa-22c6-431c-87d4-ca73af1034|"The Grapes of Wrath"}|paid=false|ua=Mozilla/5.0 (Windows NT 6.1)
|paid=false|ua=Mozilla/5.0 (Windows NT 6.1)
```

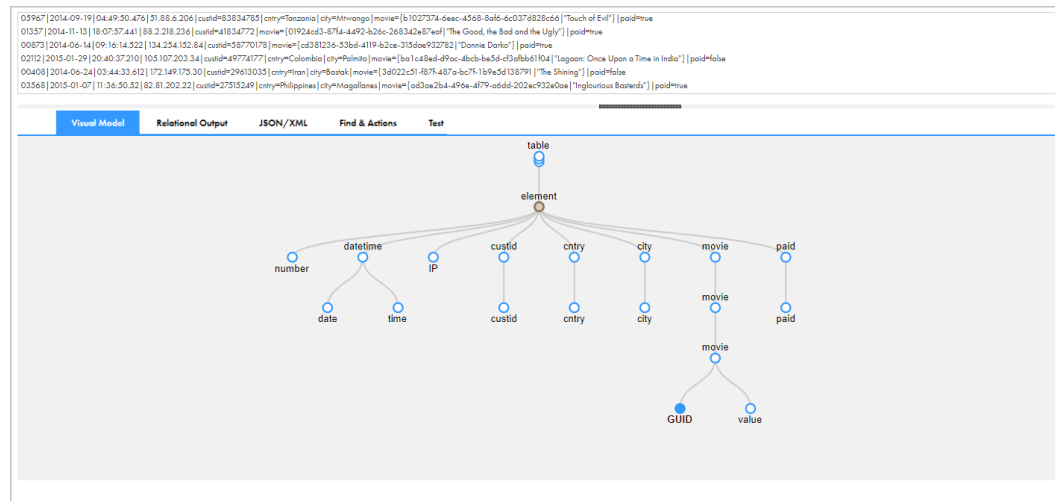
データ形式は様々で、一部のデータについては場所も変わっています。

次の図は、データの変化を示しています。



このような場合、個々のトランスフォーメーションを手動で作成する代わりにインテリジェント構造モデルを作成して、関連するデータセットを判別することができます。インテリジェント構造をインテリジェント構造検出で作成し、データの構造を自動的に識別します。

次の図は、作成するインテリジェント構造を示しています。



データを確認すると、モデルの最初の要素は number で、実際にはユーザートランザクション ID を表しています。この要素名を transactionId に変更します。

次の図は、更新されたインテリジェント構造を示しています。

A	B	C	D	E	F	G	H	I
transactio...	date	time	IP	custid	cntry	city	movie_GUID	movie_value
2014-09-19	04:49:50.476	151.88.6.206	83834785	Tanzania	Mtwanga	b1027374-6eec-4568-8af6-6c037d820c6d	"Touch of Evil"	
2014-11-13	18:07:57.441	88.2.218.236	41834772			01924ed3-87f4-4492-b26c-26834	"The Good, the Bad and the Ugly"	
2014-06-14	09:16:14.522	134.254.152.84	custid=98770178			cd387236-53bd-4119-b2ce-315d0e93292	"Donnie Darko"	
2015-01-29	20:40:37.210	105.107.203.34	custid=49774177	Colombia	Palmita	ba1c48e4-d9ac-4dcb-ba5d-d3afbb61f04	"Logan: Once Upon a Time in India"	
2014-06-24	03:44:33.612	172.149.175.30	custid=29613035	Iran	Bastak	3d022c51-8b7f-487a-bc7f-1b9a5d138791	"The Shining"	
2015-01-07	11:36:50.52	82.81.202.22	custid=27515249	Philippines	Magallanes	ad30e2b4-496e-4f79-06dd-202ec932e0e	"Inglourious Basterds"	

インテリジェント構造をインテリジェント構造モデルとして保存した後、構造パーサートランスフォーメーションを作成し、それにモデルを割り当てます。データ統合マッピングには、トランスフォーメーションをソース、ターゲット、その他のトランスフォーメーションとともに追加できます。マッピングがソース接続から Amazon S3 入力バケットなどのデータをフェッチした後、構造パーサーはインテリジェント構造モデルを使用してデータを処理します。トランスフォーメーションは、さらに処理を進めるため、Web ログデータをダウンロードストリームトランスフォーメーションに渡し、続いて Amazon S3 出力バケットなどのターゲットに渡します。

## インテリジェント構造モデルのトラブルシューティング

インテリジェント構造モデルを作成する際は、次のトラブルシューティングのヒントを参考にしてください。

### 構造が異なるファイルを使用するとデータを失う。

インテリジェント構造モデルが入力ファイルと一致しない場合、または入力ファイルと部分的にしか一致しない場合は、データを失う可能性があります。

例えば、computer ID、computer IP address、access URL、username、password、access timestamp という 6 つのデータフィールドを持つ行があるサンプルファイルのモデルを作成したとします。しかし、一部の入力ファイルには、8 つのデータフィールド (computer ID、computer name、computer IP address、country of origin、access URL、username、password、access code、access timestamp) を持つ行があるとします。この場合、データが正しく識別されず、一部のデータは未確認データとして指定される場合があります。

他の入力ファイルと比べ、データタイプの数が多い、またはデータタイプの種類が違う入力ファイルがある場合、最善の結果を得るには、すべてのデータタイプを含むサンプルファイルを作成します。

### Microsoft Word ファイルまたは Microsoft Excel ファイルのデータが解析されない。

インテリジェント構造検出は、Microsoft Word ファイルまたは Microsoft Excel ファイルに基づくモデルを作成すると、構造化されていないデータを未解析ノードとして検出し、そのノードをモデル構造や出力から除外する場合があります。例えば、ファイルにフリーテキストが含まれる場合などです。除外されたノードを構造に含めるようにモデルを編集することができます。詳細については、[「ノードの編集」\(ページ 72\)](#)を参照してください。



### PDF 形式のデータがモデル化または解析されない。

インテリジェント構造モデルは、PDF フォームのフィールド内にあるデータを解析します。PDF フォームにフィールドが含まれていることを確認してください。

### エラー: サポートされないフィールド名が原因でデータを失う場合がある。

異なる要素に重複する名前を使用しないでください。

Big Data Management 10.2.1 を使用する場合は、出力グループの名前が Informatica Developer の命名規則に従っていることを確認してください。要素名は英字 (A-Z、a-z)、数字 (0-9)、アンダースコアのみで構成する必要があります。予約済みの論理名は使用せず、要素名の先頭には数字を指定しないでください。

Big Data Management または Data Engineering Integration の新しいバージョンでは、インテリジェント構造検出は要素名の特殊文字をアンダースコアに置き換え、数字で始まる要素名の前と論理名として予約されている要素名の前にアンダースコアを挿入します。

### Union データを含むサンプル ORC ファイルに基づいてモデルを作成しようとすると、モデルの作成に失敗します

インテリジェント構造検出は、ORC 入力の Union データ型を処理しません。モデルのベースとなる Union データを含まないファイルを選択します。

## 第 10 章

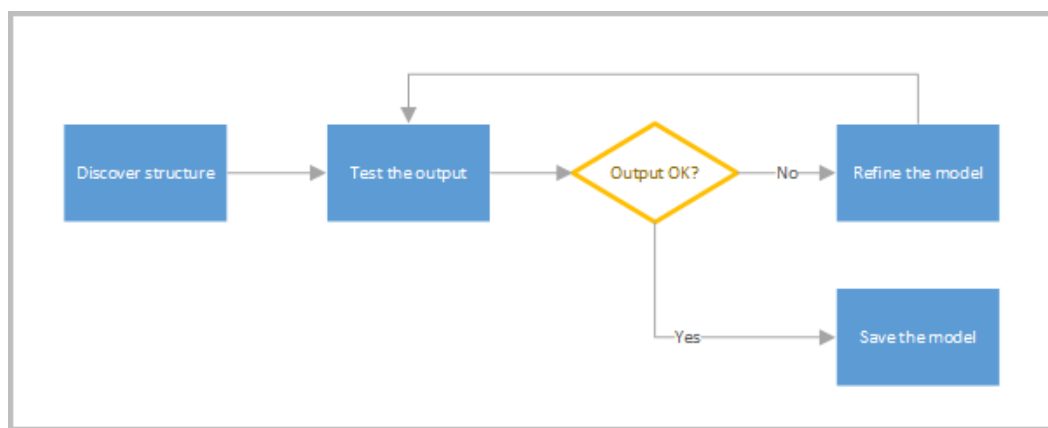
# インテリジェント構造モデルの改良

インテリジェント構造検出でモデル入力の構造が検出された後に、プロダクション環境でモデルを使用するときに出力が要件を満たすように構造を改良できます。

**[インテリジェント構造モデル]** ページを使用してモデルを改良します。例えば、ビジュアルモデルを使用して、構造の確認、ノードの検索と名前の変更、ファイル名へのプレフィックスの追加、または Microsoft Excel ファイルの行や列のテーブルヘッダーとしての定義を行うことができます。

検出された構造を改良すると、モデルに基づいてインテリジェント構造検出がさまざまな出力形式で生成した出力をテストできます。必要な出力が生成されるまで、モデルの改良とテストを繰り返します。

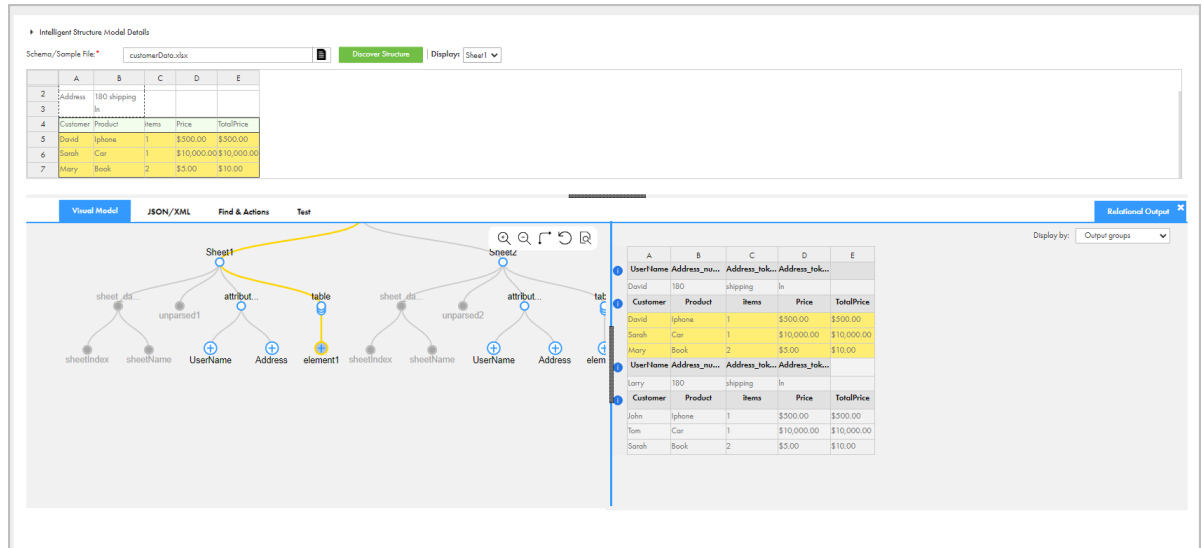
次の図は、モデルの検出、改良、テスト、および保存のプロセスを示しています。



# インテリジェント構造モデルのビュー

[インテリジェント構造モデル] ページには、モデルの視覚化、構造の調整、出力のテスト、および設定の構成に役立つさまざまなビューがあります。

次の図に、[インテリジェント構造モデル] ページのサンプルを示します。



入力ファイルをアップロードして [構造の検出] をクリックすると、パネルにデータファイルのプレビューが表示されます。サンプルファイルが複数のシートを含む Excel ファイルである場合は、どのシートをプレビューするかを選択できます。入力に長いデータ行が含まれている場合は、[単語の折り返し] を選択して各行を折り返すことができます。長い入力行を折り返さない場合、パネルには各行の一部のみが表示されます。テキストを折り返しても、インテリジェント構造や入力形式には影響しません。入力が XSD スキーマの場合、パネルは使用できません。

[インテリジェント構造モデル] ページのタブを使用して、検出された構造を調整します。次の表に、これらのタブを示します。

タブ	説明
ビジュアルモデル	ツリービューまたはフォルダビューで、検出された構造を確認します。ビジュアルモデルには、検出されたデータ型がノードとして表示され、ルートノードの変更、ノードの名前変更、およびノードの分割などのアクションをノード上で実行できます。
リレーショナル出力	インテリジェント構造が生成する出力グループを編集します。
JSON/XML	検出された構造を JSON または XML 形式で表示します。
検索とアクション	ノードやフィールドなどの構造要素を検索し、複数のノードでアクションを実行します。

タブ	説明
テスト	インテリジェント構造検出によってモデルに基づいて生成される出力をテストします。
設定	XML 入力ファイルおよび XSD 入力ファイルの設定を行います。 ファイルの構造を検出する前に、インテリジェント構造検出で出力グループを定義する方法を設定できます。詳細については、「 <a href="#">出力グループ定義</a> (ページ 55)」を参照してください。 ファイルの構造を検出したら、インテリジェント構造モデルで先頭と末尾のスペースを処理する方法を設定できます。デフォルトでは、モデルの先頭のスペースと末尾のスペースはすべてトリミングされます。

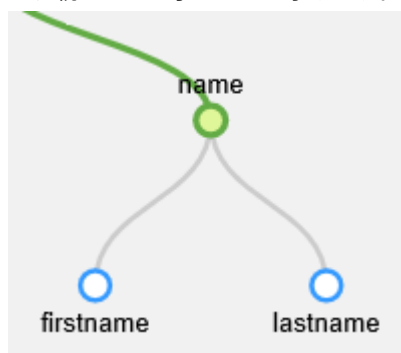
## ビジュアルモデルの操作

**【ビジュアルモデル】** タブを使用して、インテリジェント構造モデルの構造を表示します。ビジュアルモデルには、検出されたデータの型がノードとして表示され、互いのリレーションがグラフィカル形式で表示されます。

モデルをドラッグして、特定のセクションに焦点を合わせることができます。次のようなアクションを使用して構造を移動することもできます。

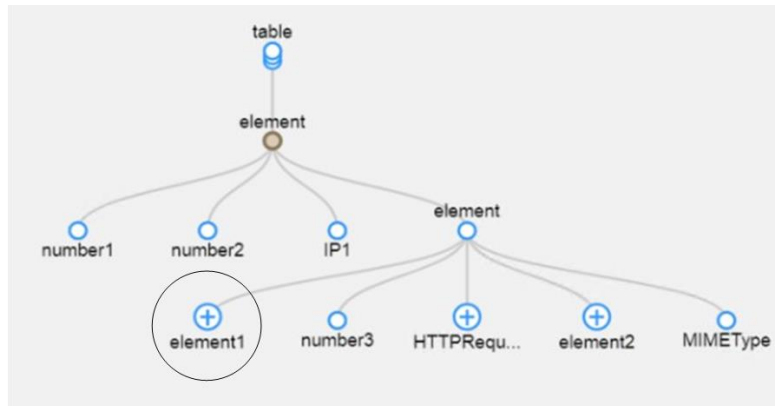
- エクスポート。モデルをエクスポートします。適切なライセンスがある場合に利用できます。
- ズームインとズームアウト。モデルの表示サイズを大きくしたり、小さくしたりします。
- 回転。フォルダビューとツリービューの間で表示を切り替えます。
- 元に戻す。最後のアクションを元に戻します。
- 検索。**【検索とアクション】** タブを開きます。

ツリービューでは、データフィールドのグループが子ノードを持つノードとして表示されます。例えば、入力データに Tatanya Morales などのフルネームが含まれている場合があります。インテリジェント構造モデルは、name という親ノードを持つ名前、名に対する子ノード、および姓に対する子ノードで名前を表します。次の図に、親ノードと子ノードを示します。



インテリジェント構造モデルに大量の子ノードを持つ親ノードが含まれている場合は、親ノードを折りたたむことができます。ノードを縮小するには、ノードをクリックして **【縮小】** を選択します。ビジュアルモデルで

は子ノードが非表示になりますが、インテリジェント構造モデルには子ノードが表示されたままになります。次の図に、ノードを縮小したモデルを示します。



注: モデルが複数のシートを含む Excel ファイルに基づいている場合、ビジュアルモデルにはすべてのシートが表示されます。

## リレーショナル出力の表示

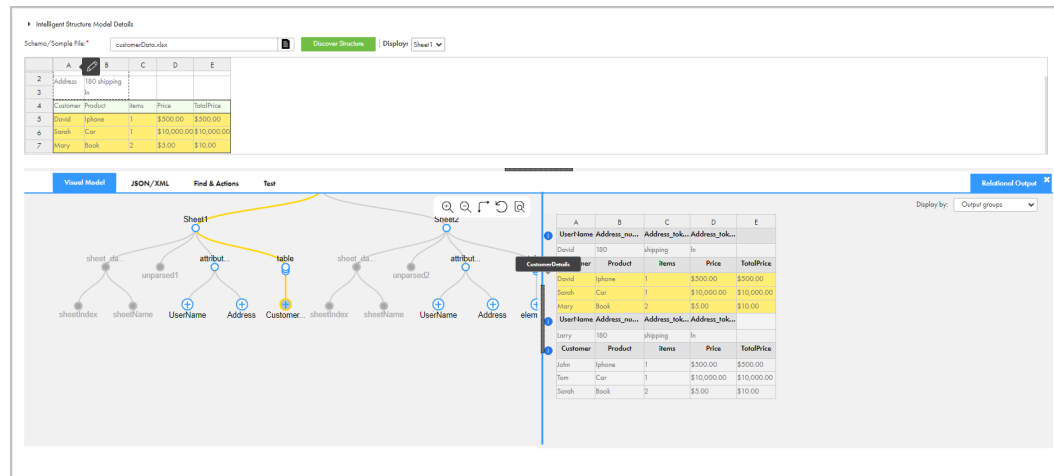
[リレーショナル出力] タブには、インテリジェント構造モデルが生成した出力が表示されます。

出力は、1 つ以上の出力グループに編成されます。出力グループには、インテリジェント構造モデルからの 1 つ以上のノードが含まれます。各出力グループには、カラムの左側に情報アイコンが表示されます。情報アイコンの上にカーソルを合わせると、出力グループ全体が強調表示され、出力グループの名前が表示されます。次の図に、強調表示された出力グループとグループ名を示します。

A	B	C	D	E		
①	UserName	Address_nu...	Address_tok...	Address_tok...		
	David	180	shipping	In		
