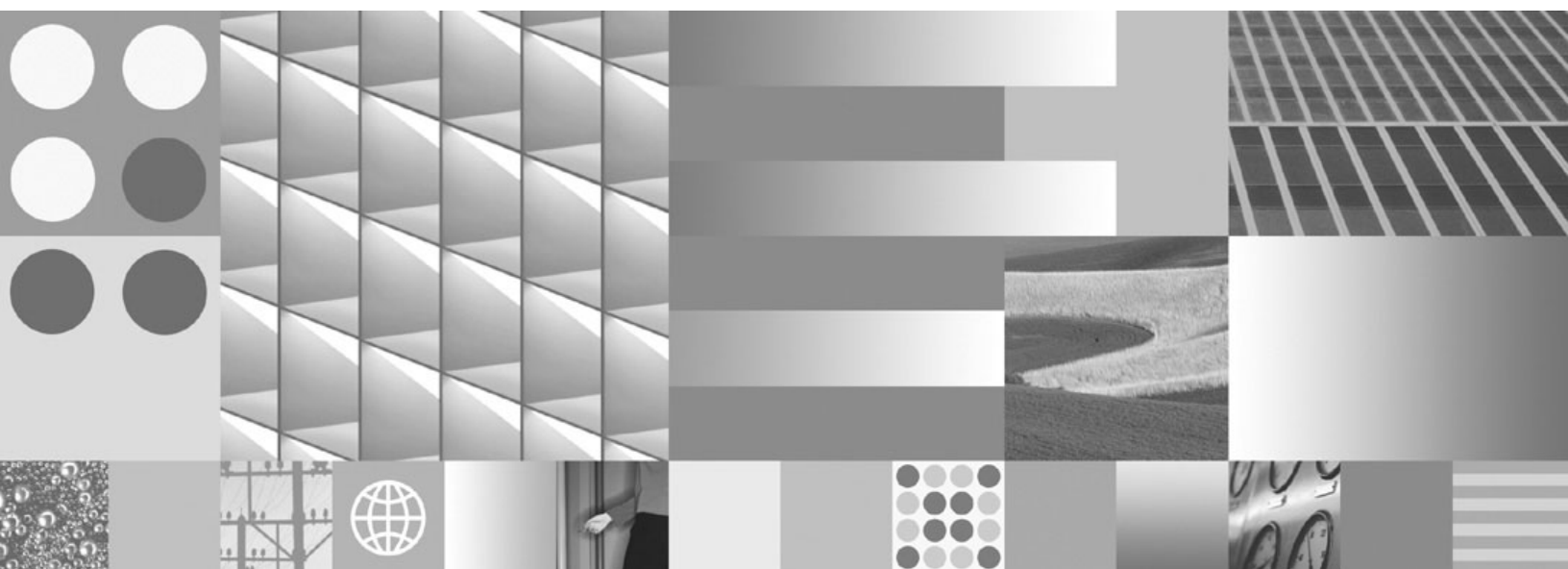
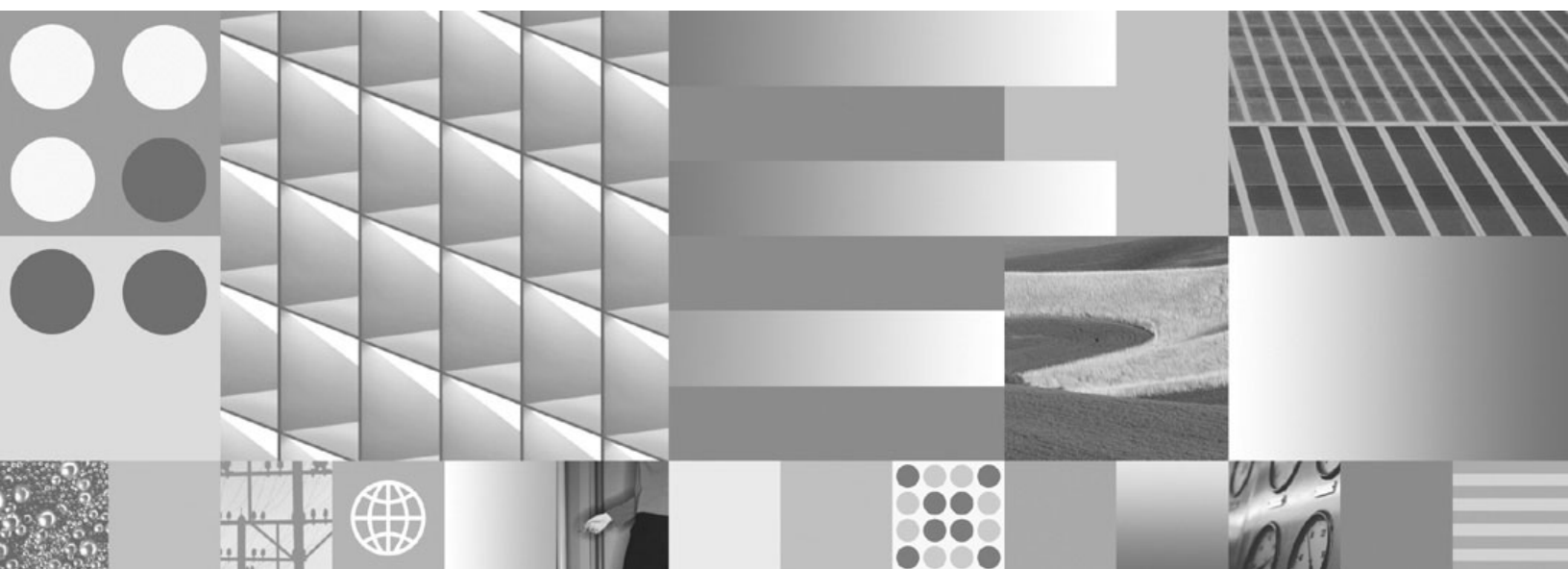


バージョン 2.0 リリース 1



IBM Optim
デザイナー ユーザー・ガイド

バージョン 2.0 リリース 1



IBM Optim
デザイナー ユーザー・ガイド

— ご注意 —

本書および本書で紹介する製品をご使用になる前に、75 ページの『特記事項』に記載されている情報をお読みください。

本書は、Optim デザイナー バージョン 2、リリース 1、モディフィケーション 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： Version 2.0 Release1
IBM Optim
Designer User Guide

発行： 日本アイ・ピー・エム株式会社

担当： トランスレーション・サービス・センター

第1版第1刷 2009.10

© Copyright International Business Machines Corporation 1996, 2009.

目次

第 1 章 Optim デザイナーの概要 1

Optim デザイナーの新機能	2
始めに	2
データ設計プロジェクトの作成	2
リレーショナル・データベース内のデータのマスキ ング	3
Optim パースペクティブ	4
サンプル・データ	6
データベース・サポート	7
アクセシビリティ機能	8
作業時のヘルプ	8

第 2 章 データ・ソース接続の定義 9

JDBC 接続の詳細	9
ネイティブ・データ・ソース接続での作業	11
ネイティブ・データ・ソース接続の自動的な定義	11
ネイティブ・データ・ソース接続の手動での定義	11

第 3 章 データ・モデルの管理 13

リバース・エンジニアリングに基づく物理データ・ モデル	13
リバース・エンジニアリングに基づく物理デー タ・モデルの定義	13
Database Relationship Analyzer 物理データ・モデ ルの使用	14
Optim プロパティがある新規論理データ・モデル の作成	15
データ・アクセス計画の作成	16
データ・アクセス計画の追加	16
データ・アクセス計画のポリシーの編集	16
選択ポリシーでの作業	16

第 4 章 データ管理サービスの設計 21

データ管理サービスの作成	21
サブセット・サービスの作成	21
データ変換サービスの作成	22
データ管理サービスの編集	23
サービス計画の編集	23
データ管理サービスのテスト	27
Optim Executor の構成	27
ローカル Optim 管理データ・ストアでの作業	27
データ管理サービスの実行	28
Optim リポジトリでの作業	29
デフォルトの Optim リポジトリの場所の入力	29
サービス要求の公開	29

第 5 章 Optim 相互運用性モデルでの作 業 31

アーカイブ要求モデル	31
アーカイブ要求の作成	31

削除要求モデル	31
削除要求の作成	32
抽出要求モデル	32
抽出要求の作成	32
挿入要求モデル	32
挿入要求の作成	33
ロード要求モデル	33
ロード要求の作成	33
復元要求モデル	34
復元要求の作成	34
Optim for z/OS 要求モデルでの作業	34
z/OS バッチ・ホストの構成	35
Optim for z/OS 要求の実行	36
Linux、UNIX、および Windows での Optim 要求モ デルの作業	36
pr0cmnd の場所の定義	37
Optim ディレクトリでの作業	37
Optim 定義のエクスポート	38
Optim 定義のインポート	39
Optim 要求の実行	40

第 6 章 データ・プライバシー・ポリシー の使用 43

日付マスク・ポリシー	43
範囲内のランダム日付	43
日付を月に丸める	44
日付を年に丸める	45
身元マスク・ポリシー	46
E メール・アドレス・ポリシー	46
クレジット・カードのポリシー	48
国民 ID のポリシー	50
ルックアップ・マスク・ポリシー	56
住所情報のルックアップ・ポリシー	57
個人情報のルックアップ・ポリシー	57
ファーストネーム情報のルックアップ・ポリシー	58
ラストネーム情報のルックアップ・ポリシー	58
会社名ルックアップ・マスク・ポリシー	59
ルックアップ・ポリシーの作成	59
数値マスク・ポリシー	60
ガウス・ランダム・ダブル	60
ガウス・ランダム整数	61
範囲内一様ランダム・ダブル	61
範囲内一様ランダム・ロング	62
スクランブル・マスク・ポリシー	63
反復可能な置換	63
正規表現による反復可能な置換	65
文字の置換	66
正規表現による文字の置換	67
文字のスクランブル	68
正規表現による文字のスクランブル	69
単純な文字のスクランブル	69

JavaScript ポリシー	71
JavaScript ポリシーの作成	72
JavaScript ポリシーへの JavaScript ファイルの追 加	72

JavaScript ポリシーの JavaScript ファイルの編集	73
JavaScript 式の例	73
索引	79

第 1 章 Optim デザイナーの概要

Optim™ デザイナーでは、データ・モデル、データ・プライバシー・ポリシー、および Optim 相互運用性要求モデルの定義ができます。Optim デザイナーを使用して、Optim および Optim z/OS® 要求を実行することができます。Optim デザイナーを使用して、データ管理サービスの作成とテストもできます。

論理データ・モデルおよび物理データ・モデル

Optim 相互運用性モデルを定義する、またはデータ管理サービス要求を処理するには、論理データ・モデルを使用してソースまたはターゲットのデータを定義する必要があります。物理データ・モデルを変換することにより、新規の論理モデルを作成できます。

Optim 相互運用性モデル

Optim 相互運用性モデルでは、Optim および Optim for z/OS 要求の編集と処理ができます。要求では Optim デザイナーで作成したデータ・モデルが使用されます。要求には、データ・プライバシー・ポリシーを含めることができます。Optim ディレクトリーに接続して、Optim 定義のインポートとエクスポートが行えます。

データ・プライバシー・ポリシー

データ・プライバシー・ポリシーを使用することにより、Optim 相互運用性モデルまたはデータ管理サービスのデータをマスクすることができます。プライバシー・ポリシーを使ったデータのマスクには、ルックアップ、ルール・ベース、および JavaScript™ の 3 つのオプションがあります。ルックアップ・オプションでは、ルックアップ表を使用して、マスクされたデータを作成します。ルール・ベース・オプションでは、マスクされるデータを関数を使用して生成します。JavaScript オプションは、JavaScript 式を使用してデータ変換を定義し、データ管理サービスでの使用のみ可能です。

以下を行うことができます。

- ルックアップ関数を使用して、選択されたソース・エンティティの値を、対応するルックアップ表の列の値に置き換えます。
- ルール・ベース関数を使用して、国民 ID 番号、クレジット・カード番号、および E メール・アドレスを、有効な固有値でマスクします。
- ルール・ベース関数を使用して、日付、文字、および番号の値を生成します。
- 「切り替え」値に基づいて、ルックアップ関数またはルール・ベース関数を適用します。
- JavaScript を使用して、データ管理サービスにカスタム変換を定義します。

データ・アクセス計画

データ・アクセス計画には、Optim 相互運用性モデルまたはデータ管理サービスのソース論理データ・モデルで、どのデータを処理または変換するかを決定するためのポリシーが含まれます。データ・アクセス計画を使用することで、論理データ・モデルのための選択ポリシーおよびデータ・プライバシー・ポリシーを定義できます。選択ポリシーによって、相互運用性モデルまたはデータ管理サービスで使用されるエンティティおよび属性が決まります。

データ管理サービス

Optim データ管理サービスでは、データの変換、およびスキーマ間のデータのコピーが可能です。サービスが処理するエンティティにデータ・プライバシー・ポリシーを適用することにより、データをマスクすることができます。

Optim デザイナーの新機能

Eclipse プラットフォームを基盤とする Optim デザイナーを使用することで、データ・モデルとデータ管理サービスを作成できます。

定義した Optim サービスは、中央リポジトリに公開されます。 Optim 管理コンソールを使用して、 Optim 管理サーバーと Optim プロキシにサービスを割り当てます。 Optim デザイナーを Optim Executor のインストール環境で使用することで、 Optim デザイナーのテスト・モードを使用してサービスをローカルに実行することも可能です。 Derby データベースが提供され、これにはサンプル・データと事前定義のソース/ターゲット・データ・ソースが含まれています。

Optim デザイナーの機能は、以下のとおりです。

- リレーショナル・データ・ソースのサポート。
- リレーショナル・スキーマからモデルを発見して作成するウィザード・ベースの機能。
- データ・プライバシー・フィーチャー。以下が含まれます。
 - グローバル・データ・マスキング・ポリシーを作成するためのサポート
 - 日付、文字と数値、E メール・アドレス、主要クレジット・カード協会のクレジット・カード番号、カナダ、スペイン、フランス、イタリア、英国、米国の国民 ID を対象としたルール・ベースのデータ・マスキング
 - 各国特有の置換値をもとに、置換値でデータをマスクするルックアップ・ポリシー
 - 日付、文字と数値、個人 ID をマスクするためのデータ・プライバシー・モデル
 - 住所、名、姓、会社名などの共通属性を対象とした組み込みの置換データ
- Optim 相互運用性モデル (OIM) を作成するためのサポート。相互運用性モデルは、Optim と IBM® Optim for z/OS のデータ管理要求 (抽出、挿入、削除、ロード、アーカイブ、および復元) の実行、エクスポート、およびインポートのための機能です。それらの要求は、Optim デザイナーから直接実行され、Optim 製品と Optim for z/OS 製品のランタイムが必要です。
- サブセット・サービスまたはデータ変換サービスを作成して実行する機能 (これらサービスをサポートするソリューションの場合)。それらのサービスは、Optim デザイナーから実行するか、 Optim 管理コンソールから公開して実行します。リレーショナル・サービスのテストは、 IBM DB2® for Linux®, UNIX®, and Windows®, DB2 for z/OS, IBM Informix®, Oracle, Sybase、および Microsoft® SQL Server の各データベースで実施されています。

始めに

まず、Data Project Explorer でデータ設計プロジェクトを作成する必要があります。プロジェクトを使用してオブジェクトを作成することで、リレーショナル・データをマスクできるようになります。