	①	Customer	Product	items	Price	TotalPrice
	David	Iphone	1	\$500.00	\$500.00	
	Sarah	Car	1	\$10,000.00	\$10,000.00	
	Mary	Book	2	\$5.00	\$10.00	
	①	UserName	Address_nu...	Address_tok...	Address_tok...	
	Larry	180	shipping	In		
	①	Customer	Product	items	Price	TotalPrice

出力をさらに詳しく確認するには、ビジュアルモデルのノードを選択し、入力データパネルと出力グループの関連するカラムにある対応するデータを表示します。[ビジュアルモデル] タブでインテリジェント構造モデル

を改良すると、**[リレーショナル出力]** タブの出力グループに変更内容が表示されます。次の図に、強調表示された出力グループを示します。入力データパネルとビジュアルモデルで同じデータが強調表示されています。



**[リレーショナル出力]** タブの **[表示基準]** フィールドでは、次のいずれかの表示オプションを選択できます。

- 出力グループを表示する。出力でデータがどのようにグループ化されるかを確認できます。
- 受信レコードを表示する。Intelligent Structure Discovery が入力レコードを受信する方法を確認できます。

選択した表示オプションは、モデルを使用するマッピングを実行するときのデータの構造には影響しません。

**注:** **[リレーショナル出力]** タブには、最大 200 行のデータを表示できます。大量のデータまたは多数の繰り返しグループがある場合、**[リレーショナル出力]** タブにはすべての出力グループが表示されないことがあります。

## リレーショナル出力の操作

**[リレーショナル出力]** タブを使用して、モデルを使用するマッピングを実行するときのデータの構造に影響を与える出力グループに対してアクションを実行します。

**[リレーショナル出力]** タブでは、次のようなアクションを実行できます。

### データ正規化の設定

**[データの正規化]** フィールドを使用して、出力グループを正規化するか非正規化するかを選択します。

デフォルトでは、ネストされた繰り返しグループが JSON、XML、または XSD ファイルに基づくモデルに含まれている場合、インテリジェント構造検出は、入力データを正規化し、ネストされた各繰り返しグループをその出力グループに割り当てます。他の入力タイプに基づくモデルの場合、インテリジェント構造検出は、デフォルトではデータを正規化しません。

### ノードの名前の変更

ノードの名前を変更するには、ノードを表す出力グループ内のカラムを右クリックし、新しい名前を入力します。

### ノードの除外

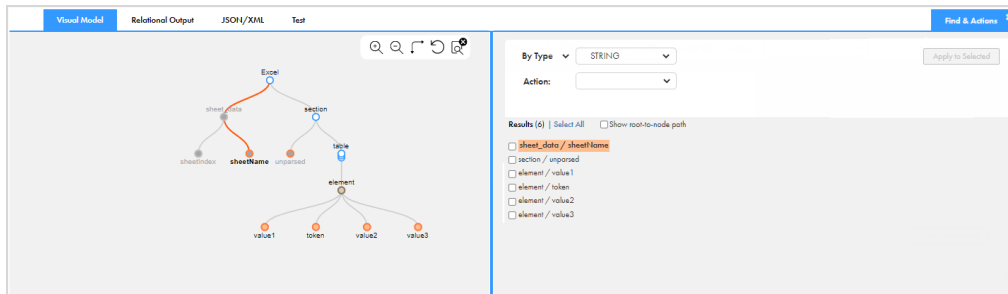
ノードを出力グループから除外するには、カラムを右クリックして **[構造から除外]** を選択します。ノードを除外すると、モデルを使用するマッピングを実行したときにデータは解析されません。

# 構造内のノードの検索

**【検索とアクション】** タブを使用して、構造内のノードを見つけます。名前またはデータ型でノードを検索できます。

1. **【インテリジェント構造モデル】** ページで、**【検索とアクション】** タブをクリックします。
2. ノードを検索するには、次のいずれかのアクションを実行します。
  - ノードを名前で検索するには、**【検索】** フィールドに検索する文字列を入力し、矢印をクリックします。
  - ノードをデータ型で検索するには、**【タイプ】** をクリックして、検索するデータ型を選択します。リストは基本タイプとセマンティックデータタイプに分かれており、モデルで使用されているすべてのデータ型が含まれています。

検索結果が**【結果】** 領域に表示され、見つかったノードが**【ビジュアルモデル】** タブのモデル内で強調表示されます。ノードごとに、**【結果】** 領域には、ノードへのパスの最後の2つのノードが表示されます。次の図は、STRING データ型で検索した結果を示しています。



3. 必要に応じて、ノードへのフルパスを表示するには、**【ルートからノードまでのパスを表示】** を選択します。

# ノードデータの表示とアクションの実行

アルゴリズム結果の精度、ノードに関連すると考えられるノードタイプ、およびノードに対応する入力ファイル内のサンプルデータなど、構造内のノードに関する情報を表示できます。

ノード内のデータに対してアクションを実行できます。実行できるアクションは、モデルのベースとなるサンプルデータのタイプとノードの出力データタイプによって異なります。

1. **【インテリジェント構造モデル】** ページの**【ビジュアルモデル】** タブでノードをダブルクリックするか、ノードを右クリックして、**【データを開く】** を選択します。

**["<ノード名>"のデータ]** ダイアログボックスが表示されます。
2. ノードのデータ型を変更するには、**【要素タイプ】** リストからデータ型を選択します。データタイプ選択は、Informatica Developer にエクスポートするインテリジェント構造モデルに適用されます。
3. ノードのデータ型に基づいて、次のアクションの1つ以上を実行します。
  - 型形式を変更するには、新しい形式を**【形式】** フィールドに入力します。
  - 精度ノードを変更するには、**【精度】** フィールドに値を入力します。

**注:** 既存のインテリジェント構造モデル内の要素の精度を変更しても、構造パーサートランスフォーマーションの出力グループ要素の順序は同じままです。
  - スケールを変更するには、**【スケール】** フィールドに値を入力します。
4. **【OK】** をクリックします。

# ノードの編集

モデルの出力は、インテリジェント構造ノードを編集することで変更できます。使用可能なオプションは、モデルのベースとなるサンプル入力のタイプおよびノードの出力データ型によって異なります。

**[ビジュアルモデル]** タブでノードに対してアクションを実行します。

次のアクションを実行して、ノードを編集します。

## ルートノードを変更する。

ルートを変更するには、新しいルートノードを右クリックして、**[ルートとしてマーク]** を選択します。ルートノードへの変更は、構造パーサトランスフォーメーションのXML、JSON、またはリレーショナル出力には影響しません。

## ノードの名前を変更する。

ノードの名前を変更するには、ノードを右クリックし、**[名前の変更]** を選択して、**Street Address** などの名前を入力します。ノード名では大文字小文字は区別されません。名前は、兄弟ノード、つまり同じノードの下にあるノード間で一意である必要があります。

## ノードへの一意の名前の適用

構造内で一意の名前を適用するには、ルートノードを右クリックして **[一意の名前の適用]** を選択します。一意の名前をモデルに適用する際に、サンプル入力内のデータに他のグループと同じ名前のフィールドが含まれる場合、インテリジェント構造検出は、サフィックスを割り当てて同一の名前を一意的な名前に変更します。

## 元のノード名をリストアする。

元のノード名をリストアし、インテリジェント構造検出によって、異なる階層レベル内の同一の名前に追加されるサフィックスを削除します。

構造内の元のノード名をリストアするには、構造のルートノードを右クリックし、**[元の名前に戻す]** を選択します。

**注:** リレーショナル出力を使用したデータ統合マッピングで使用する予定のモデルの元のノード名はリストアしないでください。こうした条件において、ノード名は構造全体で一意的にする必要があります。

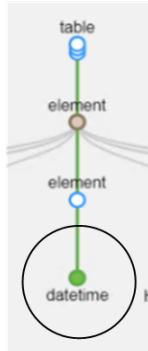
## 2つのノードのデータを結合する。

2つのノードを結合するには、1つのノードをクリックしてもう1つのノードにドラッグします。例えば、次のモデルでは、**time** ノードを選択して、**date** ノードにドラッグしています。





次の図は、結合後のノードを示しています。



#### すべての子ノードデータを親ノードに結合する。

親ノードをフラット化してその子ノードをマージするには、ノードを右クリックして **【フラット化】** を選択します。ノードをフラット化すると、子ノードのデータが、親ノードの1つの出力フィールドにマージされます。出力では、子ノードは別のフィールドではなくなります。

#### ノードを構造から除外する。

構造からノードを除外するには、ノードを右クリックし、**【構造から除外】** を選択します。構造からノードを除外すると、ノードのデータはモデルの出力の一部ではなくなり、実行時に解析されません。

#### ノードを構造に含める。

除外されたノードを再度構造に含め、インテリジェント構造検出がデフォルトでは構造に含めないノードを含めることができます。例えば、Excel スプレッドシートの名前を構造に含めることができます。ノードを構造に含めるには、ノードを右クリックし、**【構造に含める】** を選択します。

#### インテリジェント構造検出が Microsoft Excel ファイルのデータを解析する方法を指定します。

インテリジェント構造検出で、Microsoft Excel に表示されるフォーマット済みのデータを解析するか、または Microsoft Excel ファイルのフィールドの解析時にベースとなるフォーマットされていないデータを解析するかを指定できます。

インテリジェント構造検出がデータを解析する方法を指定するには、ノードを右クリックして、次のオプションのいずれかを選択します。

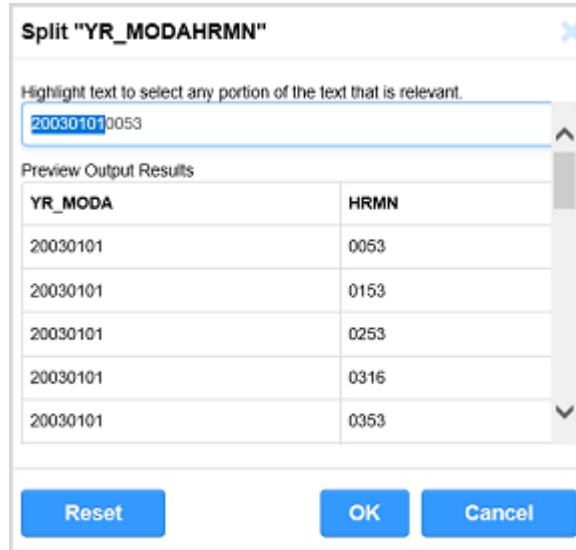
- フォーマット済みのデータを抽出する。インテリジェント構造検出によって、フィールドがフォーマット済みのデータとして解析されます。
- フォーマットされていないデータを抽出する。インテリジェント構造検出によって、フィールドがフォーマットされていないデータとして解析されます。

#### ノードを分割する。

固定幅ファイルに基づく構造でノードを分割できます。ノード内のデータを2つのノードに分割する場合、例えば、クレジットカード番号の最後の4桁のみを保存するには、次の手順を実行します。

1. ノードを右クリックして **【分割】** を選択します。**【分割】** ダイアログボックスが表示されます。
2. **【分割】** ダイアログボックスで、表示フィールドのデータを強調表示し、データを分割する場所を示します。データが分割され、2つのフィールド（強調表示されたデータを含むフィールドと、強調表示されないデータを含むフィールド）として表示されます。
3. 分割を承認するには、**【OK】** をクリックします。
4. 分割を取り消すには、**【リセット】** をクリックします。

例えば、1つのフィールドに「year-month-date-hour-minute」というデータがあるノードを、2つのフィールドに分割するとします。「year-month-date」フィールドと「hour-minute」フィールドを作成します。次の図は、分割対象として強調表示した部分を示しています。



#### JSON モデルを繰り返しとして処理

デフォルトでは、サンプル JSON モデルに繰り返しが含まれない場合、構造パーサトランスフォーマーションは繰り返しグループの解析を行いません。繰り返しグループの解析を行う場合、JSON ルートノードを右クリックして **[定期として処理]** を選択します。

#### 要素を繰り返しとして処理

デフォルトでは、サンプル JSON または XML モデルの要素には繰り返しは含まれず、構造パーサトランスフォーマーションは要素の繰り返しの解析を行いません。要素の繰り返しの解析を行う場合、ノードを右クリックして **[リストとして処理]** を選択します。**[リストとして処理]** アクションを元に戻すには、ノードを右クリックして **[単発として処理]** を選択します。

## 繰り返しグループの操作

グループの出力はリレーショナルテーブルです。インテリジェント構造に、互いに関連する繰り返しグループがある場合は、ネストされたグループに対する異なるテーブルを含むようにリレーショナル出力を設定し、レコード ID を選択できます。

インテリジェント構造に、ネストされた繰り返しグループがある場合、**[ビジュアルモデル]** タブで次のアクションを実行できます。

#### グループに昇格。

モデル内のポートの数を減らすには、ネストされた各繰り返しグループを独自の出力グループに割り当てます。繰り返しグループ内でネストされたグループを昇格すると、Intelligent Structure Discovery は親出力グループにレコード ID を作成し、子出力グループに外部キーを生成します。グループを独立した出力グループに昇格するには、ネストされた繰り返しノードを右クリックし、**[グループに昇格]** を選択します。

**注:** JSON、XML、または XSD ファイルに基づくモデルの場合、インテリジェント構造検出は、モデルの作成時に、ネストされた繰り返しグループを独自の出力グループに割り当てます。

### 親グループに参加。

ネストされたグループを、自身の出力グループに割り当てた後、親グループに再び含めます。グループを再び含めるには、ネストされた繰り返しノードを右クリックし、**【親グループに参加】** を選択します。

### レコード ID として選択。

レコード ID を変更します。親グループのノードを選択し、レコード ID として割り当てます。Intelligent Structure Discovery は、ネストされたグループの外部キーを自動的に変更します。ノードをレコード ID として割り当てするには、ノードを右クリックして、**【レコード ID として選択】** を選択します。

### レコード ID として選択解除。

最初のレコード ID と外部キーを繰り返しグループに戻します。ノードをレコード ID として選択解除するには、ノードを右クリックして、**【レコード ID として選択解除】** を選択します。

## モデル内のデータへのドキュメント識別子の追加

JSON、XML、または XSD ファイルに基づくモデルに非反復の出力グループが含まれている場合、インテリジェント構造検出は、非反復の各出力グループにドキュメント識別子を追加します。

他の入力タイプに基づくモデルの場合は、ドキュメント識別子を出力グループに手動で追加できます。

例えば、構造パーサートランスフォーメーションを同じ入力に基づく出力グループに参加させてから単一のターゲットに出力する場合に、ドキュメント識別子をモデルに追加するとします。ドキュメント識別子を使用して、ジョイナトランスフォーメーションでグループに参加することもできます。

ドキュメント識別子を出力グループに追加するには、グループを表すノードを右クリックして、**【ドキュメント ID を追加】** を選択します。識別子を削除するには、ノードを右クリックして **【ドキュメント ID を削除】** を選択します。

モデルでは、**【リレーショナル出力】** タブにドキュメント識別子フィールドが「<グループ名>\_documentId」という形式で表示されます。例えば、「organization」という名前のグループの場合、ドキュメント識別子フィールドは「organization\_documentId」となります。

構造パーサートランスフォーメーションで、各出力グループに対するドキュメント識別子の出力フィールドが表示されます。

## フィールド名へのプレフィックスとサフィックスの追加

検出された構造内の 1 つのフィールド名または複数のフィールド名にプレフィックスあるいはサフィックスを追加できます。

プレフィックスとサフィックスには、英数字とアンダースコアを含めることができます。

1. **【検索とアクション】** タブで、フィールド名またはフィールド名の一部を検索します。  
検索と一致したフィールド名が **【検索とアクション】** タブに表示されます。
2. プレフィックスまたはサフィックスを追加するフィールドを選択し、**【アクション】** メニューから **【プレフィックスを追加】** あるいは **【サフィックスを追加】** を選択します。

3. **【アクション】** メニューの右にあるテキストフィールドにプレフィックスまたはサフィックスを入力し、**【選択に適用】** をクリックします。  
インテリジェント構造検出で、選択したフィールドの名前にプレフィックスまたはサフィックスが追加されます。

## 複数ノードでのアクションの実行

複数のノードでアクションを実行できます。実行可能なアクションは、モデルのベースとなるサンプルファイルのタイプとノードの出力データタイプによって異なります。

1. **【検索とアクション】** タブでノードを検索し、**【結果】** リストでアクションを実行するノードを選択します。
2. **【アクション】** リストから、以下のいずれかのアクションを選択します。

オプション	説明
プレフィックスの追加	ノード名にプレフィックスを追加します。 <b>【権限】</b> フィールドにプレフィックスを入力します。
サフィックスの追加	ノード名にサフィックスを追加します。 <b>【権限】</b> フィールドにサフィックスを入力します。
除外	インテリジェント構造からノードを除外します。ノードのデータはモデルの出力の一部ではなくなり、実行時に解析されません。
含む	除外されたノードを構造に再度含めます。
展開	ノードを展開して、ノードの下に折りたたまれているモデルの部分を表示します。
縮小	ノードの下にネストされているモデルの部分を非表示にします。
データ型の更新	ノードのデータ型を変更します。 <b>【変更先】</b> フィールドでデータ型を選択します。選択したデータ型に基づいて、ノードの精度、スケール、または型形式を編集できます。データタイプ選択は、Informatica Developer にエクスポートするインテリジェント構造モデルに適用されます。
子プレフィックスの追加	子ノード名にプレフィックスを追加します。 <b>【権限】</b> フィールドにプレフィックスを入力します。 <b>注:</b> このコマンドでは、直接の子ノードにのみプレフィックスが適用されます。ノードパスのさらに下にある子ノードにはプレフィックスが適用されません。
子サフィックスの追加	子ノード名にサフィックスを追加します。 <b>【権限】</b> フィールドにサフィックスを入力します。 <b>注:</b> このコマンドでは、直接の子ノードにのみサフィックスが適用されます。ノードパスのさらに下にある子ノードにはサフィックスが適用されません。
タイプを文字列に変更	ノードのデータ型を文字列に変更します。

オプション	説明
置換	ノード名の文字列を、 <b>【置換先】</b> フィールドに入力した文字列に置き換えます。ノード名では大文字小文字は区別されません。名前は、兄弟ノード、つまり同じノードの下にあるノード間で一意である必要があります。
コンテンツでデータ型を検出	データを含むノードの内容に基づいて、二重引用符で囲まれた数値のデータ型を検出します。JSON ファイルに適用されます。 <b>注:</b> このアクションを 1 つのノードまたは複数のノードに適用すると、二重引用符で囲まれた数字を文字列として識別するデフォルトモードに戻すことは選択できなくなります。デフォルトモードを再適用するには、サンプルファイルを選択して、構造を再度検出する必要があります。
フォーマットされたデータを抽出	Microsoft Excel が表示するフォーマットされたデータを解析します。Microsoft Excel ファイルに適用されます。
フォーマットされていないデータを抽出	基になるフォーマットされていないデータを解析します。Microsoft Excel ファイルに適用されます。

3. **【選択に適用】** をクリックします。

## 新しいサンプルでの既存のモデルのエンリッチ化

JSON、XML、ORC、AVRO、または PARQUET サンプルファイルに基づいてモデルを作成した後に、追加のサンプルファイルを使用して、新しいサンプルに存在するフィールドで構造をエンリッチ化できます。

例えば、マップするデータに構造内にないデータが含まれ、そのためマップされない場合、そのデータを含む新しいサンプルファイルに基づいた構造にそのデータを追加する事ができます。

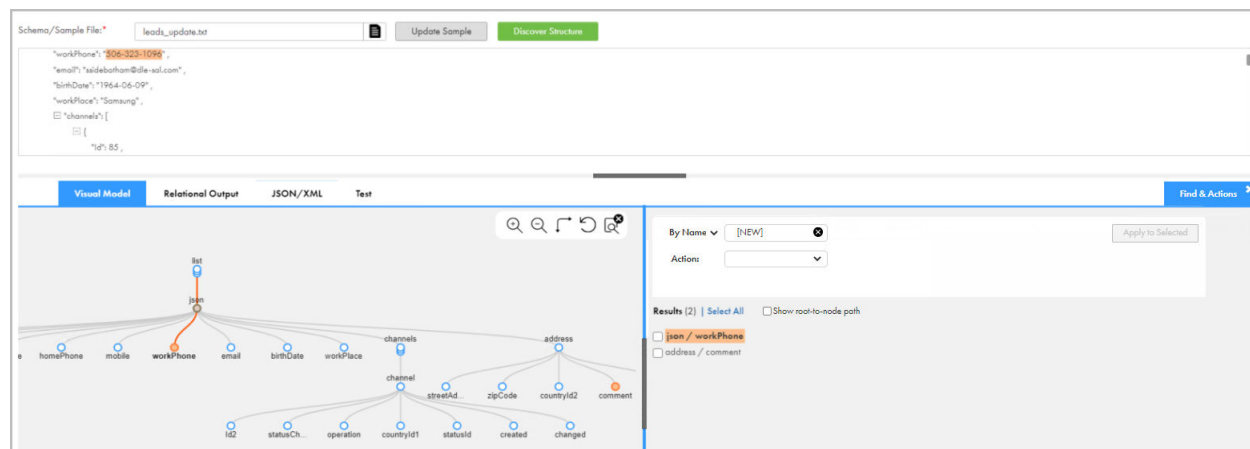
構造をエンリッチ化するために使用するファイルは、構造が基盤とするものと同じファイルタイプでなければなりません。例えば、JSON サンプルファイルに基づく構造をエンリッチ化するには、追加の JSON サンプルファイルを使用します。

構造に新しいサンプルファイルを追加するには、**【サンプルの更新】** をクリックして新しいサンプルファイルを選択します。

インテリジェント構造検出は、サンプルファイルに新しいデータのノードを作成します。新しいノードの検索を容易にするには、検索パターン**[NEW]**で **【検索とアクション】** タブを開きます。新しいノードのリストが表示され、これらのノードが構造内で強調表示されます。

**注:** 新しいサンプルに構造内にあったすべてのデータが含まれていない場合、データは構造から削除されず構造全体が表示されます。ただし、入力データパネルと **【リレーショナル出力】** タブには新しいサンプルにあるデータのみが表示されます。

次の画像は、新しいサンプルのデータを使用して更新されたモデルを示しています。



モデルを保存すると、構造は新しいデータで更新されます。

## Microsoft Excel 入力の構造の編集

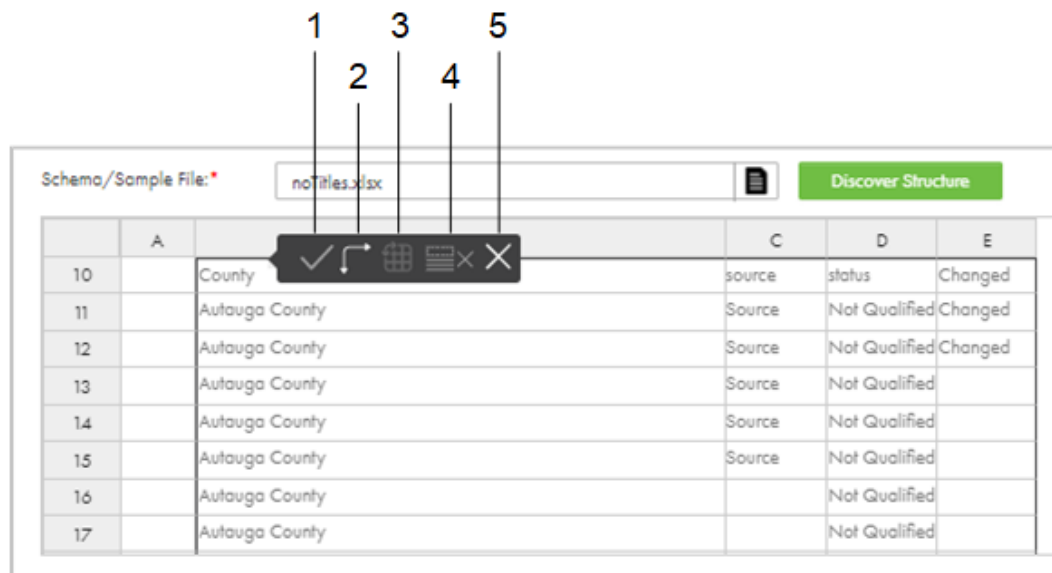
モデルの入力が Microsoft Excel ファイルの場合、デフォルトでは、インテリジェント構造検出は、Excel スプレッドシートのメタデータ（スプレッドシートの名前やインデックス番号など）を構造に含めません。

スプレッドシートの名前やインデックス番号を構造に含めるには、**[インテリジェント構造モデル]** ページの **[ビジュアルモデル]** タブで、sheetName ノードまたは sheetIndex ノードをそれぞれ右クリックし、**[構造に含める]** を選択します。

インテリジェント構造検出により、**[インテリジェント構造モデル]** ページの入力データパネルにモデルデータがテーブル形式で表示されます。テーブルにカーソルを合わせると、**[編集]** ボタンが表示されます。ボタンをクリックすると、テーブルで次のアクションを実行するために使用できるメニューが開きます。

- テーブルを転置します。これは、テーブルヘッダーの方向を行からカラムに、またはカラムから行に変更することを意味します。
- 2カラムまたは2行のテーブルを名前と値のペアに変換し、名前と値のペアをテーブルに変換します。
- テーブルヘッダーを定義します。

次の図は、Microsoft Excel 入力ファイルの場合の入力データパネルの例と、テーブルの編集に使用するメニューを示しています。



1. 適用
2. 置き換え
3. 名前と値のペアに変換/テーブルに変換
4. すべてのヘッダーの削除
5. キャンセル

## テーブルの転置

Microsoft Excel ファイルに基づくモデルのテーブルを転置できます。

1. **【インテリジェント構造モデル】** ページの入力データパネルで、テーブルにカーソルを合わせ、**【編集】** ボタンをクリックします。
2. **【置き換え】** ボタンをクリックします。  
 テーブルの向きが変わります。変更は、入力データパネルと **【ビジュアルモデル】** タブのモデルに表示されます。

## テーブルと名前と値のペアの切り替え

Microsoft Excel ファイルに基づくモデルでは、2 カラムまたは 2 行のテーブルと名前と値のペアを切り替えることができます。

1. **【インテリジェント構造モデル】** ページの入力データパネルで、テーブルにカーソルを合わせ、**【編集】** ボタンをクリックします。
2. 必要に応じて、**【名前と値のペアに変換】** ボタンまたは **【テーブルに変換】** ボタンをクリックします。  
**注:** **【テーブルに変換】** ボタンは、2 カラムおよび 2 行のテーブルに対してのみ有効になります。それより大きなテーブルを名前と値のペアに変換することはできません。  
 修正された構造は、入力データパネルと **【ビジュアルモデル】** タブのモデルに表示されます。

## テーブルヘッダーの定義

Microsoft Excel ファイルに基づくモデルでは、行とカラムをテーブルヘッダーとして定義できます。

1. **【インテリジェント構造モデル】** ページの入力データパネルで、テーブルにカーソルを合わせ、**【編集】** ボタンをクリックし、必要な行またはカラムを選択します。

選択した行またはカラムが緑色で表示されます。

	A		C	D	E
10		County	source	status	Changed
11		Autauga County	Source	Not Qualified	Changed
12		Autauga County	Source	Not Qualified	Changed
13		Autauga County	Source	Not Qualified	
14		Autauga County	Source	Not Qualified	
15		Autauga County	Source	Not Qualified	
16		Autauga County		Not Qualified	
17		Autauga County		Not Qualified	

2. **【適用】** ボタンをクリックします。

選択した行またはカラムは、**【ビジュアルモデル】** タブでモデルのヘッダーとして表示されます。

3. テーブルヘッダーの選択をクリアするには、入力データパネルで **【すべてのヘッダーの削除】** ボタンをクリックします。

## 出力のテスト

モデルに指定した入力の構造をインテリジェント構造検出が検出すると、モデルの出力をテストできます。モデルに対するさまざまな出力形式をテストできます。