データ設計プロジェクトの作成

データ・モデルまたはその他のデータ設計オブジェクトを作成する前に、オブジェクトを保管するためのデータ設計プロジェクトを作成します。

データ設計プロジェクトは主に、モデリング・オブジェクトを保管するのに使用されます。次のタイプのオブジェクトをデータ設計プロジェクトに保管できます。

- 論理データ・モデル
- 物理データ・モデル
- データ管理サービス要求

- Optim 相互運用性モデル
- ドメイン・モデル
- グロッサリー・モデル
- SQL スクリプト (DDL スクリプトを含む)
- (製品によって、Information Integrator がインストールされている場合) マッピング・モデルおよび XML スキーマ

その他のファイル・タイプ (doc ファイル、テキスト・ファイル、プレゼンテーション、またはスプレッドシート) もデータ設計プロジェクトに保管できます。上記以外のファイルまたは Eclipse 固有のファイル (.project ファイルなど) 以外の他のファイルは、Data Project Explorer のデータ設計プロジェクト以下にある「**その他のファイル (Other files)**」フォルダーに表示されます。

データ設計プロジェクトを作成する前に、データ・ソース・エクスプローラーでデータベース接続を作成する必要はありません。ただし、データ設計プロジェクトでよく行う操作では (例、物理データ・モデルのリバース・エンジニアリングなど)、データベース接続を必要とするものがあります。

「新規データ設計プロジェクト (New Data Design Project)」ウィザードを使用することで、名前やファイルを保管するローカル・ディレクトリーなど、データ設計プロジェクトの基本情報を指定します。プロジェクト参照を指定することもできます。

これは、最初に行う必要があります。

データ設計プロジェクトを作成するには、次のようにします。

1. メインメニュー・バーで、「ファイル」→「新規」→「データ設計プロジェクト (Data Design Project)」をクリックします。あるいは、Data Project Explorer の空きスペースを右クリックして、「新規」→「プロジェクト」→「データ設計プロジェクト (Data Design Project)」を選択することもできます。「新規データ設計プロジェクト (New Data Design Project)」ウィザードが開きます。
2. ウィザードのステップを完了して、「終了」をクリックします。データ設計プロジェクトが Data Project Explorer ビューに表示されます。

リレーショナル・データベース内のデータのマスキング

Optim デザイナーを使用して、機密のリレーショナル・データをマスクするデータ・プライバシー・ポリシーを定義できます。

リレーショナル・データベースのデータをマスクするには、次のようにします。

1. 9 ページの『第 2 章 データ・ソース接続の定義』で説明されているように、リレーショナル・データベースへのデータ・ソース接続を定義します。
2. 13 ページの『リバース・エンジニアリングに基づく物理データ・モデル』で説明されているように、データベースからのリバース・エンジニアリングに基づく物理データ・モデルを定義します。
3. 15 ページの『Optim プロパティがある新規論理データ・モデルの作成』で説明されているように、Optim プロパティがある論理データ・モデルを定義します。
4. 16 ページの『データ・アクセス計画の作成』で説明されているように、データ・アクセス計画を定義します。
5. 16 ページの『選択ポリシーでの作業』で説明されているように、選択ポリシーを定義します。
6. 43 ページの『第 6 章 データ・プライバシー・ポリシーの使用』で説明されているように、プライバシー・ポリシーを定義します。

プライバシー・ポリシーを定義した後で、 Optim 相互運用性モデルまたはデータ管理サービスで論理データ・モデルを使用して、データをマスクすることができます。

Optim パースペクティブ

Optim デザイナーの Optim パースペクティブは、データ・モデルおよびプライバシー・ポリシーを定義するために必要なツールを提供します。インストール後に Optim Studio を初めて開いたときに、デフォルトで Optim パースペクティブが表示されます。

Optim パースペクティブには以下のビューがあります。

Data Project Explorer

Data Project Explorer を使用して、データ・オブジェクト、プライバシー・ポリシー、 Optim 相互運用性モデル、およびデータ管理サービスを定義します。

データ・ソース・エクスプローラー

データ・ソース・エクスプローラーを使用して、データ・ソースへの接続を定義します。

Optim パースペクティブから他の場所にナビゲートした後、Optim パースペクティブに戻るには、「ウィンドウ」 → 「パースペクティブを開く」 → 「その他」をクリックします。「パースペクティブを開く」ウィンドウで「Optim」を選択します。

Data Project Explorer

Data Project Explorer では、データ・オブジェクトをローカルで処理できます。

Data Project Explorer は、以下のプロジェクトを表示します。

データ設計プロジェクト

データ設計プロジェクトは、データベース設計および情報統合のために使用されます。このタイプのプロジェクトを使用して、物理データ・モデル、論理データ・モデル、ドメイン・モデル、グロッサリー・モデル、XSD モデル、およびスクリプトを開発します。

- リレーショナル・データベースからのリバース・エンジニアリングに基づく物理データ・モデルを使用して、 Optim プロパティのある論理データ・モデルを作成します。物理データ・モデルは、データベース・サーバーにデプロイ可能な DDL ステートメントを生成するために使用できます。
- Optim プロパティのある論理データ・モデルを使用します。これは、データ・アクセス計画が含まれる論理データ・モデルです。データ・アクセス計画には、データの選択とマスキングのためのポリシー、およびデータベース接続を識別するデータ・ストア・ポリシーが含まれます。

論理データ・モデルは、組織がデータを収集するときの対象となる物事とそれらの複数の物事間の関係を記述するデータベースに固有のものではありません。物理データ・モデルまたは UML モデルを論理データ・モデルから生成できます。

- サービス要求を使用して、データ管理サービスを定義します。
- Optim 相互運用性モデルを使用して、 Optim および Optim for z/OS 用の Optim 要求を定義して処理します。
- ドメイン・モデルを使用して、組織における許可されたアトミック・ドメイン・タイプとそれらの制約に関して記述します。アトミック・ドメインは、物理および論理データ・モデルのデータ・タイプとして指定できます。アトミック・ドメインは、論理データ・モデルの統合部分としても指定できます。

- グロッサリー・モデルは、命名規格準拠のためのデータ・モデルの検証、または命名規則の判別に使用します。

データ開発プロジェクト

データ開発プロジェクトは、Optim ディレクトリーを組み込むため、およびデータベース・アプリケーション開発のために使用されます。このタイプのプロジェクトは、データ・ソース・エクスプローラーの単一の接続に関連付けられます。データ開発プロジェクトを使用して、以下のタスクを実行します。

- Optim ディレクトリーをインポートして管理できます。
- ストアード・プロシージャおよびユーザー定義関数を開発、テスト、およびデプロイできます。
- ターゲット・サーバーが XML をサポートする場合、XML アプリケーション用の XML ファイルおよび成果物を開発できます。
- SQL 照会を開発してテストすることもできます。
- SQL スクリプトまたはストアード・プロシージャを使用してデータにアクセスする Web サービスを開発およびデプロイできます。

Data Project Explorer を使用すると、以下のタスクを実行することもできます。

- データ・オブジェクトの影響および従属関係の分析
- モデルの保全性を確保するためのデータ・モデルの分析
- 2 つのデータ・オブジェクトの比較
- データ・オブジェクトまたはデータ・モデル用の DDL の生成
- データ・ソース・エクスプローラーから、または Data Project Explorer 内からのデータベース・オブジェクトのドラッグ・アンド・ドロップ
- ソース制御システムを使用したプロジェクトの共有

データ・ソース・エクスプローラー

データ・ソース・エクスプローラーで、既存のデータベースに接続してその設計およびオブジェクトを表示できます。

データベース設計を参照して、Data Project Explorer にインポートしてから、それらの設計を拡張または変更することができます。ストアード・プロシージャおよびユーザー定義関数を実行して、「SQL の結果」ビューで結果を表示することもできます。

データ・ソース・エクスプローラーを使用すると、以下のタスクを実行できます。これらのタスクのいくつかは、データ・ソース・エクスプローラーを使用する一部の製品ではサポートされません。

- データベース接続を作成および管理し、接続に含まれるデータ・オブジェクトを参照する。
- データ・オブジェクトを変更し、変更を管理する。
- データ管理サービスをテストするためのネイティブ・データ・ソース接続を定義する。
- ローカル Optim 管理データ・ソースを定義する。
- データ・オブジェクト・メタデータを、データ・プロジェクトにエクスポートする。そこでは、オブジェクトを変更して再デプロイできます。
- SQL 照会およびルーチンを作成、実行、および調整する。

以前のリリースの Optim 定義の使用

以前のリリースの定義を使用することができます。

以前のリリースの Optim 定義を使用するには、ワークスペースから既存のデータ設計プロジェクトに定義を移行する必要があります。

サービス要求およびリレーショナル・データ・アクセス・モデルの移行が可能です。

- サービス要求 (.svc) は、新規のサービス要求に移行されます。要求に関連するソースおよびターゲットのデータ・モデルから新規の論理データ・モデルも作成されます。
- リレーショナル・データ・アクセス・モデル (.rdsm) は新規の論理データ・モデルに移行されます。

以前のリリースの Optim 定義の移行:

以前のリリースの Optim 定義を既存のデータ設計プロジェクトに移行することができます。ワークスペースのすべての定義、または選択した定義の移行ができます。

定義を移行するには、次のようにします。

1. 「移行 (Migrate)」 → 「Optim 1.x ワークスペースの移行 (Migrate Optim 1.x Workspaces)」をクリックします。「Optim ワークスペースの移行 (Migrate Optim Workspace)」ウィザードが開きます。
2. ウィザードのステップを完了します。

定義のターゲット・ディレクトリーを選択する必要があります。このディレクトリーは、定義をデータ設計プロジェクトにインポートするのに使用します。

3. 「ファイル」 → 「インポート」をクリックします。「インポート」ウィザードが開きます。
4. 「一般」ノードを展開して、「ファイル・システム (File System)」を選択します。
5. ウィザードのステップを完了します。

移行した定義を含むディレクトリーをインポート元を使用します。データ設計プロジェクトをインポート先に選択します。

サンプル・データ

Optim には、置換データと事前定義のソース/ターゲット・データ・ソースを含んだサンプルの Derby データベースが含まれています。

サンプル・データベース

デフォルトでは、サンプル・データベースを管理する Derby データベース・サーバーを Optim デザイナーが自動的に実行します。データ・ソース・エクスプローラー中、サンプル・データベースのプロファイル名は Optim サンプル・データベースとされています。

サンプル・データベースは、Optim ワークスペースの `¥.metadata¥plugins¥com.ibm.nex.designer.ui/database/optim` ディレクトリーにあります。データベースには以下のスキーマが含まれます。

- OPTIMUSER - 顧客、注文、在庫、および出荷データの関連表。
- OPTIMUSER2 - OPTIMUSER スキーマの表と一致するメタデータを持つスキーマ。OPTIMUSER がデータ管理サービスのソース・スキーマである場合は、このスキーマを宛先として使用できます。

EXTENDED_LOOKUP 表

EXTENDED_LOOKUP スキーマには、住所や名前、国民 ID 番号、生年月日などの個人データをマスクするためのルックアップ表が含まれます。個人データの各カテゴリーは、国ごとに分けられたいくつかの表で示されます。例えば、ある表には米国の住所が含まれ、別の表にはドイツの住所が含まれます。

Infosphere Data Architect のインストール・ディレクトリーにある `optim¥designer¥sampledata` ディレクトリーには、EXTENDED_LOOKUP スキーマに表を作成できるようにする `.ddl` および `.data` ファイルが入っています。これらの表は、データベース・ベンダーによって提供される対話式のツールを使って作成します。

`optim¥designer¥sampledata` ディレクトリーには以下のサブディレクトリーが含まれています。

extended_lookup

EXTENDED_LOOKUP スキーマの各表の `.data` ファイルが入ります。

extended_lookup_schemas

各データベース・タイプの EXTENDED_LOOKUP スキーマを作成するための `.ddl` ファイルが入ります。

ルックアップ表

EXTENDED_LOOKUP スキーマには、Optim ルックアップ・ポリシーと共に使用できるルックアップ表が含まれています。

個人データの各カテゴリーは、オーストラリア (AU)、カナダ (CA)、フランス (FR)、ドイツ (DE)、イタリア (IT)、日本 (JP)、スペイン (ES)、英国 (UK)、米国 (US) 用にそれぞれ別個の表になっています (括弧内は略語)。各表には連続番号の入った列が含まれており、この列はルックアップ表の行の選択にハッシュ値を使用するルックアップ・ポリシーと共に使用されます。

スキーマでの各表名は、略語による国別接頭部とカテゴリーから成ります (*countryabbreviation_category*)。例えば、カナダ用の住所表の名前は `CA_ADDRESSES`、ドイツ用の住所表の名前は `DE_ADDRESSES` です。

スキーマには以下のカテゴリーが含まれます。

`ADDRESSES` - 番地、市区町村、地域 (例えば州などの行政区分)、郵便番号の列が含まれます。

`FIRSTNAME` - 男性と女性のファーストネームの列が含まれます。

`FIRSTNAME_F` - 女性のファーストネームの列が含まれます。

`FIRSTNAME_M` - 男性のファーストネームの列が含まれます。

`LASTNAME` - ラストネームの列が含まれます。

`PERSON` - 生年月日、ファーストネーム、ラストネーム、性別、電話番号、国民 ID 番号、会社名、および E メール・アドレスの列が含まれます。

データベース・サポート

Optim デザイナーは、以下のデータベースをサポートします。

Optim デザイナーは、以下のデータベースのための JDBC 接続をサポートします。

- DB2 for z/OS V8.1、V9.1
- DB2 for Linux, UNIX, and Windows V8.2、V9.1、V9.5
- DB2 for i V5.4
- Informix V10

- Oracle V10.2、V11
- Sybase V12.5、V15
- Microsoft SQL Server 2005

Optim デザイナーは、以下のデータベースのためのネイティブ・データ・ソース接続をサポートします。

- DB2 for Linux, UNIX, and Windows V9.5
- Oracle V10.2

アクセシビリティ機能

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーまたはその他の特殊な必要があるユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。

Optim デザイナーは、Eclipse 環境で使用可能なアクセシビリティ機能を使用します。

作業時のヘルプ

Eclipse ヘルプ・システムを使用すると、Optim デザイナーでの作業に役立ちます。

第 2 章 データ・ソース接続の定義

データ・ソース・エクスプローラーを使用して、データ・ソース接続を定義します。

Optim デザイナーは JDBC を使用してデータベースに直接接続します。 Optim 相互運用性モデルまたはデータ管理サービスでデータ・モデルを使用するには、モデルを JDBC 接続プロファイルに関連付ける必要があります。

ネイティブ・データ・ソース接続をデータ管理サービスのテストに使用するために定義できます。ネイティブ・データ・ソース接続は、データベース・クライアント接続に基づいています。ネイティブのデータベース・クライアント接続を使用するには、データベースのクライアントが Optim デザイナーのマシンにインストールされている必要があります。

JDBC 接続の詳細

以下の JDBC 接続情報を使用して、リレーショナル・データベースに接続します。

JDBC 接続プロファイルを定義すると、「URL」には、使用データベースに合ったサンプル構文が自動的に設定されます。この構文を編集してデータベースの接続情報を指定する必要があります。

DB2 for Linux, UNIX, and Windows

DB2 for Linux, UNIX, and Windows の場合は、データベースのホスト名、ポート、およびデータベース名を、次の形式で指定する必要があります。

```
jdbc:db2://host:port/database
```

例えば、次のようにします。

```
jdbc:db2://PRODZ:50000/PRODDb
```

DB2 for i

DB2 for i の場合は、データベースのホスト名およびポートを次の形式で指定する必要があります。

```
jdbc:as400:host;prompt=false
```

例えば、次のようにします。

```
jdbc:as400:ihost;prompt=false
```

DB2 for z/OS

DB2 for z/OS の場合は、データベース・ホスト名、ポート、およびロケーション (データベース) 名を、次の形式で指定する必要があります。

```
jdbc:db2://host:port/location:  
retrieveMessagesFromServerOnGetMessage=true;
```

例えば、次のようにします。

```
jdbc:db2://PRODZ:1523/PRODDb:  
retrieveMessagesFromServerOnGetMessage=true;
```

Derby (サンプル・データベース)

サンプル・データベースの場合は、ホスト名、ポート、およびホスト (管理サーバー) 上のデータベース・ディレクトリーを指定する必要があります。ディレクトリーは、Optim インストール・ディレクトリーの *optim¥databases¥optim* にあります。次の形式を使用します。

```
jdbc:derby:net://host:port/directory
```

例えば、次のようにします。

```
jdbc:derby:net://localhost:1527/C:¥Program Files¥IBM¥optim¥databases¥opt
```

Informix

Informix の場合は、ホスト名、ポート、データベース名を、次の形式で指定する必要があります。

```
jdbc:informix-sqli://host:port/database:INFORMIXSERVER=server
```

例えば、次のようにします。

```
jdbc:informix-sqli://PRODIFX:1523/IFX94:INFORMIXSERVER=server
```

Microsoft SQL Server

Microsoft SQL Server の場合は、ホスト名、ポート、およびデータベース名を、次の形式で指定する必要があります。

```
jdbc:microsoft:sqlserver://host:port;databaseName=dbname
```

例えば、次のようにします。

```
jdbc:microsoft:sqlserver://SQLSRV1:1433/databaseName=SQL2005
```

Optim Database Relationship Analyzer (Derby)

Optim Database Relationship Analyzer データベース (Derby サーバー上でホスト処理される) の場合は、ホスト名、ポート、およびホスト (Optim Database Relationship Analyzer サーバー) 上のデータベース・ディレクトリーを指定する必要があります。ディレクトリーは、Optim Database Relationship Analyzer インストール・ディレクトリーの *optim¥databases¥optim* にあります。次の形式を使用します。

```
jdbc:derby:net://host:port/directory
```

例えば、次のようにします。

```
jdbc:derby:net://DRASRV1:49952/C:¥Program Files¥IBM¥DatabaseRelationshipAnalyzer1.1¥server¥GRDDBV1
```

Oracle

Oracle の場合は、サーバー名、ポート、データベース名を、次の形式で指定する必要があります。

```
jdbc:oracle:thin:@server:port:database
```

例えば、次のようにします。

```
jdbc:oracle:thin:@PRODORA:1523:ORA101
```

Sybase

Oracle の場合は、サーバー名、ポート、データベース名を、次の形式で指定する必要があります。

```
jdbc:oracle:thin:@server:port:database
```


例えば、次のようにします。

```
jdbc:oracle:thin:@PRODORA:1523:ORA101
```

ネイティブ・データ・ソース接続での作業

デフォルトでは、論理データ・モデルはリレーショナル・データベース用の JDBC 接続を使用します。処理を高速化するためには、Optim デザイナーからサービスを実行するときにネイティブ・データ・ソース接続を選択します。

ネイティブ・データ・ソース接続は、データベース・クライアント接続に基づいています。ネイティブのデータベース・クライアント接続を使用するには、データベースのクライアントが Optim デザイナーのマシンにインストールされている必要があります。

データ・ソース・エクスプローラーに定義された既存の JDBC 接続プロファイルに基づいて、ネイティブ・データ・ソース接続を自動的に定義できます。

ウィザードを使用して、手動でネイティブ・データ・ソース接続を定義することもできます。

ネイティブ・データ・ソース接続の自動的な定義

「データ・ソース・エクスプローラー」ビューで定義された既存の JDBC 接続プロファイルを使用して、ネイティブ・データ・ソース接続を自動的に定義できます。

ネイティブ・データ・ソース接続を自動的に定義するには、データ・ソースへの既存の JDBC 接続が必要です。

ネイティブ・データ・ソース接続を自動的に定義するには、次のようにします。

1. データ・ソース・エクスプローラーで、「データベース接続」フォルダーを拡張します。
2. データ・ソースへの JDBC 接続を右クリックして、「ネイティブ・データ・ソースの接続プロファイルの作成」をクリックします。新規のネイティブ・データ・ソース接続プロファイルが、「ODA データ・ソース」フォルダーにある「ネイティブ・データ・ソース」フォルダーの下に表示されます。

ネイティブ・データ・ソース接続の手動での定義

「新規のネイティブ・データ・ソース接続 (New Native Data Source Connection)」ウィザードを使用して、ネイティブ・データ・ソース接続を定義できます。

ネイティブのデータベース・クライアント接続を使用するには、データベースのクライアントが Optim デザイナーのマシンにインストールされている必要があります。

ネイティブ・データ・ソース接続を手動で定義するには、次のようにします。

1. データ・ソース・エクスプローラーで、「ODA データ・ソース」フォルダーを拡張します。
2. 「ネイティブ・データ・ソース」フォルダーを右クリックして、「新規」をクリックします。「新規のネイティブ・データ・ソース接続 (New Native Data Source Connection)」ウィザードが開きます。
3. ウィザードのステップを完了します。

データベース・タイプを選択して、接続ストリング、文字フォーマット、およびユーザー・クレデンシャルを入力する必要があります。接続をテストできます。

第 3 章 データ・モデルの管理

Optim 相互運用性モデルを定義する、またはデータ管理サービス要求を処理するには、論理データ・モデルを使用してソースまたはターゲットのデータを定義する必要があります。論理データ・モデルには、データベース接続を識別するメタデータが含まれている必要があります。

論理モデルに接続情報が含まれない場合、データ管理サービスを定義するときにその情報を指定できます。

Optim プロパティのある論理データ・モデルは、データ・アクセス計画が含まれる論理データ・モデルです。データ・アクセス計画には、データの選択とマスキングのためのポリシー、およびデータベース接続を識別するデータ・ストア・ポリシーが含まれます。

以下のいずれかの物理データ・モデルを変換することにより、Optim プロパティのある新しい論理モデルを作成できます。

- データ・ソース接続からのリバース・エンジニアリングによって作成された物理データ・モデル
- Optim Database Relationship Analyzer の物理データ・モデル

リバース・エンジニアリングに基づく物理データ・モデル

データベースからのリバース・エンジニアリングによって作成された物理モデルを使用して、Optim プロパティのある論理モデルを作成します。データベースには、Data Source Explorer で定義された接続プロファイルが必要です。

リバース・エンジニアリングに基づく物理データ・モデルの定義

「新規物理データ・モデル」ウィザードを使用して、データベースまたは DDL ファイルからのリバース・エンジニアリングに基づく物理データ・モデルを定義できます。

データベースには、Data Source Explorer で定義された接続プロファイルが必要です。

リバース・エンジニアリングに基づく物理データ・モデルを定義するには、次のようにします。

1. メインのファイル・メニューから、「ファイル」→「新規」→「物理データ・モデル」をクリックします。「新規物理データ・モデル」ウィザードが開きます。
2. 「モデル・ファイル (Model File)」ページで、新しいモデル・ファイルのデータベース、バージョン、および場所を指定します。
3. 「リバース・エンジニアリングから作成 (Create from reverse engineering)」を選択します。

データベースからのリバース・エンジニアリングを選択する場合、ウィザードの次のページで接続情報を指定する必要があります。DDL ファイルからのリバース・エンジニアリングを選択する場合、ウィザードの次のページで DDL ファイルへのパスを指定する必要があります。

4. ウィザードのステップを完了します。

物理データ・モデルは、「データ・モデル」フォルダーに作成されて表示されます。

Database Relationship Analyzer 物理データ・モデルの使用

Optim Database Relationship Analyzer メタデータ・データベースに定義された関連表のグループに基づく物理データ・モデルを作成できます。

Optim Database Relationship Analyzer グループ・ディスカバリー・プロセスを使用すると、リレーショナル・データベース・スキーマ内の関連する表を参照するグループを作成できます。 Database Relationship Analyzer 物理データ・モデルでグループを使用するには、そのグループが含まれている Optim Database Relationship Analyzer メタデータ・データベースおよびメタデータ・スキーマを指定する Optim Database Relationship Analyzer 接続プロファイルを定義する必要があります。

Database Relationship Analyzer 物理データ・モデルを定義するには、グループが含まれている Optim Database Relationship Analyzer 接続プロファイルを選択し、モデルの表を提供するグループを選択してから、グループで定義された表が含まれるデータベースへの接続を選択します。

Optim Database Relationship Analyzer のセットアップ

Optim デザイナーで Optim Database Relationship Analyzer を使用するには、次のようにします。

1. Optim Database Relationship Analyzer サーバーをインストールし、 Optim デザイナーで使用するリレーショナル・データのグループ・ディスカバリー・プロセスを実行します。
2. データ・ソース・エクスプローラーを使用して、 Optim Database Relationship Analyzer メタデータ・データベースへのデータ・ソース接続を定義します。
3. Optim の設定を使用し、 Optim Database Relationship Analyzer データ・ソース接続とメタデータ・スキーマに基づいた接続プロファイルを定義します。 Optim Database Relationship Analyzer バージョン 1.1.1 以降で作成したメタデータ・データベースの接続プロファイルのみを定義できます。

Optim Database Relationship Analyzer 接続プロファイルの定義

データ・ソース・エクスプローラーで作成したデータ・ソース接続を使用して、 Optim Database Relationship Analyzer 接続プロファイルを定義できます。

Optim Database Relationship Analyzer 接続プロファイルでデータ・ソースを定義するには、まず、 Optim デザイナー・マシンを Optim Database Relationship Analyzer メタデータ・データベースへ接続する必要があります。

Optim Database Relationship Analyzer バージョン 1.1.1 以降で作成したメタデータ・データベースの接続プロファイルのみを定義できます。

データ・ソース接続を Optim Database Relationship Analyzer 接続プロファイルとして定義するには、次のようにします。

1. 「Optim デザイナー」メニューから、「ウィンドウ」 → 「設定」の順にクリックして「設定」ウィンドウを開きます。
2. 「設定」ウィンドウのオプション・リストで、「Optim」ノードを展開し、「Database Relationship Analyzer」を選択します。「Database Relationship Analyzer の設定 (Database Relationship Analyzer Preferences)」ページが開きます。
3. 「Database Relationship Analyzer の設定 (Database Relationship Analyzer Preferences)」ページで、「追加」をクリックします。「Database Relationship Analyzer 接続プロファイルの追加」ウィンドウが開きます。
4. リストからデータ・ソース接続を選択します。 Optim デザイナーがデータ・ソースからスキーマを取得した後、「選択されたプロファイルのスキーマを選択します」リストが使用可能になります。

5. Optim Database Relationship Analyzer メタデータ・スキーマを選択します。
6. 「追加」をクリックして、「Database Relationship Analyzer の設定 (Database Relationship Analyzer preferences)」ページに戻ります。このページには、「Database Relationship Analyzer 接続プロファイルの追加」ウィンドウで選択した接続とメタデータ・スキーマがリストされます。
7. 「適用」または「OK」をクリックして、Optim Database Relationship Analyzer 接続プロファイルを保存します。

Database Relationship Analyzer 物理データ・モデルの作成

「新規 Database Relationship Analyzer 物理データ・モデル」ウィザードを使用して、Optim Database Relationship Analyzer メタデータ・データベースに定義された関連表のグループに基づく物理データ・モデルを作成できます。

Optim Database Relationship Analyzer メタデータ・データベースには、Database Relationship Analyzer 設定に定義された接続プロファイルが必要です。

Database Relationship Analyzer グループで選択された表が入っているソース・データベースには、Data Source Explorer で定義された接続プロファイルが必要です。

Database Relationship Analyzer 物理データ・モデルを作成するには、次のようにします。

1. Data Project Explorer で「データ・モデル」フォルダーを右クリックして、「新規」→「Database Relationship Analyzer 物理モデル」をクリックします。「新規 Database Relationship Analyzer 物理データ・モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

モデルの関係マッピングを提供する、Database Relationship Analyzer 接続プロファイルおよび Database Relationship Analyzer グループを選択する必要があります。また、モデルのソース・データベースを選択して、それに接続する必要もあります。ソース・データベースには、モデルのために選択された Database Relationship Analyzer グループ内のすべての表が含まれている必要があります。

物理データ・モデルは、「データ・モデル」フォルダーに作成されて表示されます。

Optim プロパティがある新規論理データ・モデルの作成

「論理データ・モデルへの変換」ウィザードを使用して、Optim プロパティがある新しい論理データ・モデルを作成できます。

以下のいずれかの物理データ・モデルを変換することにより、Optim プロパティのある新しい論理モデルを作成できます。

- データ・ソース接続からのリバース・エンジニアリングによって作成された物理データ・モデル
- Optim Database Relationship Analyzer の物理データ・モデル

Optim プロパティがある新規論理モデルを作成するには、次のようにします。

1. 「データ・モデル」フォルダーを拡張して、物理データ・モデルを右クリックしてから、「Optim プロパティがある論理データ・モデルに変換」をクリックします。「論理データ・モデルへの変換」ウィザードが開きます。
2. ウィザードの最初のページで、「新規モデルの作成」を選択します。
3. ウィザードのステップを完了します。

論理データ・モデルは、「データ・モデル」フォルダーに作成されて表示されます。

データ・アクセス計画の作成

データ・アクセス計画には、Optim 相互運用性モデルまたはデータ管理サービスのソース論理データ・モデルで、どのデータを処理または変換するかを決定するためのポリシーが含まれます。

論理データ・モデルのための選択ポリシーおよびデータ・プライバシー・ポリシーを定義できます。

選択ポリシーによって、Optim 相互運用性モデルまたはデータ管理サービスで使用するエンティティおよび属性が決まります。

データ・アクセス計画の追加

「データ・アクセス計画の追加」ウィザードを使用して、Optim プロパティがある論理データ・モデルにデータ・アクセス計画を追加できます。

データ・アクセス計画を論理データ・モデルに追加するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージを拡張します。
2. 「データ・アクセス計画」フォルダーを右クリックして、「データ・アクセス計画の追加」をクリックします。「データ・アクセス計画の追加」ウィザードが開きます。
3. データ・アクセス計画の名前を入力します。
4. 「終了」をクリックします。データ・アクセス計画エディターが開いて、選択ポリシーをプランに追加できます。

データ・アクセス計画のポリシーの編集

ポリシー・エディターを使用して、データ・アクセス計画のポリシーを編集できます。

データ・アクセス計画のポリシーを編集するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックして、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ソース・ポリシー」リストからポリシーを選択して、「ポリシーの編集」をクリックします。エディターまたは選択されたポリシーが開きます。
4. ポリシーを編集してから、「ファイル」→「保存」をクリックして変更内容を保存します。

選択ポリシーでの作業

選択ポリシー・エディターを使用して、選択ポリシーを編集します。

選択ポリシーによって、Optim 相互運用性モデルまたはデータ管理サービスで使用するエンティティおよび属性が決まります。

選択基準はエンティティ内の属性に適用できます。選択基準では、SQL の select ステートメントを使用して、属性内の行からデータを選択します。

選択ポリシーには、以下のエンティティ・タイプが含まれます。

開始エンティティ

開始エンティティは、処理時にデータが最初に選択されるエンティティです。関連エンティティのデータの選択は、開始エンティティとの関係に基づいて行われます。開始エンティティを選択すると、すべての関連エンティティがポリシーに追加されます。