1. **【インテリジェント構造モデル】** ページで、**【テスト】** タブを選択します。
2. **【テスト出力フォーマット】** リストから、テストする出力フォーマットを選択します。
3. **【選択して変換】** をクリックし、テストする入力ファイルを選択して、**【開く】** をクリックします。

インテリジェント構造検出は、出力をデフォルトのダウンロードフォルダの.zip ファイルに保存します。

4. .zip ファイルを解凍し、出力ファイルを開いて、インテリジェント構造検出によって生成された出力を表示します。入力に未確認データが含まれる場合、インテリジェント構造検出は未確認データを別のファイルに保存します。



# 第 11 章

## マップレット

マップレットは、ターゲットに読み込まれる前にソースデータを変換するために使用できるトランスフォーメーションロジックです。Informatica PowerCenter ユーザーの場合は、PowerCenter マップレットをインポートしてデータ統合で使用できます。

マップレットは、次のいずれかの方法で作成できます。

- データ統合でマップレットを作成します。
- PowerCenter からマップレットをインポートします。マップレットのインポートの詳細については、[「PowerCenter マップレット」 \(ページ 88\)](#)を参照してください。
- SAP BAPI または IDoc マップレットを生成します。SAP マップレットの設定とインポートの詳細については、SAP コネクタのヘルプを参照してください。

Mapping Designer でマッピングを作成する場合と同じ方法で、データ統合の Maplet Designer でマップレットを作成します。マップレットを作成した後に、マップレットトランスフォーメーションにマップレットを追加してそのトランスフォーメーションロジックを使用できます。マップレットでは、Web サービストランスフォーメーションを除く任意のトランスフォーメーションを使用できます。

マップレットは別のマップレットで使用することもできます。ただし、マップレットを循環参照することはできません。例えば、マップレット A がマップレット B を使用している場合、マップレット B がマップレット A を使用することはできません。

マップレットは、SQL ELT モードのマッピングでは使用できません。

## アクティブなマップレットとパッシブなマップレット

マップレットはアクティブかパッシブになります。

パッシブなマップレットには、単一の入力グループと単一の出力グループ、およびパッシブなトランスフォーメーションのみが含まれます。

アクティブなマップレットには、少なくとも 1 つのアクティブなトランスフォーメーションが含まれます。

マップレットトランスフォーメーションのマップレットを使用する場合は、マップレットのタイプが **[マップレット]** タブに表示されます。

# マップレットの入力と出力

マップレットトランスフォーメーションでマップレットを使用するには、マップレットの入力と出力を設定する必要があります。

マップレットは、入力トランスフォーメーションを介したアップストリームトランスフォーメーション、ソーストランスフォーメーション、またはその両方から入力を受け取ります。

マップレットは、出力トランスフォーメーションを介したダウンストリームトランスフォーメーション、ターゲットトランスフォーメーション、またはその両方に出力を渡します。

マップレットには少なくとも1つの入力トランスフォーメーションまたは出力トランスフォーメーションを含める必要があります。

## マップレットの入力

マップレットの入力は、入力トランスフォーメーション、ソーストランスフォーメーション、またはその両方です。

1つ以上のアップストリームトランスフォーメーションから入力データを受け取るようにマップレットを設定するには、入力トランスフォーメーションを使用します。1つのマップレットには、複数の入力トランスフォーメーションを使用できます。マップレットトランスフォーメーションでマップレットを使用する際、各入力トランスフォーメーションが1つの入力グループになります。マップレットに複数のパイプラインがある場合、または複数のアップストリームトランスフォーメーションから入力を受け取るようにマップレットを設定する場合は、複数の入力トランスフォーメーションを使用します。

ソースデータを提供するには、マップレットに1つ以上のソーストランスフォーメーションを含めます。マップレット入力にソーストランスフォーメーションのみを使用する場合、マップレットがマッピングパイプライン最初のオブジェクトとなり、入力グループはありません。ソーストランスフォーメーションが階層データを読み取る場合は、ソーストランスフォーメーションで検証タイプを指定する必要があります。

マップレットには少なくとも1つの入力トランスフォーメーションまたはソーストランスフォーメーションを含める必要があります。

**注:** マップレットのソーストランスフォーメーションでは、すべてのソースタイプを使用できるわけではありません。詳細については、該当するコネクタのヘルプを参照してください。

入力トランスフォーメーションとソーストランスフォーメーションの設定方法の詳細については、「トランスフォーメーション」を参照してください。

## マップレットの出力

マップレットの出力先は、出力トランスフォーメーション、ターゲットトランスフォーメーション、またはその両方です。

1つ以上のダウンストリームトランスフォーメーションにデータを渡すようにマップレットを設定するには、出力トランスフォーメーションを使用します。マップレットトランスフォーメーションでマップレットを使用する際、各出力トランスフォーメーションが1つの出力グループになります。各出力グループは、マッピング内の1つ以上のパイプラインにデータを渡すことができます。

ターゲットにデータを書き込むようにマップレットを設定するには、ターゲットトランスフォーメーションを使用します。出力トランスフォーメーションなしでターゲットトランスフォーメーションを使用する場合、マッピングは、マッピングパイプラインで最後のオブジェクトになります。

マップレットには少なくとも1つの出力トランスフォーメーションまたはターゲットトランスフォーメーションを含める必要があります。

出力トランスフォーメーションとターゲットトランスフォーメーションの設定方法の詳細については、「トランスフォーメーション」を参照してください。

# トランスフォーメーションの名前

マップレットトランスフォーメーションでマップレットを使用すると、データ統合の実行時にマップレットのトランスフォーメーションの名前が変更されます。

マップレットトランスフォーメーションでマップレットを使用すると、マップレットに、マッピング内の他のトランスフォーメーション名と競合する名前のトランスフォーメーションが含まれる場合があります。名前の競合を避けるため、データ統合では、マップレット内のトランスフォーメーションの名前の前にマップレットトランスフォーメーション名が付与されます。

例えば、Expression\_1 という名前の式トランスフォーメーションを含むマップレットがあるとします。マッピングを作成し、マップレットトランスフォーメーション Mapplet\_Tx\_1 でマップレットを使用します。マッピングを実行すると、式トランスフォーメーションの名前が Mapplet\_Tx\_1\_Expression\_1 に変更されます。

マップレットに別のマップレットトランスフォーメーションがある場合、データ統合では、トランスフォーメーション名の先頭に 2 つ目のマップレットトランスフォーメーション名が付与されます。例えば、前の例のマップレットに MappletTx という名前のマップレットトランスフォーメーションがあり、これに FilterTx\_1 が含まれているとします。マッピングでは、FilterTx\_1 は、Mapplet\_Tx\_1\_MappletTx\_FilterTx\_1 に変更されます。

データ統合では、80 文字を超えるトランスフォーメーション名は切り詰められます。

**注:** データ統合によってマップレットのトランスフォーメーション名が変更されるのは、マップレットが 2022 年 4 月リリース以降に作成されたマッピングで使用されている場合のみです。

## マップレットのパラメータ

マップレットでは、入力パラメータと入出力パラメータを使用できます。パラメータの値は、マッピングタスクを設定するときに指定します。

マップレットのパラメータは、マッピングのパラメータを設定する場合と同じ方法で設定します。パラメータの設定の詳細については、「マッピング」を参照してください。

マップレットにパラメータを含めると、マップレットトランスフォーメーションでそのマップレットを使用するときに、データ統合によってパラメータの名前が変更されます。パラメータ名の先頭には、マップレットトランスフォーメーションの名前が付きます。対応する名前は、マップレットトランスフォーメーションの [プロパティ] パネルにある [パラメータ] タブで確認できます。

例えば、マップレットのフィルタトランスフォーメーションを使用して、フィルタ条件として **[すべてパラメータを使用します]** を選択します。「Filter\_Param」という名前の文字列パラメータを作成します。マップレットトランスフォーメーションで「MyMappletTx」という名前のマップレットを使用します。マップレットトランスフォーメーションで、フィルタパラメータの名前が「MyMappletTx\_Filter\_Param」に変更されます。

マップレットで入出力パラメータを定義すると、入出力パラメータがマップレットで使用されていない場合でも、詳細モードのマッピングに対して無効になります。

# マップレットの作成

Mapplet Designer ではマップレットを作成できます。

Mapplet Designer は、Mapping Designer と同じ設計領域と機能を備えています。Mapping Designer の使い方の詳細については、「マッピング」を参照してください。

1. **【新規】** > **【マップレット】** > **【マップレット】** をクリックします。

Mapplet Designer のキャンバスに、入力トランスフォーメーションと出力トランスフォーメーションが表示されます。

2. マップレットの名前と場所を指定します。

マップレットのデフォルト名は、「Mapplet」の後に連番が付いた名前になります。

**【エクスプローラ】** ページが現在アクティブになっていて、プロジェクトまたはフォルダが選択されている場合、アセットのデフォルトの場所はその選択されているプロジェクトまたはフォルダです。そうでない場合、デフォルトの場所は直近で保存されたアセットの場所です。

**【エクスプローラ】** ページを使用して、マッピングを保存した後で、名前または場所を変更できます。

3. 必要に応じて、マップレットの説明を入力します。

マップレットトランスフォーメーションのマップレットを使用する場合は、説明が**【マップレット】** タブに表示されます。

4. 必要に応じて、変換パレットでトランスフォーメーションをフィルタリングします。

5. マップレット入力を設定します。

入力は、ソーストランスフォーメーションか入力トランスフォーメーション、またはその両方です。入力トランスフォーメーションを使用する場合は、マップレットに渡すフィールドについて入力フィールドを追加します。

6. マップレットにトランスフォーメーションを追加します。

7. マップレットの出力を設定します。

入力は、ターゲットトランスフォーメーションか出力トランスフォーメーション、またはその両方です。出力トランスフォーメーションを使用する場合は、ダウンストリームトランスフォーメーションに渡すデータについて出力フィールドを追加します。

8. マップレットを検証します。

9. **【検証】** パネルの該当するタブでエラーを解決します。

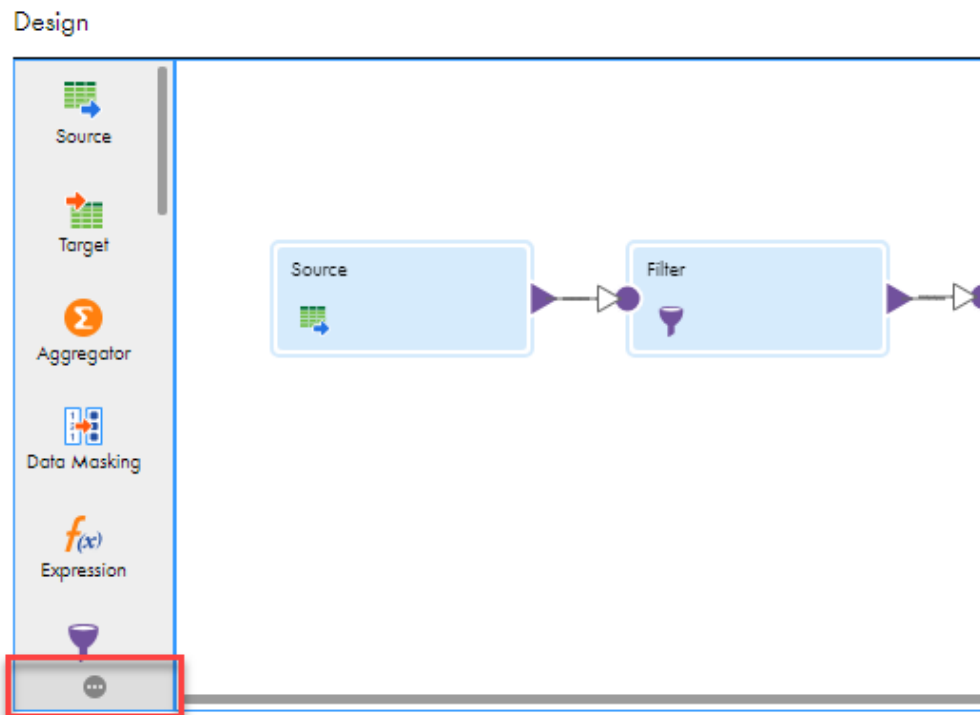
- マップレットを詳細モード以外で使用するには、**【マッピング】** タブでエラーを解決します。
- マップレットを詳細モードで使用するには、**【マッピング (詳細モード)】** タブでエラーを解決します。

10. マップレットを保存します。

# トランスフォーメーションパレットのフィルタリング

Mapplet Designer でトランスフォーメーションパレットのフィルタリングを使用することで、ライセンス取得済みトランスフォーメーション、および詳細モードと詳細モード以外で使用可能なトランスフォーメーションを表示できます。

1. Mapplet Designer で、トランスフォーメーションパレットの **[詳細]** アイコンをクリックします。  
次の図は、そのアイコンを示しています。



2. **[トランスフォーメーションパレットのフィルタ]** ダイアログボックスで、次のいずれかのマッピングオプションを選択します。
  - マッピング。詳細モード以外で使用可能なトランスフォーメーションを表示します。
  - マッピング (詳細モード)。詳細モードで使用可能なトランスフォーメーションを表示します。