関連エンティティ

関連エンティティは、開始エンティティとの関係に基づいてデータが選択されるエンティティです。

参照エンティティ

参照エンティティは、開始エンティティとの関係にかかわらず、処理時にすべての属性が選択されるエンティティです。

選択ポリシーの作成

データ・アクセス計画のための選択ポリシーを作成できます。

選択ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。ポリシー・エディターが開きます。
3. 「**選択ポリシーの追加**」をクリックします。「選択ポリシーの追加」ウィザードが開きます。
4. 選択ポリシーの名前を入力します。
5. 「**終了**」をクリックします。選択ポリシー・エディターが開いて、ポリシーの開始エンティティを選択できます。
6. 「**開始エンティティ**」で、開始エンティティの名前を入力するか、または「**参照**」をクリックして「開始エンティティの選択」ウィンドウを開きます。「開始エンティティの選択」ウィンドウを使用するには、次のようにします。
 - a. 論理データ・モデルを拡張して、開始エンティティを選択します。開始エンティティは、任意の論理データ・モデルに配置できます。
 - b. 「**終了**」をクリックします。

開始エンティティを選択した後、選択ポリシー・エディターが開きます。

7. 選択ポリシー・エディターを使用して、ポリシーを編集します。
8. 「**ファイル**」 → 「**保存**」をクリックして、ポリシーを保存します。

選択ポリシー・エディター:

選択ポリシー・エディターを使用して、選択ポリシーを定義します。

選択ポリシーによって、データ管理サービスで使用されるエンティティおよび属性が決まります。

エンティティ

「エンティティ」領域には、ポリシーに対して選択されたエンティティが表示されます。選択されたエンティティは、エンティティが含まれる論理モデルに従ってリストされます。

開始エンティティ

ポリシーの開始エンティティを入力するか、または「**参照**」をクリックして開始エンティティ

を選択します。開始エンティティは、処理時にデータが最初に選択されるエンティティです。関連エンティティのデータの選択は、開始エンティティとの関係に基づいて行われます。

エンティティの追加

「エンティティの追加」ウィザードが開きます。エンティティをポリシーに追加できます。

除去 エンティティ・リストで選択されたエンティティを除去します。親を持たない関連エンティティもすべて除去されます。

エンティティ・リスト

ポリシーに対して選択されたエンティティが、ソース論理データ・モデルに従ってリストされます。エンティティを選択すると、「**エンティティ仕様**」領域に属性が表示されます。

タイプ 開始エンティティに基づくエンティティ・タイプ。「**関連 (Related)**」は、エンティティ内のデータが開始エンティティとの関係に基づいて選択されることを示します。

「**参照**」は、エンティティには開始エンティティとの関連がなく、エンティティ内のすべてのデータが選択されることを示します。

名前 エンティティの名前。

関係/マップ

エンティティに関連付けられた関係名またはマップ名。

エンティティ仕様

「**エンティティ仕様**」領域には、エンティティ・リストで選択されたエンティティの属性がリストされます。この領域を使用して、1 つ以上の属性の選択基準を指定します。選択基準によって、エンティティから選択される行が決まります。

エンティティ・パス

データ設計プロジェクト内の選択されたエンティティへのパス。

すべての基準の結合手段

複数の列の基準を結合するオプション。以下のようになります。

AND すべての属性において行が選択基準と一致する必要があります。

OR 1 つの属性において行が選択基準と一致する必要があります。

属性リスト

エンティティ・リストで選択されたエンティティの属性が表示されます。

名前 属性名。

データ・タイプ

属性データ・タイプ。

選択基準

属性に適用する選択基準を入力します。適切な演算子と値、または置換変数の仕様を使用します。選択基準は、SQL 構文に準拠していること、および関係演算子または論理演算子を含むことが必要です。論理演算子および構文は、DBMS によって異なります。詳しくは、該当する DBMS 資料を参照してください。

選択基準の要約

選択基準を入力した後に、属性に適用される選択基準の要約が表示されます。

選択ポリシーでの関連エンティティおよび参照エンティティの定義

エンティティ・リストを使用して、関連エンティティおよび参照エンティティを選択ポリシーで定義できます。

関連エンティティおよび参照エンティティを選択ポリシーで定義するには、次のようにします。

エンティティの「タイプ」リストで、「**関連 (Related)**」または「**参照**」を選択します。

選択ポリシーへのエンティティ追加

「エンティティの追加」ウィザードを使用して、選択ポリシーにエンティティを追加できます。

選択ポリシーにエンティティを追加するには、次のようにします。

1. 選択ポリシー・エディターで、「**エンティティの追加**」をクリックします。「エンティティの追加」ウィザードが開きます。
2. 論理データ・モデルを拡張して、エンティティを選択します。
3. 「**終了**」をクリックします。

エンティティ・リストのソース論理データ・モデルの名前の下に、新しいエンティティが表示されます。

選択ポリシー内の開始エンティティの変更

「開始エンティティの選択」ウィザードを使用して、選択ポリシー内の開始エンティティを変更できます。

選択ポリシー内の開始エンティティを変更するには、次のようにします。

1. 選択ポリシー・エディターで、「**開始エンティティ**」フィールドの横にある「**参照**」をクリックします。「開始エンティティの選択」ウィザードが開きます。
2. 論理データ・モデルを拡張して、開始エンティティを選択します。開始エンティティは、任意の論理データ・モデルに配置できます。
3. 「**終了**」をクリックします。

「**開始エンティティ**」フィールドに、新しい開始エンティティが表示されます。

選択ポリシーからのエンティティの除去

エンティティ・リストを使用して、選択ポリシーからエンティティを除去できます。

選択ポリシーからエンティティを除去するには、次のようにします。

1. エンティティ・リストで、エンティティを選択します。
2. 「**除去**」をクリックします。

選択ポリシーでの選択基準の定義

選択ポリシー・エディターの属性リストを使用して、選択基準を定義できます。

選択基準を定義するには、次のようにします。

1. エンティティ・リストで、選択基準を追加する属性が含まれるエンティティを選択します。
2. 基準の適用方法を決定する「**すべての基準の結合手段**」イテレーターを選択します。
 - a. すべての属性において行が選択基準と一致する必要がある場合、「**AND**」を選択します。

- b. 1 つの属性において行が選択基準と一致する必要がある場合、「**OR**」を選択します。
- 3. 属性リストで、属性の基準を「**選択基準**」に入力します。
- 4. 「**選択基準の要約**」エディターで、基準を検討します。

第 4 章 データ管理サービスの設計

データ管理サービスは、ソース論理データ・モデルに記述されているデータを抽出して、ターゲット論理データ・モデルでデータを挿入または更新します。

データ管理サービスを使用して、論理データ・モデルのエンティティに適用されるプライバシー・ポリシーに基づいてデータをマスクできます。また、サービス要求にインライン式を定義することによってデータをマスクすることもできます。Optim 管理コンソールを使用して、サービスを実行したり、サービスのスケジュールを作成したりできます。

データ管理サービスの作成

「新規サービス」ウィザードを使用して、データ管理サービスを作成します。このサービスは論理データ・モデルを使用してソース・データを提供します。

データ管理サービスには、次の 2 つのタイプがあります。

サブセット・サービス

サブセット・サービスは、1 つのデータ・モデルから別のデータ・モデルにデータをコピーします。例えば、サブセット・サービスでフィルター条件をもとにデータをサブセットにできます。

データ変換サービス

データ変換サービスは、ソース・データ・モデルのデータをマスクします。例えば、データ変換サービスによりテスト・データ・モデルのデータをマスクできます。

サブセット・サービスの作成

「新規サービス」ウィザードを使用して、サブセット・サービスを作成できます。

ソースおよびターゲットの論理データ・モデルが必要です。

ソース・モデルには、データ・アクセス計画が含まれている必要があります。「新規サービス」ウィザードを使用することで、モデルの計画を定義できます。

ソース・モデルとターゲット・モデルはそれぞれ、データベース接続を識別するデータ・ストア・ポリシーが含まれている必要があります。「新規サービス」ウィザードを使用することで、モデルのポリシーを定義できます。

サブセット・サービスを作成するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを右クリックして、「新規」→「サービス (Service)」をクリックします。「新規サービス」ウィザードが開きます。
2. ウィザードのステップを完了します。

ソース論理データ・モデルと、モデル内のデータ・アクセス計画を選択する必要があります。

「ターゲット・モデルのオプション」ページを使用して、このサービスのターゲット・データ・モデルを選択する方法、およびターゲット・データ・モデルで実行する操作のタイプを決定します。

ターゲット・モデルの以下の選択オプションを使用できます。

ターゲット・モデルを選択し、自動マッピングを実行します

このオプションは、ターゲット論理データ・モデルを選択し、Optim デザイナーによるソースとターゲット・モデルの自動マップを可能にする場合に使用します。自動マップ機能は、ソース・モデル内のエンティティおよび属性を、ターゲット・モデル内の対応するエンティティおよび属性にマップします。自動マップ機能が機能するためには、ソース・モデルおよびターゲット・モデルのスキーマが類似している必要があります。

ソース・モデルに基づいて Optim 管理データ・ストア・モデルを作成します

このオプションは、管理データ・ストアを表すターゲット論理データ・モデルを作成する場合に使用します。このオプションを選択すると、「**Optim 管理データ・ストア・モデル**」フィールドが表示され、新しい論理データ・モデルの名前を入力できます。

ターゲット・モデルには、ソース・モデルに対応するスキーマが含まれます。このサービスがデータ・ストアにデータを追加した後、関連付けられた論理データ・モデルをサービスのソース・モデルとして使用できます。

サービス計画エディターを開き、ターゲット・モデル選択を手動で完了させて、ソースからターゲットにマップします

このオプションは、サービス計画エディターを開き、その中でターゲット・モデルを選択してソース・モデルからターゲット・モデルにマップする場合に使用します。サービス計画エディターからターゲット・モデルを選択するには、「**ターゲット・モデルの追加 (Add Target Model)**」をクリックします。

以下のターゲット操作を使用できます。

- 挿入** 新しい行をコピー先エンティティに挿入します。コピー元データの行の主キーがコピー先エンティティの行の主キーと一致しない場合に、行が挿入されます。コピー元データの行の主キーがコピー先エンティティの行の主キーと一致する場合、操作は失敗します。
- 更新** コピー先エンティティの既存の行を更新します。コピー元データの行の主キーがコピー先エンティティの行の主キーと一致する場合に、行が更新されます。コピー元データの行の主キーがコピー先エンティティの行の主キーと一致しない場合、操作は失敗します。

データ変換サービスの作成

「新規サービス」ウィザードを使用して、データ変換サービスを作成できます。

ソース論理データ・モデルが必要です。

ソース・モデルには、データ・アクセス計画が含まれている必要があります。「新規サービス」ウィザードを使用することで、モデルの計画を定義できます。

ソース・モデルには、データベース接続を識別するデータ・ストア・ポリシーが含まれている必要があります。「新規サービス」ウィザードを使用することで、モデルのポリシーを定義できます。

データ変換サービスを作成するには、次のようにします。

1. Data Project Explorer で「**サービス**」フォルダーを右クリックして、「**新規**」→「**サービス (Service)**」をクリックします。「新規サービス」ウィザードが開きます。
2. ウィザードのステップを完了します。

ソース論理データ・モデルと、モデル内のデータ・アクセス計画を選択する必要があります。

データ管理サービスの編集

サービス計画を使用してデータ管理サービスを編集します。

サービス計画には、ソース・ポリシー、ターゲット・ポリシー、およびソースからターゲットへのマップが含まれます。サービス計画エディターを使用して、サービス計画を処理します。

ソース・ポリシー

ソース・ポリシーは、ソース論理データ・モデル内のデータ・アクセス計画、またはサービス要求の作成中にソース論理データ・モデルに追加されたメタデータによって提供されます。

ソースからターゲットへのマップ

ソースからターゲットへのマップによって、ソース属性とターゲット属性の間のマッピングが決まります。このマップを使用して、マッピングの編集と、新しいターゲット論理データ・モデルの選択を行えます。

ターゲット・ポリシー

ターゲット・ポリシーには、以下のものが含まれます。

- ターゲットの論理データ・モデルへの接続を記述するデータ・ストア・ポリシー。
- ターゲット・モデル内でソース・エンティティの挿入と更新のどちらを行うかを決定する更新ポリシー。更新ポリシーは編集できます。
- サービスで使用するエンティティに定義された主キーや外部キーなどの制約を使用可能/使用不可にする「制約を使用不可にする」ポリシー
- JavaScript 式を使用して属性のデータ変換を定義する JavaScript ポリシー。
- サービスで生成されるメッセージのオプションを設定するためのサービス診断ポリシー。

サービス計画の編集

サービス計画エディターを使用して、サービス計画のターゲット・ポリシーまたはマッピングを編集できます。

サービス計画を編集するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを展開して、サービス計画が含まれるサービス要求を開きます。
2. 「サービス・プラン」ノードをダブルクリックします。 サービス計画エディターが開きます。
3. ポリシーまたはマッピングを編集してから、「ファイル」→「保存」をクリックして変更内容を保存します。

ソースからターゲットへのマッピングでの作業

ソースからターゲットへのマッピングを使用して、ソース属性とターゲット属性の間のマッピングを決定します。エンティティの追加または除去、新しいターゲット論理データ・モデルの選択、および自動マッピングのリストアを行うこともできます。

ソースからターゲットへのマッピングへのエンティティの追加:

「マップ・エンティティの追加 (Add Map Entity)」ウィザードを使用して、ソースからターゲットへのマッピングにエンティティを追加します。

ソースからターゲットへのマッピングにエンティティを追加するには、次のようにします。

1. サービス計画エディターの「ポリシー」リストで、「ソースからターゲットへのマップ」を選択します。「ソースからターゲットへのマッピング」エディターが開きます。
2. 「マップを追加」をクリックします。「マップ・エンティティの追加 (Add Map Entity)」ウィザードが開きます。
3. ウィザードのステップを完了します。

ソース・エンティティおよびターゲット・エンティティを選択する必要があります。

ターゲット論理データ・モデルの変更:

「ターゲット・モデルの選択」ウィザードを使用して、サービス要求における新しいターゲット論理データ・モデルを選択できます。

以前のターゲット論理データ・モデルに適用された更新ポリシーはすべて除去されます。

ターゲット論理データ・モデルを変更するには、次のようにします。

1. サービス計画エディターの「ポリシー」リストで、「ソースからターゲットへのマップ」を選択します。「ソースからターゲットへのマッピング」エディターが開きます。
2. 「参照」をクリックします。「ターゲット・モデルの選択」ウィンドウが開きます。
3. 論理データ・モデルを選択します。
4. 「OK」をクリックします。

ソースからターゲットへのマッピングからのエンティティの除去:

「マップされたエンティティの除去」ウィンドウを使用して、ソースからターゲットへのマッピングからエンティティを除去します。

ソースからターゲットへのマッピングからエンティティを除去するには、次のようにします。

1. サービス計画エディターの「ポリシー」リストで、「ソースからターゲットへのマップ」を選択します。「ソースからターゲットへのマッピング」エディターが開きます。
2. 「除去」をクリックします。「マップされたエンティティの除去」ウィンドウが開きます。
3. エンティティを選択します。
4. 「OK」をクリックします。

自動マッピングのリストア:

ソースからターゲットへのマッピングにデフォルト・マッピングをリストアすることができます。マッピングに対するそれ以前の編集は、上書きされます。

自動マッピングをリストアするには、次のようにします。

1. サービス計画エディターの「ポリシー」リストで、「ソースからターゲットへのマップ」を選択します。「ソースからターゲットへのマッピング」エディターが開きます。
2. 「自動マッピングのリストア」をクリックします。「自動マッピングのリストア」ウィンドウが開き、この中で、自動マッピングがソースとターゲットの論理データ・モデル内のエンティティおよび属性に基づくことが示されます。
3. 「OK」をクリックします。

制約の管理

「制約を使用不可にする」ポリシーを使用することで、サービスで使用されるエンティティに定義された主キーや外部キーなどの制約を使用可能/使用不可にすることができます。

管理データ・ストアのデータについては、「制約を使用不可にする」ポリシーを定義できません。

「制約を使用不可にする」ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、サービス計画に「制約を使用不可にする」ポリシーを作成できます。

「制約を使用不可にする」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で「**サービス**」フォルダーを展開して、ポリシーを追加するサービス計画を含むサービス要求を開きます。
2. 「**サービス・プラン**」ノードをダブルクリックします。 サービス計画エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。 「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。

エンティティのすべての制約を使用可能/使用不可にする:

「制約を使用不可にする」ポリシー・エディターを使用することで、エンティティのすべての制約を使用可能/使用不可にできます。

エンティティのすべての制約を使用可能/使用不可にするには、次のようにします。

1. Data Project Explorer で「**サービス**」フォルダーを展開して、ポリシーが使用されているサービス計画が含まれるサービス要求を開きます。
2. 「**サービス・プラン**」ノードをダブルクリックします。 サービス計画エディターが開きます。
3. 「**ターゲット・ポリシー (Target Policies)**」リストで、「制約を使用不可にする」ポリシーを選択します。 「制約を使用不可にする」ポリシー・エディターが開きます。
4. エンティティを 1 つ以上選択して、「**すべての制約を使用可能 (Enable All Constraints)**」または「**すべての制約を使用不可 (Disable All Constraints)**」をクリックします。 「**使用不可の制約 (Disabled Constraints)**」列に、各エンティティの使用不可にされた制約が表示されます。

エンティティの選択した制約を使用可能/使用不可にする:

「制約を使用不可にする」ポリシー・エディターを使用することで、エンティティの選択した制約を使用可能/使用不可にできます。

エンティティの選択した制約を使用可能/使用不可にするには、次のようにします。

1. Data Project Explorer で「**サブセット・サービス**」フォルダーを展開して、ポリシーを追加するサービス計画を含むサービス要求を開きます。
2. 「**サービス・プラン**」ノードを右クリックして、「**開く**」をクリックします。 サービス計画エディターが開きます。
3. 「**ターゲット・ポリシー (Target Policies)**」リストで、「制約を使用不可にする」ポリシーを選択します。 「制約を使用不可にする」ポリシー・エディターが開きます。
4. エンティティの「**使用不可の制約 (Disabled Constraints)**」列で、「**...**」をクリックします。 「マップされたエンティティの除去」ダイアログが開きます。

5. 使用不可にするに制約を選択するか、または使用可能にする制約を選択解除します。「OK」をクリックします。「使用不可の制約 (Disabled Constraints)」列に、エンティティの使用不可にされた制約が表示されます。

サービス診断の作業

サービス診断ポリシーを使用することで、サービスで生成されるメッセージのオプションを設定できます。

サービスで生成されるログ・メッセージは、プロキシーがインストールされたワークスペースの `¥.metadata` ディレクトリー内の `.log` ファイルに格納されます。サービス要求の以下のログ・レベルが選択できます。リストはメッセージ重大度に応じて昇順に並べてあります。

ALL すべてのメッセージをログに記録します。

FINEST

非常に詳細なメッセージ。

FINER

かなり詳細なメッセージ。

FINE 詳細メッセージ。

CONFIG

構成に関する静的メッセージ。デバッグに役立ちます。

INFO エンド・ユーザーと管理者のため通知メッセージ。これがデフォルトです。

WARNING

潜在的な問題を示すメッセージ。

SEVERE

重大な障害を示すメッセージ。

OFF ログングをオフにします。

サービス診断ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、サービス計画にサービス診断ポリシーを作成できます。

サービス診断ポリシーを作成するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを展開して、ポリシーを追加するサービス計画を含むサービス要求を開きます。
2. 「サービス・プラン」ノードをダブルクリックします。サービス計画エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。

ログ・レベルおよびサービス通知のオプションを選択します。

更新ポリシーの編集

「更新ポリシー」エディターを使用して、ターゲット・モデル内でソース・エンティティの挿入と更新のどちらが行われるのかを決定する更新ポリシーを編集します。

更新ポリシーを編集するには、次のようにします。

1. サービス計画エディターの「ポリシー」リストで、更新ポリシーを選択します。「更新ポリシー」エディターが開きます。

2. 編集する各ターゲット・エンティティのチェック・ボックスを選択します。

すべてのエンティティを選択するには、「すべて選択」をクリックします。

すべてのエンティティを選択解除するには、「すべての選択解除」をクリックします。

3. 「挿入」または「更新」をクリックして、選択されたエンティティのサービス・アクションを変更します。

データ管理サービスのテスト

Optim デザイナーを使用して、データ管理サービスを Optim 管理コンソールから実行する前にテストできます。

Optim デザイナーからサービスを実行するには、Optim デザイナーのマシンに Optim Executor がインストールされている必要があります。

ソースとターゲットのデータ用のローカル Optim 管理データ・ストアを、Optim デザイナーから実行されるサービスで使用できます。

デフォルトでは、論理データ・モデルは JDBC 接続を使用します。処理を高速化するためには、サービスを実行するときにネイティブ・データ・ソース接続を選択します。

Optim Executor の構成

Optim 設定を使用して Optim Executor のインストール・パスを入力することにより、Optim Executor を構成できます。

Optim Executor を構成するには、次のようにします。

1. 「ウィンドウ」→「設定」をクリックします。「設定」ウィンドウが開きます。
2. ナビゲーション・ツリーで、「Optim」ノードを拡張してから、「Optim Executor」をクリックします。Optim Executor エディターが開きます。
3. 「Executor の場所」フィールドで、Optim Executor の eclipse.exe ファイルへのパスを入力するか、または「参照」をクリックしてパスを選択します。
4. 「OK」をクリックします。

ローカル Optim 管理データ・ストアでの作業

ローカル Optim 管理データ・ストアを、デザイナーから実行されるサービス要求と共に使用できます。ローカル・データ・ソースは、デザイナー・マシン上にあります。

ローカル・データ・ストアをデザイナーから実行されるサービス要求と共に使用するためには、サービス要求でソースまたはターゲット・データ用の Optim 管理データ・ストアを指定する必要があります。

ローカル Optim 管理データ・ストアのデフォルト・ロケーションは、Optim 設定で指定されます。

ローカル・データ・ストアを作成するには、「新規ローカル Optim 管理データ・ソース」ウィザードを使用して、ローカル Optim 管理データ・ストアへの接続を定義します。

ローカル Optim 管理データ・ストアの構成

Optim 設定を使用して、ローカル Optim 管理データ・ストア・ディレクトリーを構成できます。このディレクトリーには、ローカル Optim 管理データ・ストアが含まれています。データ・ストア・ディレクトリーの場所、およびデータを表示するとき表示される行数を定義できます。

ローカル Optim 管理データ・ストアの場所を入力するには、次のようにします。

1. 「ウィンドウ」 → 「設定」をクリックします。「設定」ウィンドウが開きます。
2. ナビゲーション・ツリーで、「**Optim**」ノードを拡張してから、「**Optim 管理データ・ストア**」をクリックします。
3. 「表示する行の最大数」フィールドで、ローカル・データ・ストアでデータを表示するとき表示される行数を指定します。
4. 「**ローカル管理データ・ストア・ディレクトリー (Local managed data store directory)**」フィールドで、ローカル・データ・ストアの場所を指定します。
5. 「**OK**」をクリックします。

ローカル Optim 管理データ・ストアへの接続

データ・ソース・エクスプローラーを使用して、ローカル Optim 管理データ・ソースに接続できます。

ローカル Optim 管理データ・ストアに接続するには、次のようにします。

1. データ・ソース・エクスプローラーで、「**ODA データ・ソース**」および「**ローカル Optim 管理データ・ソース**」フォルダーを拡張します。データ・ソース・プロファイル名が表示されます。
2. データ・ソース・プロファイル名を右クリックして、「**接続**」をクリックします。

ローカル Optim 管理データ・ストアのデータの表示

表データ・エディターを使用して、ローカル Optim 管理データ・ストアに保管されたデータを表示できます。

ローカル Optim 管理データ・ストアでデータを表示する前に、データ・ストアへの接続を確立する必要があります。

表示可能な行数は、Optim 設定で設定されています。

ローカル Optim 管理データ・ストアのデータを表示するには、次のようにします。

1. データ・ソース・エクスプローラーで、「**ODA データ・ソース**」および「**ローカル Optim 管理データ・ソース**」フォルダーを拡張してから、ローカル Optim 管理データ・ストアのプロファイルを拡張してローカル・データ・ストアを表示します。
2. 表示するデータが入っているデータ・ストアおよびスキーマを拡張します。スキーマ内の表が表示されます。
3. 表名を右クリックして、「**OMDS**」 → 「**データの表示**」をクリックします。データが表データ・エディターに表示されます。

データ管理サービスの実行

「Optim サービスの実行」ウィザードを使用して、データ管理サービスを実行できます。

デザイナーからサービスを実行するには、デザイナーのマシンに Optim Executor がインストールされている必要があります。

データ管理サービスを実行するには、次のようにします。

1. 「サービス」フォルダーを開きます。
2. データ管理サービスを右クリックして、「**Optim サービスの実行**」をクリックします。「Optim サービスの実行」ウィザードが開きます。
3. ウィザードのステップを完了します。

サービスが Optim 管理データ・ストアを参照する場合、ローカル Optim 管理データ・ストアの名前を入力する必要があります。サービス・プロパティを確認して、Optim サポートで使用するためにプロパティを保存できます。

「終了」をクリックしたら、コンソール・ビューを使用してサービスの進行をモニターします。

Optim リポジトリでの作業

Optim リポジトリを使用して、Optim 管理コンソールから実行可能なサービス要求を保管します。

サービス要求を定義した後に、その要求を Optim リポジトリに公開する必要があります。管理コンソールはそこでサービス要求を使用できるようになります。

Optim レジストリーは、リポジトリに保管されるサービス要求の索引です。

デフォルトの Optim リポジトリの場所の入力

Optim 設定を使用して、デフォルトの Optim リポジトリの場所を入力できます。デフォルトの場所は、「Optim サービスの公開」ウィザードに表示されます。

デフォルトの Optim リポジトリの場所を入力するには、次のようにします。

1. 「ウィンドウ」→「設定」をクリックします。「設定」ウィンドウが開きます。
2. ナビゲーション・ツリーで、「**Optim**」ノードを拡張してから、「**Optim レジストリー**」をクリックします。
3. 「レジストリーの URL」フィールドで、デフォルト・リポジトリの URL を入力します。
4. 「OK」をクリックします。

サービス要求の公開

「Optim サービスの公開」ウィザードを使用して、サービス要求を Optim リポジトリに追加できます。

Optim 設定を使用して、ウィザードのデフォルトの Optim レジストリーの場所を選択する必要があります。

サービス要求を公開するには、次のようにします。

1. 「サブセット・サービス」フォルダーを拡張して、サービス名を右クリックしてから、「**Optim サービスの公開**」をクリックします。「Optim サービスの公開」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、Optim レジストリーおよび Optim リポジトリの場所を「レジストリーおよびリポジトリの場所の入力」ページに入力する必要があります。レジストリーの場所を入力した後に、「**リポジトリの参照**」をクリックして、リポジトリの場所を自動的に追加します。「**リポジトリの妥当性検査**」をクリックして、リポジトリの場所を妥当性検査します。

サービス要求のバージョン番号も入力する必要があります。新しい番号は前のバージョンより大きくなければなりません。新しい番号が有効かどうかの検証を選択できます。

第 5 章 Optim 相互運用性モデルでの作業

Optim 相互運用性モデル (OIM) を使用して、Optim および Optim for z/OS で要求を処理できます。

アーカイブ要求モデル

アーカイブ要求の相互運用性モデルを使用して、Optim および Optim for z/OS でアーカイブ要求を処理できます。

アーカイブ・プロセスは、1 つ以上の表から関連する行のセットをコピーして、このデータをアーカイブ・ファイル内に保管します。アーカイブ・プロセスは、アーカイブ要求で開始します。そこでは、ソース表のデータをアーカイブおよび (必要であれば) 削除するためのパラメーター、およびそのデータをアーカイブ・ファイルに保存するためのパラメーターが定義されています。アーカイブ要求は、アクセス定義を参照して、アーカイブするデータ、およびアーカイブ・プロセスの実行に必要なパラメーターを定義します。

アーカイブ・プロセスの詳細については、「*IBM Optim Archive User Manual*」を参照してください。

アーカイブ要求の作成

「新規アーカイブ要求モデル」ウィザードを使用して、Optim デザイナーでアーカイブ要求を作成できます。

アーカイブ要求を作成する前に、Optim プロパティーがある論理データ・モデルを作成する必要があります。

アーカイブ要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「アーカイブ要求」をクリックします。「新規アーカイブ要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、アーカイブ要求の名前を指定して、データ設計プロジェクトと、アーカイブ・ファイルの作成に使用される論理データ・モデルとデータ・アクセス計画を選択する必要があります。

Optim が要求を実行するプラットフォームを指定し、Optim for z/OS 要求については、要求で使用するアクセス定義の名前を入力する必要があります。その後、アーカイブ・ファイルの名前を入力し、アーカイブ・プロセスのオプションを指定して、アーカイブするオブジェクトを選択します。

削除要求モデル

削除要求の相互運用性モデルを使用して、Optim および Optim for z/OS で削除要求を処理できます。

削除プロセスは、抽出またはアーカイブ・プロセスの後に、関連するデータのセットをデータベースから除去します。削除プロセスは、削除要求によって開始します。ここでは、削除対象のデータが入ったソース・ファイルとして抽出またはアーカイブ・ファイルが示され、削除プロセスのためのパラメーターが指定されています。

削除プロセスの詳細については、「*IBM Optim Archive User Manual*」を参照してください。

削除要求の作成

「新規削除要求モデル」ウィザードを使用して、Optim デザイナーで削除要求を作成できます。

削除要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「削除要求」をクリックします。「新規削除要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、削除要求の名前を指定して、ソースのアーカイブまたは抽出ファイルを含むデータ設計プロジェクトを選択する必要があります。

Optim が要求を実行するプラットフォームを指定し、Optim 要求については、削除要求の名前を入力する必要があります。その後、アーカイブまたは抽出のソースおよび制御ファイルの名前を指定し、削除プロセスのオプションおよび限度を指定します。

抽出要求モデル

抽出要求の相互運用性モデルを使用して、Optim および Optim for z/OS で抽出要求を処理できます。

抽出プロセスは、1 つ以上の表から関連する行のセットをコピーして、このデータを抽出ファイル内に保管します。抽出プロセスには、表および列の定義が常に含まれます。主キー、関係、および索引を含むオブジェクト定義を抽出するように選択することもできます。抽出プロセスは、抽出要求で開始します。そこでは、抽出するデータ、および抽出プロセスの実行に必要なパラメーターを定義するための、アクセス定義が指定されています。

抽出プロセスの詳細については、「*IBM Optim Move User Manual*」を参照してください。

抽出要求の作成

「新規抽出要求モデル」ウィザードを使用して、Optim デザイナーで抽出要求を作成できます。

抽出要求を作成する前に、Optim デザイナーで、Optim プロパティがある論理データ・モデルを作成する必要があります。

抽出要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「抽出要求」をクリックします。「新規抽出要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、抽出要求の名前を指定して、データ設計プロジェクトと、ソース・データが入っている論理データ・モデルの両方を選択する必要があります。論理データ・モデルからデータ・アクセス計画を選択する必要もあります。