  - 共有。詳細モードと詳細モード以外で使用可能なトランスフォーメーションを表示します。
  - すべて。すべてのトランスフォーメーションを表示します。
3. 次のいずれかのライセンスオプションを選択します。
  - ライセンス取得済み。組織で使用可能なトランスフォーメーションを表示します。
  - すべて。すべてのトランスフォーメーションを表示します。

# マップレットの編集

Mapplet Designer ではマップレットを編集できます。マップレットを編集するには、[エクスプローラ] ページからマップレットを開きます。

マップレットを編集する際、Mapplet Designer によってトランスフォーメーションのロジックが検証されません。マップレットの保存時にマップレットが詳細モードのマッピングまたは詳細モード以外のマッピングで有効な場合、データ統合は、そのマップレットを使用する各マッピングおよびマップレットに変更をデプロイします。

例えば、マップレットが詳細モードおよび詳細モード以外で有効で、マップレットのインタフェースを変更して詳細モード以外のみでマップレットが有効になるようにする場合、データ統合は、そのマップレットを使用するすべてのマッピングへの変更をデプロイします。

マッピングや他のマップレットで使用されるマップレットを編集する際は、慎重に使用する必要があります。マップレットインタフェースを編集すると、そのマップレットを使用するマッピングとマップレットで検証エラーが発生します。検証エラーを解決するには、マップレットを使用して最新の変更を取得するマップレットトランスフォーメーションを同期する必要があります。マップレットトランスフォーメーションを同期しないでタスクを実行すると、タスクは失敗します。

マップレットを使用するアセットを表示するには、Mapplet Designer で **[アクション]** メニューを開き、**[依存関係の表示]** を選択します。

マップレットトランスフォーメーションでマップレットを編集することはできず、PowerCenter からインポートしたマップレットを編集することもできません。

## 依存関係に影響する変更

マップレットを変更すると、そのマップレットを使用するマッピングとマップレットに影響を及ぼす可能性があります。

マップレットに対する次の変更は、そのマップレットを使用するアセットに影響を及ぼしません。

- トランスフォーメーションの名前、説明、またはプロパティの変更。
- マップレット内のトランスフォーメーションの追加または削除（ただしマップレットタイプをアクティブからパッシブまたはパッシブからアクティブに変更しない場合）。

マップレットに対して次のような変更を行うと、マップレットのインタフェースが変更され、そのマップレットを使用するアセットに影響を及ぼします。

- マップレットタイプをアクティブからパッシブまたはパッシブからアクティブに変更する、トランスフォーメーションの追加または削除。
- 関連付けられた入力フィールドまたは出力フィールドのデータ型、精度、またはスケールの変更。
- 入力フィールドまたは出力フィールドの追加または削除
- 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追加または削除

## マップレットの同期

マップレットのインタフェースが、マップレットトランスフォーメーションに追加された後で変更された場合は、そのマップレットを同期して変更を反映する必要があります。マップレットの同期は、**[マップレット]** タブで実行します。

マップレットを使用するマッピングとマップレットは、マップレットが同期されるまでは無効です。変更されたマップレットを含むマッピングタスクを実行すると、タスクは失敗します。

マップレットを同期すると、更新によって、マッピングまたはマップレット内の他のトランスフォーメーションで検証エラーが発生する可能性があります。

PowerCenter または SAP からデータ統合にインポートしたマップレットを同期することはできません。

マップレットを同期するには、次の手順を実行します。

1. マップレットを使用するマッピングまたはマップレットを開きます。
2. マップレットトランスフォーメーションを選択します。
3. **【マップレット】** タブで、**【同期】** をクリックします。
4. トランスフォーメーションロジックでエラーを修正します。

## データ分類

組織がメタデータコマンドセンターを使用している場合は、メタデータコマンドセンターのカatalogソースのオブジェクトに関連付けられているデータ分類にマップレットをリンクできます。マッピングでソースオブジェクトを使用する際に、CLAIRE はオブジェクトカラムに関連付けられているデータ分類に基づいてマップレットを推奨します。

データ分類をマップレットにリンクする場合は、メタデータコマンドセンターでデータ分類を作成してから、それらのデータ分類をCatalogソースで設定して、データ分類スキャンジョブを実行します。データ統合でデータ分類をマップレットに関連付けて、データ分類にも関連付けられているソースを使用したときに、CLAIRE によってマップレットが推奨されるようにします。

データ統合で作成した有効なマップレットにデータ分類を追加できます。データ分類は、PowerCenter または SAP からインポートしたマップレットには追加できません。

### データ分類の追加

マップレットにデータ分類を追加して、一致するデータ分類を持つソースオブジェクトを追加したときに、CLAIRE によってマッピングでマップレットが推奨されるようにします。

1. **【エクスプローラ】** ページで、マップレットに移動します。
2. マップレットを含む行で、**【アクション】** > **【データ分類の編集】** をクリックします。
3. **【データ分類】** ウィンドウで、マップレットに関連付けるデータ分類をドラッグアンドドロップします。
4. **【保存】** をクリックします。

## マップレットの検証

式エディタおよび**【検証】** パネルの該当するタブでエラーを解決することで、マップレットを検証します。

Mapplet Designer の次の領域で検証エラーを解決します。

### 式エディタでの検証エラー

式エディタは、詳細モードと詳細モード以外で使用する式を検証します。式エディタには、データ統合サーバーと詳細クラスタでの処理の違いに応じて、異なる数の検証メッセージが表示されます。

式が有効な場合、または検証エラーが詳細モードと詳細モード以外の使用で同じである場合、式エディタには1つの検証メッセージが表示されます。式が詳細モードまたは詳細モード以外のいずれかでのみ有効な場合、または検証エラーが異なる場合、式エディタには複数の検証メッセージが表示されます。

#### 【検証】 パネルの検証エラー

【検証】 パネルにはマップレットのエラーが表示されます。詳細モード以外でマップレットを使用するには、**【マッピング】** タブでエラーを解決します。詳細モードでマップレットを使用するには、**【マッピング (詳細モード)】** タブでエラーを解決します。

1つのタブでマップレットが有効な場合、他のタブではマップレットが無効な場合があります。どちらのタブでもマップレットに検証エラーが含まれない場合は、そのマップレットを任意のマッピングで使用できます。

次の表に、マップレットの状態と、マップレットを使用できるマッピングを示します。

マップレットの状態	マップレットを使用できるマッピング
有効	すべてのマッピングで使用できます
不正	すべてのマッピングで使用できません
有効 (マッピングの場合)	詳細モード以外のマッピングで使用できます
有効 (詳細モードのマッピングの場合)	詳細モードのマッピングで使用できます

## PowerCenter マップレット

Informatica PowerCenter ユーザーの場合は、PowerCenter マップレットを使用してデータ統合にマップレットを作成できます。

PowerCenter マップレットを使用するには、PowerCenter でマップレットを作成して、このマップレットをXML ファイルにエクスポートします。次に、このXML ファイルをデータ統合にアップロードします。

マップレットには一連のトランスフォーメーションが含まれます。PowerCenter マップレットには、1つまたは複数のソーストランスフォーメーションを含めることができますが、ターゲットトランスフォーメーションを含めることはできません。以下のデータ統合タスクで PowerCenter マップレットを使用できます。

- 同期タスク1つの同期タスクで1つのマップレットを使用できます。
- マッピング1つのマッピングタスクで複数のマップレットを使用できます。Visio テンプレートまたはマッピングでマップレットを使用できます。詳細モードで PowerCenter マップレットを使用することはできません。
- マスキングタスクマスキングタスクでパッシブなマップレットを使用すると、ターゲットフィールドをマスクできます。

## アクティブおよびパッシブな PowerCenter マップレット

マップレットはアクティブかパッシブになります。

データ統合で PowerCenter マップレットをアップロードする場合は、マップレットがアクティブまたはパッシブのどちらかを指定します。



アクティブなマップレットには、少なくとも1つのアクティブな PowerCenter トランスフォーメーションが含まれます。アクティブなマップレットによって返される行の数は、マップレットに渡されるソース行の数と異なることがあります。たとえば、アクティブなマップレットでは5つのソース行が集計されて、1つの行が返されることがあります。

パッシブなマップレットには、パッシブな PowerCenter トランスフォーメーションのみが含まれます。パッシブなマップレットで返される行の数は、ソースから渡される行の数と同じです。

## マップレット内のストアードプロシージャ

使用する PowerCenter マップレットにストアードプロシージャが含まれる場合、このストアードプロシージャには例外処理が含まれている必要があります。

例外処理には必要に応じて複雑な処理を記述できます。または、次の簡単な例を使用できます。

```
Exception
when NO_DATA_FOUND
then NULL;
END;
```

例えば、PowerCenter ワークフローに次のようなストアードプロシージャがあるとします。

```
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER)
AS
BEGIN
SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
```

このワークフローをエクスポートするには、次のように例外処理を追加します。

```
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER)
AS
BEGIN
SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
Exception
when NO_DATA_FOUND
then NULL;
END;
```

## マップレットの PowerCenter XML ファイル

データ統合でマップレットを使用するには、PowerCenter マップレットを定義する PowerCenter XML ファイルをアップロードします。

PowerCenter XML ファイルを使用してデータ統合マップレットを作成する場合は、次のルールを検討します。

- マップレットに接続を使用するトランスフォーメーションが含まれる場合、PowerCenter XML ファイルに含まれるのは、ワークフローが1つ、セッションタスクが1つ、マッピングが1つ、マップレットが1つのみです。  
マップレットに接続を使用するトランスフォーメーションが含まれない場合、PowerCenter XML ファイルにマップレットを1つ含める必要があります。ワークフロー、セッションタスク、およびマッピングは省略可能です。
- セッションでは、あらゆるタイプの接続を使用できます。
- PowerCenter マッピングでは、すべてのソースフィールドとターゲットフィールドをマッピングする必要はありません。
- PowerCenter マップレットには、次のサポート対象トランスフォーメーションを含めることが出来ます。
  - アグリゲータトランスフォーメーション
  - 式トランスフォーメーション

- フィルタトランスフォーメーション
- HTTP トランスフォーメーション
- ルックアップトランスフォーメーション
- Salesforce ルックアップトランスフォーメーション (複数の一致で単一の一致を返す)
- Salesforce Picklist トランスフォーメーション
- Salesforce 統合トランスフォーメーション
- ソータートランスフォーメーション
- ストアドプロシージャトランスフォーメーション (例外処理あり)
- トランザクション制御トランスフォーメーション
- Web サービスコンシューマトランスフォーメーション
- ソースがフラットファイルかデータベースの XML ジェネレータートランスフォーメーション
- ソースがフラットファイルかデータベースの XML パーサートランスフォーメーション
- 同期タスクでマップレットを使用する場合、PowerCenter マップレットに複数の入力トランスフォーメーションを含める事は出来ません。
- マップレットをマッピングタスクで使用する場合は、PowerCenter マップレットに複数の入力トランスフォーメーションを含めることが出来ます。
- データ統合では、複数の入力グループがある PowerCenter マップレットが、1つの入力グループがあるマップレットにフラット化されます。このため、PowerCenter マップレットの各入力グループのポートには一意の名前が必要です。名前が一意でない場合は、PowerCenter で入力ポートの名前を変更してから、マップレットを含む PowerCenter XML ファイルをエクスポートします。
- データ統合は再利用可能なオブジェクトを保存するためにリポジトリを使用しないので、PowerCenter マップレットにはショートカットなどの再利用可能なオブジェクトを含める事は出来ません。再利用可能なオブジェクトを含まないマップレットをエクスポートします。

## データ統合タスクでの PowerCenter マップレット

データ統合タスクで PowerCenter マップレットを使用する場合は、次のルールとガイドラインに従ってください。

- ソースとマップレットの間にはルックアップトランスフォーメーションを追加できます。
- ソースとマップレットの間およびマップレットとターゲットの間には、式トランスフォーメーションを追加できます。
- 同期タスクにマップレットを追加すると、同期タスクウィザードは既存のフィールドマッピングを削除しません。ソースとターゲットの間にパッシブマップレットを追加しても、同期タスクウィザードは既存のフィールドマッピングを削除しません。
- 保存済みクエリが含まれる同期タスクでアクティブなマップレットを使用すると、同期タスクは、タスクに設定されたターゲットオプションを無視して、データをターゲットに挿入します。
- データ統合は、PowerCenter セッションレベルの上書きをマッピングに保持します。

## PowerCenter マップレットの設定

データ統合でマップレットを作成するには、PowerCenter マップレットが含まれる PowerCenter XML ファイルをアップロードします。

PowerCenter XML ファイルをアップロードすると、その XML ファイルのマップレット定義に基づいて、データ統合がマップレットを作成します。

1. マップレットを作成するには、**[新規]** > **[マップレット]** > **[マップレット - PC インポート]** をクリックし、**[作成]** をクリックします。

マップレットを編集するには、**[エクスプローラ]** ページでマップレットに移動します。マップレットが含まれている行で、**[アクション]** をクリックし、**[編集]** を選択します。

2. 次の詳細を設定します。

明細	説明
マップレット名	マップレットの名前。 マップレット名には、英数字、スペース、および次の特殊文字を含めることができます。 + - 最大長は 100 文字です。マップレット名では大文字と小文字が区別されません。
場所	マップレットを格納するプロジェクトフォルダ。
説明	マップレットの説明。 最大長は 255 文字です。
マップレットタイプ	マップレットがアクティブまたはパッシブのどちらなのか。マップレットタイプは、マップレットの PowerCenter トランスフォーメーションのタイプによって異なります。 - アクティブ。このマップレットには、1 つ以上のアクティブな PowerCenter トランスフォーメーションが含まれます。 - パッシブ。このマップレットには、パッシブな PowerCenter トランスフォーメーションのみが含まれます。

3. PowerCenter XML ファイルをアップロードするには、**[アップロード]** をクリックします。
4. **[マップレット XML ファイルのアップロード]** ダイアログボックスで、**[ファイルを選択]** をクリックします。
5. 適切な場所を参照して、PowerCenter XML ファイルを選択します。
6. **[OK]** をクリックします。  
[マップレット XML ファイルの詳細] 領域には、接続、入力フィールド、出力フィールドが表示されます。
7. **[保存]** をクリックします。

## 第 12 章

# 保存済みクエリ

保存済みクエリは、データベースに対して SQL 文を実行するために作成するコンポーネントです。保存済みクエリは、同期タスクでソースオブジェクトとしてまたは SQL トランスフォーメーションでクエリとして使用出来ます。

保存済みクエリは次の場所で使用出来ます。

### 同期タスクでソースとして

単一オブジェクトまたは複数オブジェクトのソースオプションを使用して設定することができないデータベースソースを使用する場合、保存済みクエリを作成します。例えば、保存済みクエリを作成して、ソースフィルタを含めたり、複数テーブルの複雑な結合を実行したりすることもできます。クエリは SELECT 文である必要があります。

同期タスクで保存済みクエリを使用するには、最初に保存済みクエリコンポーネントを作成します。次に、同期タスクを構成する場合に、保存済みクエリをソースオブジェクトとして選択します。1 つの同期タスクに 1 つの保存済みクエリを追加出来ます。

### マッピング内の SQL トランスフォーメーションでクエリとして

保存済みクエリを作成し、SQL トランスフォーメーションでクエリとして使用出来ます。クエリには、1 つ以上の SQL 文を含める事が出来ます。SQL トランスフォーメーションで 1 つの保存済みクエリを使用出来ます。

SQL トランスフォーメーションで保存済みクエリを使用するには、最初に保存済みクエリコンポーネントを作成します。その後、マッピングで SQL トランスフォーメーションの設定時に使用する保存済みクエリを選択します。

保存済みクエリを使用するには、組織が適切なライセンスを所持している必要があります。詳細については、Informatica グローバルカスタマサポートにお問い合わせ下さい。

## 保存済みクエリ構文

保存済みクエリを作成する際には、クエリを再度実行するデータベースで有効な SQL 文を入力します。

保存済みクエリを使用する場所に基づいて異なる SQL 文を使用できます。

### 同期タスク

同期タスクでソースとして使用するために保存済みクエリを作成する場合、SQL 文は SELECT 文である必要があります。データ統合はこの SQL 文を使用してソースカラム情報を取得します。保存済みクエリを保存する前に、各カラムのデータ型、精度、スケールを編集できます。

例えば、次の SQL 文を使用して 2016 年以降のトランザクションが含まれる TRANSACTIONS テーブルに基づいて保存済みクエリを作成することもできます。

```
SELECT TRANSACTION_ID, TRANSACTION_TOTAL, TRANSACTION_TIMESTAMP from dbo.TRANSACTIONS WHERE  
TRANSACTION_TIMESTAMP>'0:0:0 01/01/2016'
```

データ統合では、保存済みクエリのカラム名が必ず一意になります。SQL 文で重複するカラム名が返されると、データ統合は次のように重複するカラム名に数値を追加します。

<column\_name><number>

## SQL トランスフォーメーション

SQL トランスフォーメーションで使用するために保存済みクエリを作成する場合、クエリで次の 1 つ以上の SQL 文を使用できます。

ALTER  
CALL  
COMMENT  
COMMIT  
CREATE  
DELETE  
DROP  
EXPLAIN PLAN  
GRANT  
INSERT  
LOCK TABLE  
MERGE  
RENAME  
REVOKE  
ROLLBACK  
SELECT  
TRUNCATE  
UPDATE

クエリを作成するときには、以下のガイドラインに従ってください。

- COUNT などの集計関数は Salesforce 接続でのみ使用できます。
- TO\_CHAR や TO\_DATE などの変換関数は使用しないでください。
- アスタリスク (\*) を使用してテーブルのすべてのカラムを選択しないでください。選択するカラムを一覧表示します。
- SQL トランスフォーメーションパラメータのバインドまたは?input\_field?や~input\_field~などの文字列置換表記はクエリで使用しないでください。保存済みクエリには、SQL トランスフォーメーション入力フィールドに関する情報はありません。

**ヒント:** 保存済みクエリを作成する前に、ソースデータベースで使用する SQL 文をテストします。データ統合は、無効な SQL ステートメントに固有のエラーメッセージを表示しません。

## 同期タスクでの保存済みクエリの使用

同期タスクで保存済みクエリを使用するには、次のルールおよびガイドラインに従ってください。

- 1つの保存済みクエリを各同期タスクに追加できます。
- 同期タスクで使用されている保存済みクエリは削除できません。
- 保存済みクエリのカラム情報を編集しても、同期タスクは、実行時にエラー行をエラー行ファイルに書き込みません。
- 保存済みクエリが含まれる同期タスクで有効なマップレットを使用すると、別のターゲットオプションを選択した場合、同期タスクはターゲットにデータを挿入します。

## SQL トランスフォーメーションでの保存済みクエリの使用

SQL トランスフォーメーションを設定して、**【プロパティ】** パネルの **【SQL】** タブで保存済みクエリを使用します。SQL トランスフォーメーションで1つの保存済みクエリを使用出来ます。

1. **【プロパティ】** パネルの **【SQL】** タブで、クエリを再度実行するテーブルを含む接続を選択します。
2. SQL タイプを **【SQL クエリ】** に設定します。
3. クエリタイプを **【保存済みクエリ】** に設定します。
4. SQL トランスフォーメーションで処理する保存済みクエリを選択します。

SQL トランスフォーメーションの詳細については、トランスフォーメーションに関するセクションを参照して下さい。

## 保存済みクエリの設定

同期タスクでソースオブジェクトとしてまたはSQL トランスフォーメーションでクエリとして使用出来る保存済みクエリを作成します。

1. **【新規】** > **【コンポーネント】** > **【保存済みクエリ】** をクリックし、**【作成】** をクリックします。

保存済みクエリを編集するには、**【エクスプローラ】** ページで保存済みクエリに移動します。保存済みクエリが含まれている行で、**【アクション】** をクリックし、**【編集】** を選択します。

2. 次の情報を入力します。

明細	説明
保存済みクエリの名前	保存済みクエリの名前。 クエリ名には、英数字、スペース、および次の特殊文字を含めることができます。_ . + - 最大長は 100 文字です。クエリ名では大文字と小文字は区別されません。
場所	保存済みクエリの場所。保存済みクエリを保存するフォルダを参照するか、デフォルトの場所を使用します。 [Explore (参照)] ページが現在アクティブになっていて、プロジェクトまたはフォルダが選択されている場合、アセットのデフォルトの場所はその選択されているプロジェクトまたはフォルダです。そうでない場合、デフォルトの場所は直近で保存されたアセットの場所です。
説明	保存済みクエリの説明。 最大長は 255 文字です。
データベースタイプ	ソースデータベースのタイプ。次のいずれかのデータベースタイプを選択します。 - Salesforce - Oracle - SQL Server - MySQL - ODBC - MS Access  SQL トランスフォーメーションでクエリを使用するには、データベースタイプが Oracle または SQL Server である必要があります。
SQL クエリ	SQL クエリを構成する SQL 文。 同期タスクでソースとしてクエリを使用するには、SELECT 文を入力します。データ統合はこの SELECT 文を使用してソースカラム情報を取得します。 SQL トランスフォーメーションでクエリを使用するには、1 つ以上の有効な SQL 文を入力します。保存済みクエリには SQL トランスフォーメーション入力フィールドに関する情報がないため、クエリでは SQL トランスフォーメーションパラメータバインディングまたは文字列置換表記を使用しないで下さい。

- SELECT 文を入力するには、**[カラムの取得]** をクリックして **[接続]** を選択します。  
[保存済みクエリのカラムの詳細] テーブルに、SQL 文で選択したカラムが表示されます。
- 必要に応じて、[保存済みクエリのカラムの詳細] テーブルでは、データ型、精度、スケールを編集します。  
これらの値を編集した場合、データ統合はエラー行をエラー行ファイルに書き込みません。
- [保存]** をクリックします。