Optim が要求を実行するプラットフォームを指定し、Optim for z/OS 要求については、要求で使用するアクセス定義の名前を入力する必要があります。その後、抽出ファイルの名前を入力し、抽出プロセスのオプションを指定して、抽出するオブジェクトを選択します。

挿入要求モデル

挿入要求の相互運用性モデルを使用して、Optim および Optim for z/OS で挿入要求を処理できます。

挿入プロセスは、データをソース・ファイルから指定の宛先表にコピーします。挿入プロセスは、挿入要求で開始します。そこでは、挿入または更新するデータ、およびプロセスの実行に必要なパラメーターを格納する、ソース・ファイルが指定されています。

挿入プロセスの詳細については、「*IBM Optim Archive User Manual*」または「*IBM Optim Move User Manual*」を参照してください。

挿入要求の作成

「新規挿入要求モデル」ウィザードを使用して、Optim デザイナーで挿入要求を作成できます。

挿入要求を作成する前に、Optim プロパティがある論理データ・モデルを作成する必要があります。

挿入要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「挿入要求」をクリックします。「新規挿入要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、挿入要求の名前を指定して、データ設計プロジェクトと、ソース抽出ファイルの作成に使用される論理データ・モデルとデータ・アクセス計画を選択する必要があります。

Optim が要求を実行するプラットフォームを指定し、Optim for z/OS 要求については、挿入要求で使用する表マップの名前を入力する必要があります。その後、ソースおよび制御ファイルの名前を指定し、挿入プロセスのオプションおよび限度を指定します。

ロード要求モデル

ロード要求の相互運用性モデルを使用して、Optim および Optim for z/OS でロード要求を処理できます。

ロード・プロセスは、ソース・ファイル (抽出ファイルまたはアーカイブ・ファイルのいずれか) の内容を、サポートされるデータベースのロード・ユーティリティ形式に変換します。ロード・プロセスは、ロード要求で開始します。そこでは、ロードするデータとその他の処理パラメーターを格納するソース・ファイルが指定されています。

ロード・プロセスの詳細については、「*IBM Optim Archive User Manual*」または「*IBM Optim Move User Manual*」を参照してください。

ロード要求の作成

「新規ロード要求モデル」ウィザードを使用して、Optim デザイナーでロード要求を作成します。

ロード要求を作成する前に、Optim プロパティがある論理データ・モデルを作成する必要があります。

ロード要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「ロード要求」をクリックします。「新規ロード要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、ロード要求の名前を指定して、データ設計プロジェクトと、ソース・データが入っている論理データ・モデルの両方を選択する必要があります。Optim for z/OS 要求については、ロード要求で使用する表マップの名前を入力する必要があります。論理データ・モデルからデータ・アクセス計画を選択する必要があります。

Optim が要求を実行するプラットフォームを指定してから、ロード要求の名前を入力する必要があります。その後、ソースおよび制御ファイルの名前を指定し、該当するデータベースに合わせてロード・ユーザーリティーのプロパティーを入力します。

復元要求モデル

復元要求の相互運用性モデルを使用して、Optim および Optim for z/OS で復元要求を処理できます。

復元プロセスは、1 つ以上のアーカイブ・ファイルからデータを選択して、そのデータを元のデータベースまたは異なるデータベースに復元します。復元プロセスは、復元要求で開始します。そこでは、アーカイブ・ファイルが指定されていて、アーカイブ・データの復元に使用される挿入またはロード要求が定義されています。

復元プロセスの詳細については、「*IBM Optim Archive User Manual*」を参照してください。

復元要求の作成

「新規復元要求モデル」ウィザードを使用して、Optim デザイナーで復元要求を作成します。

復元要求を作成する前に、Optim プロパティーがある論理データ・モデルを作成する必要があります。

復元要求を作成するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを右クリックして、「新規」→「復元要求」をクリックします。「新規復元要求モデル」ウィザードが開きます。
2. ウィザードのステップを完了します。

ウィザードを完了するには、復元要求の名前を指定して、データ設計プロジェクトと、ソース・アーカイブ・ファイルの作成に使用される論理データ・モデルとデータ・アクセス計画を選択する必要があります。

Optim が要求を実行するプラットフォームを指定し、Optim 要求については、復元要求の名前を入力する必要があります。その後、アーカイブ・ファイルの名前を入力し、必要であれば選択復元のプロパティーを指定して、Optim for z/OS 要求については、要求で使用する表マップの名前を入力します。その後、ソースおよび制御ファイルの名前を指定し、挿入プロセス（または、Optim 要求の場合はロード・プロセス）のオプションおよび限度を指定します。

Optim for z/OS 要求モデルでの作業

デザイナーを使用して、Optim for z/OS 要求を定義して実行できます。

Optim 設定の「z/OS バッチ・ホストの編集」ウィンドウを使用して、Optim for z/OS バッチ・ホストへの接続を構成できます。

z/OS バッチ・ホストの構成

Optim 設定を使用して、接続情報およびジョブ情報を入力することにより、Optim for z/OS バッチ・ホストへの接続を構成できます。

構成情報を使用して、Optim デザイナーから実行される Optim 要求用のバッチ JCL が生成されます。

Optim for z/OS バッチ・ホストを構成するには、次のようにします。

1. 「ウィンドウ」 → 「設定」をクリックします。
2. ナビゲーション・ツリーで、「Optim」ノードを拡張してから、「z/OS バッチ・ホスト」をクリックします。z/OS バッチ・ホスト・エディターが開きます。
3. 「追加 (Add)」をクリックします。「z/OS バッチ・ホストの編集」ウィンドウが開きます。
4. 「ジョブのデフォルト」および「DB2 のデフォルト」の情報を入力します。
5. 「OK」をクリックします。z/OS バッチ・ホスト・エディターにバッチ・ホストのプロファイルが表示されます。

z/OS バッチ・ホストの編集

「z/OS バッチ・ホストの編集」ウィンドウを使用して、接続情報およびジョブ情報を入力することにより、Optim for z/OS バッチ・ホストへの接続を構成します。

このウィンドウに入力した情報を使用して、Optim デザイナーから実行される Optim 要求用のバッチ JCL が生成されます。

ジョブのデフォルト

ホスト名

Optim for z/OS ホストのマシン名。

ユーザー名

ホスト・マシンで要求を実行するユーザーの ID。

パスワード

ユーザー ID のパスワード。

ジョブ名

バッチ・ジョブ名。

アカウントティング情報

ユーザー・アカウント番号。

プログラマーの名前

バッチ・ジョブに関連付けられたプログラマーの名前。

ジョブ・クラス

バッチ・ジョブのジョブ・クラス。

メッセージ・クラス

バッチ・ジョブのメッセージ・クラス。

メッセージ・レベル

バッチ・ジョブ出力のメッセージ・レベル。

通知 通知メッセージを受け取るユーザーの ID。

サイト・オプション・ライブラリー

サイト・オプションを含んだライブラリーの場所。ご自分のサイトでの場所については、 Optim または DB2 管理者にお問い合わせください。

DB2 のデフォルト

サブシステム

現在の DB2 サブシステム。

プラン名

DB2 プラン名。

SQL ID

現在の SQLID。

ステップ・ライブラリー

ステップ・ライブラリーの場所。ご自分のサイトでの場所については、 Optim または DB2 管理者にお問い合わせください。

Optim for z/OS 要求の実行

Optim デザイナーから Optim for z/OS 要求を実行できます。

Optim for z/OS 要求を実行する前に、Optim 設定で Optim for z/OS バッチ・ホストをセットアップする必要があります。

Optim for z/OS 要求を実行するには、次のようにします。

1. Data Project Explorer で、「相互運用性モデル」フォルダーを拡張します。
2. 要求を右クリックして、「**Optim 要求の実行**」を選択します。「Optim 要求の実行」ウィザードが開きます。
3. ウィザードのステップを完了します。

ウィザードを完了するには、Optim for z/OS バッチ・ホストを選択して、ジョブ要求の JCL を確認します。

「終了」をクリックしたら、ジョブ・ビューを使用して要求の進行をモニターします。ジョブ・ビューには、すべての Optim for z/OS ジョブ、各ジョブの状況、および各ジョブに関連したスプール・ファイルがリストされます。スプール・ファイルをダブルクリックすると、内容が表示されます。

Linux、UNIX、および Windows での Optim 要求モデルの作業

Optim デザイナーを使用して、Linux、UNIX、および Windows での Optim 要求を定義して実行できます。

Optim デザイナーから Optim 要求を実行するには、次のようにします。

- Optim デザイナー・マシンに Optim インストール環境が含まれている必要があります。
- Optim 設定を使用して、Optim インストール環境の Optim pr0cmnd ユーティリティーの場所を定義する必要があります。

pr0cmnd ユーティリティーで実行される Optim 要求はすべて、Optim インストール環境に関連したデフォルトの Optim ディレクトリーに関連付けられます。

Optim ディレクトリーへの接続を定義して、そのディレクトリーを Optimディレクトリー・プロジェクトに含めることもできます。プロジェクトを使用することで、Optim 定義のインポートとエクスポートが行えます。

pr0cmnd の場所の定義

Optim 設定を使用して、Optim インストール環境の pr0cmnd ユーティリティーの場所を定義できます。

Optim が Optim デザイナー・マシンにインストールされている必要があります。

pr0cmnd の場所を定義するには、次のようにします。

1. 「ウィンドウ」 → 「設定」をクリックします。「設定」ウィンドウが開きます。
2. ナビゲーション・ツリーで、「Optim」ノードを拡張してから、「Optim pr0cmnd」をクリックします。Optim pr0cmnd エディターが開きます。
3. 「pr0cmnd の場所 (pr0cmnd location)」フィールドに pr0cmnd.exe ファイルのパスを入力するか、「参照」をクリックしてパスを選択します。デフォルトの場所は、C:\Program Files\IBM Optim\RT\BIN\pr0cmnd.exe です。
4. 「OK」をクリックします。

Optim ディレクトリーでの作業

Optim ディレクトリーに接続して、Optim 定義のインポートとエクスポートが行えます。

Optim ディレクトリー接続を使用するには、以下を行う必要があります。

1. データ・ソース・エクスプローラーを使用して、ディレクトリーを含んだデータベースへの接続を定義します。
2. Optim ディレクトリー接続プロファイルを定義します。
3. Optim ディレクトリー・プロジェクトを作成します。

Optim ディレクトリー・プロジェクトを使用することで、Optim ディレクトリー内の定義を表示できます。

Optim ディレクトリー接続プロファイルの定義

Optim 設定を使用して、Optim ディレクトリー接続プロファイルを定義できます。

Optim ディレクトリー接続プロファイルを定義する前に、データ・ソース・エクスプローラーを使用して、ディレクトリーを含んだデータベースへの接続を定義する必要があります。

Optim ディレクトリー接続プロファイルを定義するには、次のようにします。

1. 「ウィンドウ」 → 「設定」をクリックします。
2. ナビゲーション・ツリーで、「Optim」ノードを拡張してから、「Optim ディレクトリー」をクリックします。Optim ディレクトリー・エディターが開きます。
3. 「追加 (Add)」をクリックします。「Optim ディレクトリー接続プロファイルの追加 (Add an Optim Directory Connection Profile)」ウィンドウが開きます。
4. Optim ディレクトリーを含むデータベースへの接続を選択します。
5. Optim ディレクトリー・スキーマを選択します。
6. 「追加 (Add)」をクリックします。Optim ディレクトリー・エディターに接続プロファイルが表示されます。

Optim ディレクトリー・プロジェクトの作成

「新規 Optim ディレクトリー・プロジェクト (New Optim Directory Project)」ダイアログを使用して、Optim ディレクトリー・プロジェクトを作成できます。

Optim ディレクトリー・プロジェクトを作成する前に、Optim の設定を使用して Optim ディレクトリー接続プロファイルを定義する必要があります。

Optim ディレクトリー・プロジェクトを作成するには、次のようにします。

1. 「ファイル」→「新規」→「**Optim ディレクトリー・プロジェクト (Optim Directory Project)**」をクリックします。「新規 Optim ディレクトリー・プロジェクト (New Optim Directory Project)」ダイアログが開きます。
2. プロファイル名を入力します。
3. Optim ディレクトリー接続プロファイルを選択します。
4. 「終了」をクリックします。新規の Optim ディレクトリー・プロジェクトが Data Project Explorer に表示されます。

Optim 定義のエクスポート

Optim 定義を、Optim 相互運用性モデル (OIM) または Optim ディレクトリーからエクスポートできます。

Optim 定義を、1 つ以上の OIM から、または Optim ディレクトリーから、Optim エクスポート・ファイル (OEF) にエクスポートできます。さらに、Optim 定義を、1 つ以上の OIM から Optim ディレクトリーにエクスポートすることもできます。

Optim 相互運用性モデルから Optim エクスポート・ファイルへの定義のエクスポート

「エクスポート」ウィザードを使用して、1 つ以上の Optim 相互運用性モデル (OIM) から Optim エクスポート・ファイル (OEF) に定義をエクスポートします。

1 つ以上の OIM から OEF に定義をエクスポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「相互運用性モデル」フォルダーを拡張します。
2. 1 つ以上の OIM を右クリックして、「エクスポート」をクリックします。「エクスポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「相互運用性モデル」フォルダーを拡張して、「**Optim エクスポート・ファイル**」をエクスポート先として選択します。

「定義選択」ページで、エクスポートする 1 つ以上の定義を選択して、出力 OEF ファイル名を入力します。

Optim 相互運用性モデルから Optim ディレクトリーへの定義のエクスポート

「エクスポート」ウィザードを使用して、Optim 相互運用性モデル (OIM) から Optim ディレクトリーに定義をエクスポートします。

1 つ以上の OIM から Optim ディレクトリーに定義をエクスポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「相互運用性モデル」フォルダーを拡張します。

2. 1 つ以上の OIM を右クリックして、「**エクスポート**」をクリックします。「エクスポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「**相互運用性モデル**」フォルダーを拡張して、「**Optim ディレクトリー**」をエクスポート先として選択します。

「定義選択」ページで、エクスポートする 1 つ以上の定義を選択します。

Optim ディレクトリーから Optim エクスポート・ファイルへの定義のエクスポート

「エクスポート」ウィザードを使用して、Optim ディレクトリーから Optim エクスポート・ファイル (OEF) に、定義または定義のグループをエクスポートします。

Optim ディレクトリーから OEF に定義または定義のグループをエクスポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「**Optim ディレクトリー**」フォルダーを拡張します。
2. 定義または定義のグループを右クリックして、「**エクスポート**」をクリックします。「エクスポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「**Optim ディレクトリー**」フォルダーを拡張して、「**Optim エクスポート・ファイル**」をエクスポート先として選択します。

「Optim ディレクトリー」ページでプロジェクト・フォルダーを拡張して、エクスポートする定義または定義のグループを選択してから、出力 OEF ファイル名を入力します。

Optim 定義のインポート

Optim 定義を、Optim 相互運用性モデル (OIM) または Optim ディレクトリーにインポートできます。

Optim 定義を、Optim エクスポート・ファイル (OEF) から、OIM または Optim ディレクトリーにインポートできます。

また、Optim 定義を Optim ディレクトリーから OIM にインポートすることもできます。例えば、アクセス定義を抽出要求モデルにインポートできます。

Optim エクスポート・ファイル定義の Optim 相互運用性モデルへのインポート

「インポート」ウィザードを使用して、Optim エクスポート・ファイル (OEF) から Optim 相互運用性モデル (OIM) にすべての定義をインポートします。