## 第 13 章

# 共有シーケンス

共有シーケンスは再利用可能なシーケンスで、複数のシーケンスジェネレータトランスフォーメーションで使用できます。共有シーケンスを使用する際、シーケンスジェネレータトランスフォーメーションは、共有シーケンスのプロパティを使用して値を生成します。同じシーケンス内の数値を、複数のマッピングタスクのデータに割り当てる場合は、共有シーケンスを使用します。

複数のマッピングとマップレットで同じ共有シーケンスを使用できます。マッピングタスクを実行すると、データ統合がシーケンス内の一連の値を予約するため、各マッピングタスクでは一意の値が生成されます。

例えば、一意の ID をカスタマテーブルのエントリに割り当てるとします。同じターゲットテーブルに顧客データを読み込むマッピングが 2 つあり、それらのマッピングは同時刻に実行されるようにスケジュール設定されています。ターゲットで重複 ID が発生しないようにするため、共有シーケンスを使用します。

共有シーケンスを **【新しいアセット】** ページで作成します。非共有シーケンスはシーケンスジェネレータトランスフォーメーションで作成できます。非共有シーケンスの詳細については、「トランスフォーメーション」を参照してください。

共有シーケンスは、SQL ELT モードのマッピングでは使用できません。

## 共有シーケンスのプロパティ

共有シーケンスを作成する際は、シーケンス名などの全般プロパティや、シーケンスを定義するプロパティ（初期値、増分値、終了値など）を定義します。

次の表に、共有シーケンスに定義するプロパティを示します。

プロパティ	説明
名前	必須。共有シーケンスの名前。 英数字とアンダースコア ( _ ) を使用できます。最大長は 200 文字です。
場所	シーケンスが存在するプロジェクトまたはフォルダ。 <b>【エクスプローラ】</b> ページが現在アクティブになっていて、プロジェクトまたはフォルダが選択されている場合、アセットのデフォルトの場所はその選択されているプロジェクトまたはフォルダです。そうでない場合、デフォルトの場所は直近で保存されたアセットの場所です。
説明	オプション。共有シーケンスの説明。



プロパティ	説明
増分	生成されるシーケンスでの2つの連続する値の差。例えば、[増分] が2で既存の値が4の場合、シーケンスで生成される次の値は6になります。 デフォルトは1です。 最大は2,147,483,647です。
終了値	マッピングタスクが生成する最大値。 タスク実行中に、シーケンスが未処理の行を残したままこの値に到達し、シーケンスのサイクルが設定されていない場合、その実行は失敗します。サイクル動作を実行するようにシーケンスが設定されている場合、シーケンスはこの値を使用し、サイクル開始値から開始します。 最大値は9,223,372,036,854,775,807です。
初期値	マッピングタスクで、シーケンスを初めて使用するときシーケンスで初期値として使用する値が使用されます。 一連の値間でサイクルする場合、この値は、[サイクル開始値] 以上、[終了値] 未満である必要があります。 共有シーケンスをリセットすると、シーケンスはこの値にリセットされます。 デフォルトは0です。
サイクル	有効にすると、マッピングタスクはシーケンスの範囲でサイクル動作を実行します。無効にすると、タスクは設定されている終了値でシーケンスを停止します。タスクが終了値に到達した時点で処理する必要がある行が残っていると、セッションが失敗します。 デフォルトでは無効になっています。
サイクル開始値	サイクル動作を実行するようにシーケンスが設定されている場合にマッピングタスクで使用する、生成されたシーケンスの開始値。シーケンスは、終了値に達すると、この値に戻ります。 デフォルトは0です。
予約値の数	マッピングタスクが一度にキャッシュするシーケンス値の数。 1000以上を指定する必要があります。
現在の値	シーケンスでの現在の開始値を表示します。

## 予約値の数

予約値の数により、マッピングタスクが、各タスクの実行中に1回でキャッシュするシーケンスの値の数が決まります。

複数のシーケンスジェネレータートランスフォーメーションが同じ共有シーケンスを使用すると、共有シーケンスのインスタンスが同時に複数できる可能性があります。重複する値が生成されないようにするため、予約値の数を設定することで、各マッピングについて一定の範囲のシーケンス値を予約します。

例えば、予約値の数 = 1000、初期値 = 1、増分 = 1 という共有シーケンスを設定します。2つのマッピングが共有シーケンスを使用しています。最初のマッピングタスクが実行されると、データ統合は1000個の値（1～1000）をマッピングタスクのために予約し、リポジトリ内の現在値を1001に更新します。2番目のマッピングタスクが実行されると、データ統合は次の1000個の値（1001～2000）を予約し、現在値を2001に更新します。どちらかのマッピングタスクが予約値をすべて使用すると、データ統合は別の一連の値を予約し、現在値を更新します。

データ統合がシーケンスで予約する値の数は、予約値の数に増分値を乗算することで決定されます。

例えば、2で始まる共有シーケンスがあり、増分が2で、予約値が1000の場合、データ統合はシーケンスで1000個の値を予約するか、数値の2~2002を予約し、現在値を2003に更新します。

デフォルトでは、予約値の数は1000です。パフォーマンスを向上するには、予約値の数を増やします。これにより、データ統合がリポジトリに対して行う呼び出しの数が減少します。予約値の数を1000より少なくすることはできません。

タスク実行のために予約されたものの、タスクでは使用されない値は、タスクの完了時に失われます。例えば、1250行あるテーブルでIDを生成するとします。共有シーケンスを設定して増分1の値を生成し、予約値の数を2000に設定します。タスクが実行されると、データ統合は、タスクのために2000個の値を予約し、現在値を2001に更新します。次のタスクが実行される際、シーケンスは2001から始まり、1251~2000の値は失われます。

**注:**パーティションを使用する場合、データ統合は、各パーティション用に一連の値を予約します。

## 共有シーケンスの作成

**[新しいアセット]** ページで共有シーケンスを作成します。

1. **[新規]** > **[コンポーネント]** > **[共有シーケンス]** をクリックし、**[作成]** をクリックします。
2. シーケンスのプロパティを定義します。
3. シーケンスを保存します。

## 共有シーケンスの使用

共有シーケンスは、マッピングまたはマップレットのシーケンスジェネレータートランスフォーメーションで使用できます。

1. マッピングまたはマップレットを開きます。
2. シーケンスジェネレータートランスフォーメーションをキャンバスに追加して、ダウンストリームトランスフォーメーションに接続します。
3. **[シーケンス]** タブで **[共有シーケンスの使用]** を選択します。
4. 使用するシーケンスを選択します。
5. シーケンスジェネレータの詳細プロパティと出力フィールドを設定します。  
シーケンスジェネレータートランスフォーメーションの設定の詳細については、「トランスフォーメーション」を参照してください。

**注:** **[共有シーケンスの使用]** を選択しない場合、シーケンスジェネレータートランスフォーメーションは非共有シーケンスを生成します。

# 共有シーケンスのリセット

現在のシーケンス値を、定義した初期値に更新するには、共有シーケンスをリセットします。

1. 共有シーケンスを開きます。
2. **【編集】** をクリックします。
3. **【シーケンスプロパティ】** 領域の **【現在の値のリセット】** をクリックします。
4. 共有シーケンスを保存します。

## 第 14 章

# ユーザー定義関数

ユーザー定義関数は再利用可能な関数で、式で使用できます。ユーザー定義関数を作成すると、Informatica Intelligent Cloud Services のトランスフォーメーション言語を使用して複雑な式を作成できます。ユーザー定義関数は、トランスフォーメーションやフィールド式と同じ構文を使用します。また、同じトランスフォーメーション言語コンポーネントを使用することもできます。

ユーザー定義関数は、マッピングまたはマップレットのトランスフォーメーション式、マッピングタスクのフィールド式、または他のユーザー定義関数に含めることができます。式でユーザー定義関数を使用するには、式エディタでユーザー定義関数を選択し、必要な引数を入力します。

同期タスクの式でユーザー定義関数を使用することはできません。

詳細モードでは、次のトランスフォーメーションにユーザー定義関数を含めることができます。

- アグリゲータ
- 式

ユーザー定義関数は、SQL ELT モードのマッピングの式では使用できません。

### 例

名前や住所などのテキスト文字列から先頭または末尾のスペースを削除するとします。この場合、テキスト文字列を引数とした RemoveSpaces というユーザー定義関数を作成し、LTRIM 関数と RTRIM 関数を実行します。このユーザー定義関数を設定する場合は、以下の式を入力します。

```
LTrim( RTrim(TextString) )
```

作成した関数を式トランスフォーメーションで使用して、受信フィールド LAST\_NAME から先頭および末尾のスペースを削除します。式フィールドに含まれる式は次のとおりです。

```
:UDF.RemoveSpaces(LAST_NAME)
```

式では、ユーザー定義関数名の先頭に「:UDF」が付きます。受信フィールド LAST\_NAME が引数として関数に渡されます。

## ユーザー定義関数の作成

**【新しいアセット】** ページでユーザー定義関数を作成します。作成したユーザー定義関数は、組織内のすべてのユーザーがそれぞれの権限に基づいて使用できます。

1. **【新規】** > **【コンポーネント】** > **【ユーザー定義関数】** をクリックし、**【作成】** をクリックします。
2. 関数名、場所、戻り型などの全般プロパティを設定します。
3. 必要に応じて、関数の引数を作成します。

引数を作成する際は、各引数の名前、データ型、および説明を設定します。

- 関数式を作成します。
- 関数を検証して保存します。

## ユーザー定義関数の全般プロパティ

ユーザー定義関数を作成する際は、関数名や戻り型などの全般プロパティを定義する必要があります。全般プロパティは、**[全般]** タブで定義します。

以下の表に、プロパティを示します。

プロパティ	説明
名前	関数の名前。組織内で一意である必要があります。 先頭は文字にする必要があります。文字、数字、および次の特殊文字を使用できます。 _ @ \$ # 名前 100 文字を超えない範囲で指定する必要があります。また、空白文字は使用できません。
場所	ユーザー定義関数の場所。ユーザー定義関数を保存するフォルダを参照するか、デフォルトの場所を使用します。 デフォルトの場所は直近で保存されたアセットの場所です。
説明	ユーザー定義関数の説明。 説明は、トランスフォーメーション式、フィールド式、または他のユーザー定義関数で関数を選択したときに式エディタに表示されます。
戻り型	関数から返される値のデータ型。バイナリ、日付、数値、または文字列のいずれかです。

## ユーザー定義関数の引数

ユーザー定義関数を作成する際は、関数に渡す値を定義する引数を含めることができます。引数は **[引数]** タブで作成します。最大 10 個の引数を作成できます。

引数を作成するには、**[追加]** をクリックして次の情報を入力します。

プロパティ	説明
名前	引数の名前。関数内で一意であることが必要です。 名前は英字で始める必要があり、英字、数字、およびアンダースコア ( _ ) を含めることができます。100 文字を超えない範囲で指定する必要があります。また、空白文字は使用できません。
タイプ	引数のデータ型。バイナリ、日付、数値、または文字列のいずれかです。
説明	引数の説明。 説明は、引数を選択するときに、 <b>[式]</b> タブの引数と関数の領域に表示されます。トランスフォーメーションまたはフィールド式で関数を使用するときに、式エディタにも表示されます。

引数は、**[引数]** タブに表示される順で関数に渡されます。順序を変更するには、変更する引数を選択して **[上に移動]** または **[下に移動]** をクリックします。

引数を削除するには、引数を含む行で **[削除]** をクリックします。

## ユーザー定義関数の式

関数を定義する式は【式】タブで作成します。関数式には、定数、演算子、組み込み関数、その他のユーザー定義関数を含めることができます。関数内に関数をネストすることで複雑な式を作成できます。

式を作成するには、式エディタで式を入力します。作成する式は、トランスフォーメーションまたはフィールド式と同じガイドラインに従い、同じ構文を使用する必要があります。式の作成の詳細については、「タスク」を参照してください。

式には、引数、組み込み関数、他のユーザー定義関数を追加できます。これらは、式エディタの引数と関数の領域で選択して【追加】をクリックすることで行えます。または、**Ctrl+スペースキー**を押すと、推奨される引数と関数のリストがインラインで表示されます。式を手動で入力することも可能です。