OEF から OIM にすべての定義をインポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「**相互運用性モデル**」フォルダーを拡張します。
2. OIM を右クリックして、「**インポート**」をクリックします。「インポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「**Optim 相互運用性モデル**」フォルダーを拡張して、「**Optim エクスポート・ファイル**」をインポートのソースとして選択します。

「Optim 相互運用性モデルの選択」ページで、OEF 定義のインポート先となる OIM を選択するかまたは OIM ファイル名を入力します。

Optim エクスポート・ファイル定義の Optim ディレクトリーへのインポート

「インポート」ウィザードを使用して、Optim エクスポート・ファイル (OEF) から Optim ディレクトリーにすべての定義をインポートします。

OEF から Optim ディレクトリーにすべての定義をインポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「**Optim ディレクトリー**」フォルダーを拡張します。
2. Optim 定義を右クリックして、「**インポート**」をクリックします。「インポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「**Optim ディレクトリー**」フォルダーを拡張して、「**Optim エクスポート・ファイル**」をインポートのソースとして選択します。

「Optim ディレクトリー」ページで、Optim ディレクトリー・プロジェクトを選択します。

「Optim エクスポート・ファイル」ページで、すべての定義のインポート元となる OEF を選択して、入力ファイル名を入力します。

Optim ディレクトリー定義の Optim 相互運用性モデルへのインポート

「インポート」ウィザードを使用して、Optim ディレクトリーの定義または定義のグループを Optim 相互運用性モデル (OIM) にインポートします。

Optim ディレクトリー定義または定義のグループを OIM にインポートするには、次のようにします。

1. Data Project Explorer で、プロジェクトの「**Optim ディレクトリー**」フォルダーを拡張します。
2. OIM を右クリックして、「**インポート**」をクリックします。「インポート」ウィザードが開きます。
3. ウィザードのステップを完了します。

「選択」ページで、「**Optim 相互運用性モデル**」フォルダーを拡張して、「**Optim ディレクトリー**」をインポートのソースとして選択します。

「Optim 相互運用性モデルの選択」ページで、Optim ディレクトリーの定義または定義のグループのインポート先となる OIM をプロジェクトから選択するか、または OIM ファイル名を入力します。

「Optim ディレクトリー」ページで、OIM にインポートする定義または定義のグループを選択します。

Optim 要求の実行

Optim デザイナーから Optim 要求を実行できます。

Optim 要求を実行する前に、Optim 設定で Optim pr0cmd の場所を定義する必要があります。

Optim 要求を実行するには、次のようにします。

1. Data Project Explorer で、「**相互運用性モデル**」フォルダーを拡張します。
2. 要求を右クリックして、「**Optim 要求の実行**」を選択します。「Optim 要求の実行」ウィザードが開きます。
3. ウィザードのステップを完了します。
 - 要求は、Optim デザイナー・マシンの Optim インストール環境に関連したデフォルトの Optim ディレクトリーにインポートされます。インポート要求は編集が可能です。
 - 要求を処理する Optim サーバーの名前を入力できます。

- 要求の実行に使用するコマンド行構文を編集できます。

「終了」をクリックしたら、コンソール・ビューを使用して要求の進行をモニターします。

第 6 章 データ・プライバシー・ポリシーの使用

データ・プライバシー・ポリシーを使用することにより、Optim 相互運用性モデルまたはデータ管理サービスで処理されるデータをマスクすることができます。プライバシー・ポリシーを使ったデータのマスクには、ルックアップ、ルール・ベース、および JavaScript の 3 つのオプションがあります。ルックアップ・オプションでは、ルックアップ表を使用して、マスクされたデータを作成します。ルール・ベース・オプションでは、マスクされるデータを関数を使用して生成します。JavaScript オプションは、JavaScript を使用してデータ変換を定義し、データ管理サービスでの使用のみ可能です。

ルックアップおよびルール・ベースのオプションは、論理データ・モデルのエントティに適用されます。ルックアップまたはルール・ベースのオプションを使用するポリシーをエンティティに適用するとき、エンティティに対してデータ管理サービスを実行して、エンティティ内のデータを変換する必要があります。データ・アクセス計画を使用することで、ルックアップまたはルール・ベースのオプションを使用するポリシーを適用します。データ・アクセス計画にデータ・プライバシー・ポリシーを作成するため、「ポリシーの追加」ウィザードを使用します。

JavaScript オプションは JavaScript ポリシーに含まれ、これにより、ルックアップおよびルール・ベースのオプションとは異なる固有のサービスのデータ変換を実行します。JavaScript ポリシーは、データ管理サービスのエンティティに適用されます。ポリシーは、このサービスに定義され、サービスの実行時に実行されます。サービス計画を使用することで、JavaScript ポリシーをサービスに追加します。JavaScript ポリシーで定義された変換は、Optim がソース・データに対してルックアップまたはルール・ベースの変換を実行した後に行われます。

以下を行うことができます。

- ルックアップ関数を使用して、選択されたソース・エンティティの値を、対応するルックアップ表の列の値に置き換えます。
- ルール・ベース関数を使用して、国民 ID 番号、クレジット・カード番号、および E メール・アドレスを、有効な固有値でマスクします。
- ルール・ベース関数を使用して、日付、文字、および番号の値を生成します。
- 「切り替え」値に基づいて、ルックアップ関数またはルール・ベース関数を適用します。
- JavaScript を使用して、データ管理サービスにカスタム変換を定義します。

日付マスク・ポリシー

日付マスク・ポリシーを使用して、日付をマスクします。このポリシーには、ルール・ベースと JavaScript のオプションが含まれます。

範囲内のランダム日付

「範囲内のランダム日付」ポリシーは、指定された日付範囲内でランダムに選択した日付を生成します。マスクされる日付として、いくつかの形式を使用できます。

JavaScript ポリシーの構文

このポリシーは、関数 `DateMask.randomDateInRange()` で次のようにして使用できます。

```
DateMask.randomDateInRange(<arg:startDate>, <arg:endDate>, <arg:dateFormat>)
```

例えば、1999 年 1 月 1 日から 2009 年 1 月 1 日までの日付を、MM-dd-yyyy の形式でランダムに生成するには、以下を使用します。

```
DateMask.randomDateInRange('1999-01-01', '2009-01-01', 'MM-dd-yyyy')
```

引数	説明
endDate	日付範囲の終了日 (yyyy-MM-dd 形式)。
startDate	日付範囲の開始日 (yyyy-MM-dd 形式)。
dateFormat	マスクされる日付の形式。デフォルトは yyyy-MM-dd です。以下の形式がサポートされます。 <ul style="list-style-type: none">• dd-MM-yyyy• dd-MM-yyyy HH:mm:ss• MM-dd-yyyy• MM-dd-yyyy HH:mm:ss• MMM dd, yyyy• yyyy-MM-dd

「範囲内のランダム日付」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「範囲内のランダム日付」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「範囲内のランダム日付」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - ランダム値または入力値に基づく値のいずれかを生成できます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスク番号を生成する基礎となるシード番号を指定できます。

日付を月に丸める

「日付を月に丸める」ポリシーは、日付を元の月の最初の日付に丸めることによって日付をマスクします。例えば、8 月 21 日は 8 月 1 日に丸められます。生成される日付の形式は入力日付と一致します。

JavaScript ポリシーの構文

このポリシーは、関数 `DateMask.roundDateToMonth()` で次のようにして使用できます。

```
DateMask.roundDateToMonth(record.getItem(<arg:inputAttribute>), <arg:dateFormat>)
```

例えば、日付を丸めて MM-dd-yyyy という形式で月の最初の日にするには、次のように入力します。

```
DateMask.roundDateToMonth(record.getItem('/DEMO/ORDERS/ORDER_DATE'), 'MM-dd-yyyy')
```

引数	説明
inputAttribute	マスクする日付を含んだソース属性。
dateFormat	日付の形式。デフォルトは yyyy-MM-dd です。以下の形式がサポートされます。 <ul style="list-style-type: none">• dd-MM-yyyy• dd-MM-yyyy HH:mm:ss• MM-dd-yyyy• MM-dd-yyyy HH:mm:ss• MMM dd, yyyy• yyyy-MM-dd

「日付を月に丸める」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「日付を月に丸める」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「日付を月に丸める」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスクされる日付の形式を指定することができます。

日付を年に丸める

「日付を年に丸める」ポリシーは、日付を元の年の 1 月 1 日に丸めることによって日付をマスクします。例えば、2008 年 8 月 21 日は、2008 年 1 月 1 日に丸められます。生成される日付の形式は、入力日付と一致します。

JavaScript ポリシーの構文

このポリシーは、関数 `DateMask.roundDateToYear()` で次のようにして使用できます。

```
DateMask.roundDateToYear(record.getItem(<arg:inputAttribute>), <arg:dateFormat>)
```

例えば、日付を丸めて MM-dd-yyyy という形式で年の最初の日にするには、次のように入力します。

```
DateMask.roundDateToYear(record.getItem('/DEMO/ORDERS/ORDER_DATE'), 'MM-dd-yyyy')
```

引数	説明
inputAttribute	マスクする日付を含んだソース属性。
dateFormat	マスクされる日付の形式。デフォルトは yyyy-MM-dd です。以下の形式がサポートされます。 <ul style="list-style-type: none">• dd-MM-yyyy• dd-MM-yyyy HH:mm:ss• MM-dd-yyyy• MM-dd-yyyy HH:mm:ss• MMM dd, yyyy• yyyy-MM-dd

「日付を年に丸める」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「日付を年に丸める」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「日付を年に丸める」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスクされる日付の形式を指定することができます。

身元マスク・ポリシー

身元マスク・ポリシーを使用して、E メール・アドレス、クレジット・カード番号、国民 ID 番号などの個人情報をマスクします。このポリシーには、ルール・ベースと JavaScript のオプションが含まれます。

E メール・アドレス・ポリシー

E メール・アドレス・ポリシーは、E メール・アドレスをマスクするために使用します。「自動生成される E メール名」および「フォーマット設定された E メール名」の、2 つのポリシーを使用できます。

自動生成される E メール名

「自動生成される E メール名」ポリシーは、連番が連結されたりテラルに基づいたユーザー名で E メール・アドレスを生成します。連番は 1 で始まり 1 ずつ増える接尾部です。このポリシーは、指定したソース値列にある E メール・アドレスのドメイン・ネームを使用します。

自動生成ユーザー名ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に自動生成ユーザー名ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

自動生成ユーザー名ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。 ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - ユーザー名としてリテラルを入力する必要があります。
 - E メール・アドレスを大文字または小文字に変換できます。

フォーマット設定された E メール名

「フォーマット設定された E メール名」ポリシーは、1 つまたは 2 つの属性から取得した値に基づいたユーザー名で E メール・アドレスを生成します。このポリシーは、指定されたソース値列にある E メール・アドレスからのドメイン・ネームを使用します。

「フォーマット設定された E メール名」ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「フォーマット設定された E メール名」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「フォーマット設定された E メール名」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。 ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - ユーザー名を提供するためのソース属性を 2 つまで選択する必要があります。

- ユーザー名の最初の部分を指定する属性から先頭文字だけを使用して、ユーザー名の 2 つの部分の間に区切り記号を含めるように選択することができます。
- E メール・アドレスを大文字または小文字に変換できます。

クレジット・カードのポリシー

クレジット・カード識別ポリシーは、発行者 (American Express、Diners Club、Discover、JCB、MasterCard、VISA) のクレジット・カード番号 (CCN) をマスクします。各ポリシーは、アカウント番号およびチェック・ディジットをマスクします。

ISO 7812 で定義された CCN の構成では、まず 6 桁の発行者 ID があり、その後に可変長のアカウント番号が続き、最後にチェック・ディジットとして 1 桁の数字があります。チェック・ディジットは CCN の正確性を検査するためのもので、発行者 ID とアカウント番号を Luhn アルゴリズムに通すことによって生成されます。CCN の最大長は 19 桁です。

クレジット・カードに固有のポリシー

クレジット・カードのポリシーには、サポートされる各クレジット・カード発行者のポリシーが含まれます。クレジット・カードに固有のポリシーは、指定したクレジット・カード発行者と一致する番号のみをマスクします。

以下のクレジット・カードに固有のポリシーを使用できます。

- American Express のクレジット・カード番号のマスク
- Discover のクレジット・カード番号のマスク
- Diners Club のクレジット・カード番号のマスク
- JCB のクレジット・カード番号のマスク
- MasterCard のクレジット・カード番号のマスク
- VISA のクレジット・カード番号のマスク

クレジット・カード固有のポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画にクレジット・カード固有のポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

クレジット・カード固有のポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - ランダム値または入力値に基づく値のいずれかを生成できます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。

- マスク番号を生成する基礎となるシード番号を指定できます。
- 発行者番号をマスクするように選択することができます。

すべての提供者からのクレジット・カード番号をマスクする

「すべての提供者からのクレジット・カード番号をマスクする」ポリシーは、CCNの発行者を判別し、発行者のフォーマットに従って番号をマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `CCNMask.randomCCN()` および `CCNMask.maskCCN()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。 `CCNMask.randomCCN()`

入力値に基づいて値を生成するには、次の関数を使用します。 `CCNMask.maskCCN(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
CCNMask.maskCCN(record.getItem('/DEMO/ORDERS/CCN'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。 `CCNMask.maskCCN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

例えば、CCN 属性に基づいて値を生成し、CUST_INFO 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。 `CCNMask.maskCCN(record.getItem('/DEMO/CUSTOMERS/CCN'), '/DEMO/CUSTOMERS/CUST_INFO')`

引数

説明

`inputAttribute`

マスクする入力値を含んだ属性。

`additionalAttribute`

入力値のすべてのインスタンスがマスクされる追加属性。

「すべての提供者からのクレジット・カード番号をマスクする」ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に、「すべての提供者からのクレジット・カード番号をマスクする」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「すべての提供者からのクレジット・カード番号をマスクする」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。

- ランダム値または入力値に基づく値のいずれかを生成できます。
- マスク番号を生成する基礎となるシード番号を指定できます。
- 発行者番号をマスクするように選択することができます。

提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする

「提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする」ポリシーは、切り替えオプションを使用して、選択された切り替え属性の値に基づいて CCN をマスクします。例えば、行の切り替え属性に値「VISA」が含まれる場合、ポリシーは行の VISA クレジット・カード番号をマスクします。

切り替えオプションは、American Express、Diners Club、Discover、JCB、MasterCard、および VISA という値に基づいています。

「提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする」ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に、「提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「**開く**」をクリックします。ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - ランダム値または入力値に基づく値のいずれかを生成できます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 切り替え値を提供するためのソース属性を選択する必要があります。
 - マスク番号を生成する基礎となるシード番号を指定できます。
 - 発行者番号をマスクするように選択することができます。

国民 ID のポリシー

国民 ID ポリシーを使用して、国民 ID 番号をマスクします。

各国特有の国民 ID のポリシー

各国特有の国民 ID のポリシーは、特定の国民識別番号をマスクします。

各国特有の国民 ID ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に各国特有の国民 ID ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

各国特有の国民 ID ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - ランダム値または入力値に基づく値のいずれかを生成できます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスク番号を生成する基礎となるシード番号を指定できます。

カナダ社会保険番号のマスク:

「カナダ社会保険番号のマスク」ポリシーは、ソース値の最初の 3 桁を含んだランダムなカナダ社会保険番号 (SIN) を生成します。

JavaScript ポリシーの構文

このポリシーは、関数 `SINMask.randomSIN()` および `SINMask.maskSIN()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。 `SINMask.randomSIN()`

入力値に基づいて値を生成するには、次の関数を使用します。 `SINMask.maskSIN(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
SINMask.maskSIN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。 `SINMask.maskSIN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