式に循環参照を含めることはできません。例えば、ユーザー定義関数 Function\_A の式に、Function\_A を含めることはできません。また、ユーザー定義関数 Function\_A がユーザー定義関数 Function\_B を呼び出す場合、Function\_B の式から Function\_A を呼び出すことはできません。

式を検証するには、【検証】をクリックします。データ統合で式が検証されます。その式を使用するユーザー定義関数と式は検証されません。

## ユーザー定義関数の編集および削除

ユーザー定義関数を編集または削除する際は、特定の制限が適用されます。

ユーザー定義関数を編集すると、それを使用する式や関数に影響が及ぶ可能性があります。ユーザー定義関数を編集する際には、以下のガイドラインを検討します。

- ユーザー定義関数を編集し、更新された関数が有効な場合、データ統合は、その関数を使用するすべての式とユーザー定義関数に変更内容をプロパゲートします。

これにより、ユーザー定義関数を使用する式や関数が無効になることがあります。例えば、関数を編集し、引数の数を1つから2つに変更して、関数を検証するとします。更新した関数は有効でも、関数に渡される引数が1つだと、そのユーザー定義関数を使用する式や関数は無効になります。

- ユーザー定義関数を編集して保存し、更新された関数が有効でない場合、データ統合は、その関数を使用する式やユーザー定義関数には変更内容をプロパゲートしません。
- ユーザー定義関数を保存した後にその名前を変更することはできません。

ユーザー定義関数を削除するには、まず、それを使用するすべての式や関数からそのユーザー定義関数を削除する必要があります。式や他のユーザー定義関数で使用されているユーザー定義関数を削除することはできません。

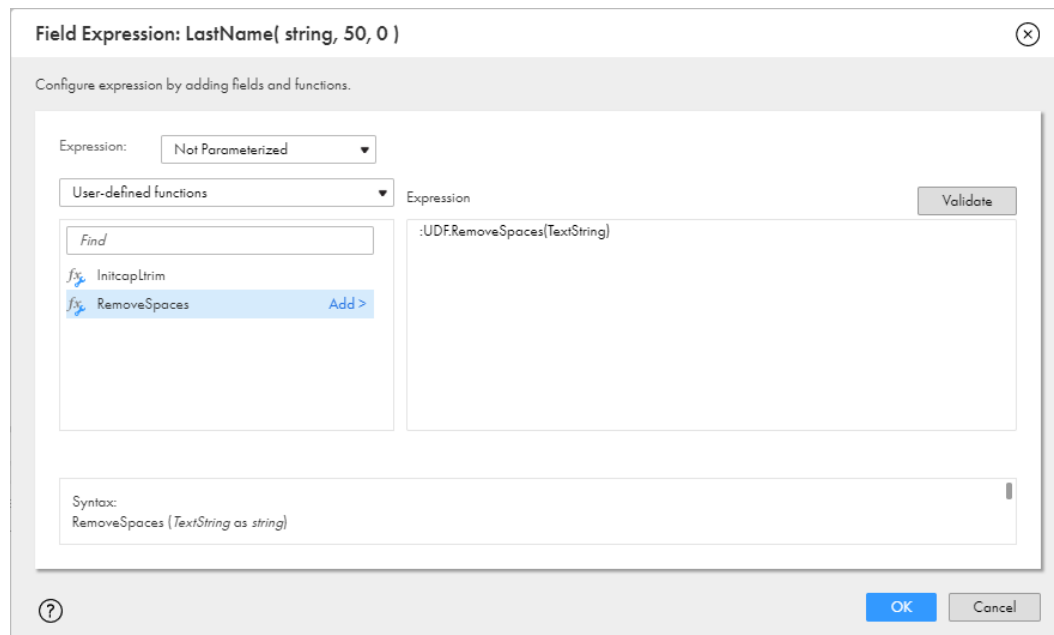
**ヒント:** ユーザー定義関数の依存関係を表示して、関数を使用するアセットを見つけます。

## ユーザー定義関数を使用した式の作成

ユーザー定義関数は、トランスフォーメーションまたはフィールド式に追加できます。

式を作成する際は、有効なユーザー定義関数が式エディタに表示されます。式を手動で入力する場合は、各ユーザー定義関数の先頭に:UDF を付けます。

次の図は、式トランスフォーメーションの式エディタでユーザー定義関数を選択した状態を示しています。



ユーザー定義関数を選択すると、式エディタには関数構文が以下の形式で表示されます。

<関数名> (<引数 1> as <データ型>, <引数 N> as <データ型>)

以下に例を示します。

RemoveSpaces(TextString as string)

式に関数を追加する際、以下のように接頭語として「:UDF」が挿入されます。

:UDF.RemoveSpaces(TextString)

式に関数を追加したら、引数をフィールド名または入出力パラメータに置換します。引数にはテーブル名を含めないでください。次の例に示すように、フィールド名のみを使用します。

:UDF.RemoveSpaces(NAME)

式の作成の詳細については、「タスク」を参照してください。

式の検証時、データ統合はユーザー定義関数を検証しません。式のみを検査します。

## ユーザー定義関数のパラメータ化

式変数として設定された入出力パラメータを使用して、式内のユーザー定義関数の全部または一部をパラメータ化できます。

ユーザー定義関数をパラメータ化するには、文字列の入出力パラメータを作成し、**[式の変数です]** を有効にします。ユーザー定義関数全体をパラメータ化するには、ユーザー定義関数全体の代わりにパラメータを使用します。ユーザー定義関数の一部をパラメータ化するには、ユーザー定義関数の引数の代わりに入出力パラメータを使用します。実行時に、マッピングタスクまたはパラメータファイル内のパラメータを解決します。

入出力パラメータの詳細については、「マッピング」を参照してください。

### すべてパラメータ化

式でユーザー定義関数をパラメータ化する場合は、**[すべてパラメータ化]** を使用します。ユーザー定義関数の代わりに入出力パラメータを使用する場合、パラメータ名では大文字と小文字が区別されます。パラメータを解決するときは、:UDF プレフィックスを含めます。

例えば、入出力パラメータ`$$UDFparameter`があり、それを使用してユーザー定義関数をすべてパラメータ化するとします。パラメータファイルでは、次の関数を使用してパラメータを解決します。

```
$$UDFparameter=:UDF.RemoveSpaces(names)
```

### 部分的にパラメータ化

引数のフィールド名をパラメータ化する場合、またはネストになったユーザー定義関数をパラメータ化する場合は、[部分的にパラメータ化]を使用します。ネストになったユーザー定義関数をパラメータ化する場合、関数の引数のパラメータは大文字でなければなりません。ユーザー定義関数を1つネストすることができます。

たとえば、入出力パラメータ`$$UDF`を作成し、式変数として設定します。式では、次の関数を設定します。

```
:UDF.RemoveSpaces($$UDF)
```

パラメータファイルでは、次の関数を使用してパラメータを解決します。

```
$$UDF=:UDF.Replace_Chars_special(Name)
```

### すべてまたは部分的にパラメータ化

同じ関数内で、[すべてパラメーター化]と[部分的にパラメーター化]の両方を使用できます。ユーザー定義関数と関数の引数をパラメータ化する場合は、[すべてパラメータ化]と[部分的にパラメータ化]の両方を使用します。

例えば、入出力パラメータ`$$UDF`があり、それを式で使用して、ユーザー定義関数をすべてパラメータ化するとします。パラメータファイルでは、次の値を使用してパラメータを解決します。

```
$$UDF=:UDF.Replace_Chars_special($$Field)  
$$Field=Name
```

すべてパラメータ化したユーザー定義関数で、`CurrentTaskName`、`CurrentRunId`、または`SESSSTARTTIME`システム変数を引数として渡す場合は、別の入出力パラメータを使用してシステム変数を渡す必要があります。

例えば、パラメータ化したユーザー定義関数で`CurrentTaskName`を使用するとします。入出力パラメータ`$$UDF`を使用して、関数をすべてパラメータ化します。パラメータファイルでは、次の値を使用してパラメータを解決します。

```
$$UDF=:UDF.udf_InitcapLtrim($$tasknameparam)  
$$tasknameparam='$CurrentTaskName'
```

## データ分類

組織がメタデータコマンドセンターを使用している場合は、メタデータコマンドセンターのカタログソースのオブジェクトに関連付けられているデータ分類にユーザー定義関数をリンクできます。マッピングでソースオブジェクトを使用する際に、CLAIREはオブジェクトカラムに関連付けられているデータ分類に基づいてユーザー定義関数を推奨します。

データ分類をユーザー定義関数にリンクする場合は、メタメタデータコマンドセンターでデータ分類を作成してから、それらのデータ分類をカタログソースに関連付けて、データ分類スキャンジョブを実行します。データ統合でデータ分類をユーザー定義関数に関連付けて、データ分類にも関連付けられているソースオブジェクトを使用したときに、CLAIREによって関数が推奨されるようになります。

## データ分類の追加

ユーザー定義関数にデータ分類を追加して、一致するデータ分類を持つソースオブジェクトを追加したときに、CLAIREによってマッピングで関数が推奨されるようになります。

1. **【エクスプローラ】** ページで、ユーザー定義の関数に移動します。



2. 関数を含む行で、**【アクション】** > **【データ分類の編集】** をクリックします。
3. **【データ分類】** ウィンドウで、ユーザー定義の関数に関連付けるデータ分類をドラッグアンドドロップします。
4. **【保存】** をクリックします。

# 索引

## B

B2B Gateway  
インテリジェント構造モデル [52](#)

## C

Cloud アプリケーション統合コミュニティ  
URL [7](#)  
Cloud 開発者コミュニティ  
URL [7](#)

## D

Data Engineering  
インテリジェント構造モデル [52](#)

## I

Informatica Intelligent Cloud Services  
Web サイト [7](#)  
Informatica グローバルカスタマサポート  
連絡先情報 [8](#)  
intelligent structure model:インテリジェント構造モデル  
詳細モード [51](#)

## P

PowerCenter XML ファイル  
マップレットのルール [89](#)  
PowerCenter マップレット [88](#)

## S

SQL トランスフォーメーション  
保存済みクエリ [92](#)

## W

Web サイト [7](#)

## あ

アクティブなマップレット  
説明 [81](#), [88](#)  
アセットコンポーネント [9](#)  
アップグレード通知 [8](#)

## い

インテリジェント構造  
アクション [76](#)  
データドリフト  
インテリジェント構造 [58](#)  
ドキュメント識別子 [75](#)  
ノード [76](#)  
プライマリキーおよび外部キー [57](#)  
モデル入力  
インテリジェント構造 [53](#)  
改良 [66](#)  
外部キー  
説明 [57](#)  
繰り返しグループ [55](#), [74](#)  
表示 [77](#)  
編集 [70](#), [72](#), [75](#)  
編集 [77](#)  
例 [61](#)  
インテリジェント構造ノード  
編集 [72](#)  
インテリジェント構造モデル  
B2B Gateway [52](#)  
Data Engineering [52](#)  
エクスポート [61](#)  
サフィックス [75](#)  
シナリオ [62](#)  
開発 [53](#)  
階層ビルダトランスフォーメーション [51](#)  
検出プロセス [53](#)  
構造パーサトランスフォーメーション [51](#)  
作成 [59](#)  
接頭語 [75](#)  
編集 [75](#)

## こ

コンポーネント  
インテリジェント構造モデル [50](#)  
タスクフローのファイルリスナ [17](#)  
ビジネスサービス [13](#)  
ファイルリスナ [15](#), [18](#)  
マップレット [81](#)  
階層スキーマ [40](#)  
共有シーケンス [96](#)  
業種別データサービスカスタマイザ [42](#)  
固定長ファイル形式 [29](#)  
保存済みクエリ [92](#)

## し

システムステータス [8](#)

## す

ステータス  
Informatica Intelligent Cloud Services [8](#)  
ストアドプロシージャ  
マッピングでの例外処理 [89](#)

## そ

ソース  
カスタム保存済みクエリ [92](#)

## た

タスクフローのファイルリテナ [17](#)

## は

パートナーフローのファイルリテナ [18](#)  
バッチなマッピング  
説明 [81](#), [88](#)  
パラメータ  
マッピング [83](#)

## ひ

ビジネスサービス  
定義 [13](#)

## ふ

ファイルリテナ  
タスクフローの呼び出し [17](#)  
タスクフロー実行の調整 [17](#)  
設定 [19](#)  
プライマリキー  
説明 [57](#)  
フラットファイル  
固定長ファイル形式 [29](#)

プロパティ  
共有シーケンス [96](#)

## ま

マッピング  
PowerCenter [88](#)  
PowerCenter XML ファイルのルール [89](#)  
アクティブとバッチ [81](#), [88](#)  
タスクで使用する場合のルール [90](#)  
概要 [81](#)  
出力 [82](#)  
設定 [91](#)  
入力 [82](#)  
マッピングの出力 [82](#)  
マッピングの入力 [82](#)  
マッピングパラメータ [83](#)

## め

メンテナンスの停止 [8](#)

## ゆ

ユーザー定義関数  
引数 [101](#)  
概要 [100](#)  
作成 [100](#)  
削除 [102](#)  
式 [102](#)  
全般プロパティ [101](#)  
編集 [102](#)  
戻り型 [101](#)

## り

リレーショナル出力  
操作 [70](#)