例えば、`NATIONAL_ID` 属性に基づいて値を生成し、`CUST_ID` 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
SINMask.maskSIN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数

`inputAttribute`

`additionalAttribute`

説明

マスクする入力値を含んだ属性。

入力値のすべてのインスタンスがマスクされる追加属性。

フランスの National Institute for Statistics and Economic Studies Number のマスク:

「フランスの National Institute for Statistics and Economic Studies Number のマスク」ポリシーは、行政区分番号を表す 2 桁とソース値の制御キーを表す 2 桁を含んだ、ランダムなフランスの National Institute for Statistics and Economic Studies Number (INSEE) を生成します。

JavaScript ポリシーの構文

このポリシーは、関数 `INSEEMask.randomINSEE()` および `INSEEMask.maskINSEE()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。

```
INSEEMask.randomINSEE()
```

入力値に基づいて値を生成するには、次の関数を使用します。 `INSEEMask.maskINSEE(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
INSEEMask.maskINSEE(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。 `INSEEMask.maskINSEE(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

例えば、`NATIONAL_ID` 属性に基づいて値を生成し、`CUST_ID` 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
INSEEMask.maskINSEE(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数

説明

`inputAttribute`

マスクする入力値を含んだ属性。

`additionalAttribute`

入力値のすべてのインスタンスがマスクされる追加属性。

イタリアの Fiscal Code Number のマスク:

「イタリアの Fiscal Code Number のマスク」ポリシーは、ソース値の最初の 6 桁 (ファーストネームおよびラストネームを表す) を含んだ、ランダムなイタリアの Fiscal Code Number (CF) を生成します。

JavaScript ポリシーの構文

このポリシーは、関数 `CFMask.randomCF()` および `CFMask.maskCF()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。 `CFMask.randomCF()`

入力値に基づいて値を生成するには、次の関数を使用します。 `CFMask.maskCF(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
CFMask.maskCF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。 `CFMask.maskCF(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

例えば、`NATIONAL_ID` 属性に基づいて値を生成し、`CUST_ID` 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
CFMask.maskCF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数

説明

`inputAttribute`

マスクする入力値を含んだ属性。

引数	説明
additionalAttribute	入力値のすべてのインスタンスがマスクされる追加属性。

スペインの Fiscal Identification Number のマスク:

「スペインの Fiscal Identification Number のマスク」ポリシーは、ランダムなスペインの Fiscal Identification Number (NIF) を生成します。非市民であることを示すために使用される「X」接頭部がソース値に含まれる場合は、その接頭部が組み込まれます。

JavaScript ポリシーの構文

このポリシーは、関数 `NIFMask.randomNIF()` および `NIFMask.maskNIF()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。 `NIFMask.randomNIF()`

入力値に基づいて値を生成するには、次の関数を使用します。 `NIFMask.maskNIF(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
NIFMask.maskNIF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。 `NIFMask.maskNIF(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

例えば、`NATIONAL_ID` 属性に基づいて値を生成し、`CUST_ID` 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
NIFMask.maskNIF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数	説明
inputAttribute	マスクする入力値を含んだ属性。
additionalAttribute	入力値のすべてのインスタンスがマスクされる追加属性。

英国の National Insurance Number のマスク:

「英国の National Insurance Number のマスク」ポリシーは、ソース値の最初の 2 文字 (接頭部) と最後のオプション文字 (接尾部) を含んだ、ランダムな英国の National Insurance Number (NINO) を生成します。

JavaScript ポリシーの構文

このポリシーは、関数 `NINOMask.randomNINO()` および `NINOMask.maskNINO()` で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。 `NINOMask.randomNINO()`

入力値に基づいて値を生成するには、次の関数を使用します。 `NINOMask.maskNINO(record.getItem('<arg.inputAttribute>'))`

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
NINOMask.maskNINO(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。NINOMask.maskNINO(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')

例えば、NATIONAL_ID 属性に基づいて値を生成し、CUST_ID 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
NINOMask.maskNINO(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数	説明
inputAttribute	マスクする入力値を含んだ属性。
additionalAttribute	入力値のすべてのインスタンスがマスクされる追加属性。

米国社会保障番号のマスク:

「米国社会保障番号のマスク」ポリシーは、ソースの地域番号を含んだランダムな社会保障番号 (SSN) を生成します。

SSN は 3 つのサブフィールドから成ります。最初の 3 桁 (地域) は、その SSN が発行された州によって一般的に定められた地域を表します。次の 2 桁 (グループ) は、地域番号に対応するグループ番号を定義したものです。最後の 4 桁 (通し番号) は、順次シリアル番号です。このポリシーが生成するマスクされた SSN は、地域番号の該当グループ番号を含んでいます。

JavaScript ポリシーの構文

このポリシーは、関数 SSNMask.randomSSN() および SSNMask.maskSSN() で次のようにして使用できます。

入力値に基づかずにランダム値を生成するには、次の関数を使用します。SSNMask.randomSSN()

入力値に基づいて値を生成するには、次の関数を使用します。SSNMask.maskSSN(record.getItem('<arg.inputAttribute>'))

例えば、入力値に基づいてランダム値を生成するには、以下を使用します。

```
SSNMask.maskSSN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

入力値に基づいて値を生成し、入力値のすべてのインスタンスがマスクされる追加属性をマスクするには、次の関数を使用します。SSNMask.maskSSN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')

例えば、NATIONAL_ID 属性に基づいて値を生成し、CUST_ID 属性にある入力値の追加インスタンスをマスクするには、以下を使用します。

```
SSNMask.maskSSN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

引数	説明
inputAttribute	マスクする入力値を含んだ属性。
additionalAttribute	入力値のすべてのインスタンスがマスクされる追加属性。

国名または国別コードに基づいて国民識別番号をマスクする

「国名または国別コードに基づいて国民識別番号をマスクする」ポリシーは、切り替えオプションを使用して、選択された切り替え属性の値に基づいて国民識別番号をマスクします。例えば、行の切り替え属性に値「USA」が含まれる場合、ポリシーは行の米国の社会保障番号をマスクします。

切り替えオプションは、以下の値に基づいています。

カナダ社会保障番号

CA、CAN、Canada、Canadian、ca、can

スペインの Fiscal Identification Number

ES、España、Spain、Spanish、PQH_ES、SPA、ESP、es、pqh_es、spa、esp

フランスの National Institute for Statistics and Economic Studies Number

FR、France、French、FRE、PQH_FR、FRA、fr、fre、fra、pqh_fr

イタリアの Fiscal Code Number

IT、Italy、Italian、ITA、PQH_IT、it、ita、pqh_it

英国の National Insurance Number

UK、U.K.、United Kingdom、Great Britain、England、Scotland、Wales、Northern Ireland、British、English、Welsh、Scottish、BRI、PQH_GB、WEL、SCO、GBR、GB、G.B.、uk、bri、pqh_gb、wel、sco、gbr、gb

米国社会保障番号

US、U.S.、USA、U.S.A.、American、AM、us、usa、am

「国名または国別コードに基づいて国民識別番号をマスクする」ポリシーの作成:

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に、「国名または国別コードに基づいて国民識別番号をマスクする」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「国名または国別コードに基づいて国民識別番号をマスクする」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - ランダム値または入力値に基づく値のいずれかを生成できます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 切り替え値を提供するためのソース属性を選択する必要があります。
 - マスク番号を生成する基礎となるシード番号を指定できます。

ルックアップ・マスク・ポリシー

ルックアップ・マスク・ポリシーは、選択されたソース・エンティティの値を、対応するルックアップ表列のエンティティの値に置き換えることでソース値をマスクします。

ルックアップ・ポリシーは、Optim サンプル・データの EXTENDED_LOOKUP スキーマのルックアップ表にバインドされます。ルックアップ・ポリシーを作成する際、ポリシーにバインドされた属性にソース・データの属性をマップします。

ランダムおよびハッシュ・ルックアップ

ルックアップ処理には、ランダム・ルックアップおよびハッシュ・ルックアップという 2 つのオプションがあります。プライバシー・ポリシーの作成時に、ルックアップ・オプションを選択できます。

ルックアップ表には、連続した順次値を含んだ列が含まれます。ルックアップ表の順次値と、ランダムまたはハッシュ・オプションを使用して生成された値を突き合わせることで、ルックアップ表の行が選択されます。

ランダム・ルックアップ

ランダム・ルックアップは、ルックアップ表からランダムに行を選択して、置換値を入手します。

ハッシュ・ルックアップ

ハッシュ・ルックアップ処理では、ソース値をハッシュし、ハッシュされた値をルックアップ表の行の索引として使用することにより、置換値が選択されます。ハッシュされるソース列は、ルックアップ表の値に置き換えられる列である必要はありません。ソース列およびルックアップ列の最大長は 256 文字です。ハッシュ関数は大/小文字の区別があります。ソース値がハッシュされる前に大文字に変換することができます。

ハッシュする値を得るために使用するソース列に特定の値 (NULL、CHAR 列の場合のスペース、ゼロ長 VARCHAR) が含まれている場合、その値はハッシュされず、以下の予約値がルックアップ表へのキーとして使用されます。

ソース値	ルックアップ表へのキー
NULL	-1
スペース (CHAR または VARCHAR)	-2
ゼロ長 VARCHAR	-3
値のすべてが以下のうちの 1 つ以上である複数のハッシュ・ルックアップ列: NULL、スペース (CHAR または VARCHAR)、ゼロ長 VARCHAR	-4

切り替えオプション

いくつかのルックアップ・ポリシーでは、選択した切り替え属性の値に基づきデータをマスクするための切り替え値オプションを使用します。切り替えオプションは、国名または国別コードに基づいています。例えば、行の切り替え属性に値「USA」が含まれる場合、ポリシーは米国に固有のデータを使用して、行のデータをマスクします。

切り替えオプションは、以下の値を使用します。

AU - オーストラリア

AU、au、OZ、oz、Australia、australia

CA - カナダ

CA, CAN, Canada, Canadian, ca, can

DE - ドイツ

DE, de, Deutschland, deutschland, GER, ger, Germany, germany, FRG, frg, BRD, brd, Bundesrepublik Deutschland

ES - スペイン

ES, Espana, Spain, Spanish, PQH_ES, SPA, ESP, es, pqh_es, spa, esp

FR - フランス

FR, France, French, FRE, PQH_FR, FRA, fr, fre, fra, pqh_fr

IT - イタリア

IT, Italy, Italian, ITA, PQH_IT, it, ita, pqh_it

JP - 日本

JP, Japan, Japanese, jp, Nippon, Nihon

UK - 英国

UK, U.K., United Kingdom, Great Britain, England, Scotland, Wales, Northern Ireland, British, English, Welsh, Scottish, BRI, PQH_GB, WEL, SCO, GBR, GB, G.B., uk, bri, pqh_gb, wel, sco, gbr, gb

US - 米国

US, U.S., USA, U.S.A., American, AM, us, usa, am

住所情報のルックアップ・ポリシー

住所ポリシーは住所をマスクします。オーストラリア、カナダ、フランス、ドイツ、イタリア、日本、スペイン、英国、米国用があります。ポリシーでは、以下の属性が記述されます。

各ポリシーには国特有のバージョンがあります。例、米国住所情報のマスクなど。

また、選択した切り替え属性の値に基づき住所情報をマスクするための切り替え値オプションを使用するポリシーもあります (国名または国別コードをもとに住所情報をマスクする)。例えば、データ行の切り替え属性に値「USA」が含まれる場合、ポリシーは米国住所情報のマスク・ポリシーを使用して、行のデータをマスクします。

このポリシーは、以下の属性にマップされたデータをマスクします。

属性	説明
Address1	番地の 1 行目。
Address2	番地の 2 行目。
City	市区町村名。
StateOrProvince	州などの行政区分名。
ZipOrPostalCode1	ZIP コードまたは郵便番号の最初の部分。

個人情報のルックアップ・ポリシー

個人情報ポリシーは人の個人データをマスクします。オーストラリア、カナダ、フランス、ドイツ、イタリア、日本、スペイン、英国、米国用があります。

各ポリシーには国特有のバージョンがあります。例、米国個人情報のマスクなど。

また、選択した切り替え属性の値に基づき個人情報をマスクするための切り替え値オプションを使用するポリシーもあります (国名または国別コードをもとに個人情報をマスクする)。例えば、データ行の切り替え属性に値「USA」が含まれる場合、ポリシーは米国個人情報のマスク・ポリシーを使用して、行のデータをマスクします。

このポリシーは、以下の属性にマップされたデータをマスクします。

属性	説明
ID	国民 ID
FirstName	ファーストネーム。
LastName	ラストネーム。
Company	会社名。
Gender	性別。
Phone	電話番号。
BirthDate	生年月日。
EEmailAddress	E メール・アドレス

ファーストネーム情報のルックアップ・ポリシー

ファーストネーム情報ポリシーは人のファーストネームをマスクします。オーストラリア、カナダ、フランス、ドイツ、イタリア、日本、スペイン、英国、米国用があります。それぞれの国ごとに、通性ポリシーと、男性と女性それぞれのポリシーがあります。ポリシーでは、以下の属性が記述されます。

それぞれの国ごとに、男性と女性用のポリシーがあります。例えば、米国女性のファーストネーム情報のマスクと、米国男性のファーストネーム情報のマスクがあります。

各国につき、通性ポリシーも存在します。例、米国のファーストネーム情報のマスクなど。

また、選択した切り替え属性の値に基づきファーストネーム情報をマスクするための切り替え値を使用するポリシーもあります。男性用と女性用の切り替え値ポリシー (国名または国別コードに基づく女性のファーストネームのマスクと、国名または国別コードに基づく男性のファーストネームのマスク) があります。通性ポリシー (国名または国別コードに基づくファーストネームのマスク) もあります。

例えば、データ行の切り替え属性に値「USA」が含まれる場合、ポリシーは米国のファーストネーム情報のマスク・ポリシーを使用して、行のデータをマスクします。

このポリシーは、以下の属性にマップされたデータをマスクします。

属性	説明
FirstName	ファーストネーム。

ラストネーム情報のルックアップ・ポリシー

ラストネーム情報ポリシーは人のラストネームをマスクします。オーストラリア、カナダ、フランス、ドイツ、イタリア、日本、スペイン、英国、米国用があります。

各ポリシーには国特有のバージョンがあります。例、米国のラストネーム情報のマスクなど。

また、選択した切り替え属性の値に基づきラストネーム情報をマスクするための切り替え値オプションを使用するポリシーもあります (国名または国別コードをもとにラストネームをマスクする)。例えば、データ行の切り替え属性に値「USA」が含まれる場合、ポリシーは米国のラストネーム情報のマスク・ポリシーを使用して、行のデータをマスクします。

このポリシーは、以下の属性にマップされたデータをマスクします。

属性	説明
LastName	ラストネーム。

会社名ルックアップ・マスク・ポリシー

会社名マスク・ポリシーは、会社の名前をマスクします。

このポリシーは、以下の属性にマップされたデータをマスクします。

属性	説明
CompanyName	会社名。

ルックアップ・ポリシーの作成

「ポリシーの追加」ウィザードを使用してルックアップ・ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

ルックアップ・ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - ポリシーにバインドされた属性にソース属性をマップする必要があります。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - ランダムまたはハッシュのどちらかのルックアップ・オプションを選択する必要があります。ハッシュ・オプションを選択した場合は、以下を行う必要があります。
 - ハッシュ値の生成に使用する属性を選択します。
 - これら属性の順序を設定します。
 - ハッシュ値の生成時に無視する値を指定します。
 - 切り替えオプションを使用するポリシーを選択する場合は、切り替え値を提供するためのソース属性を選択する必要があります。
 - ルックアップ表の行を選択するための値の生成に使用するシード番号を指定できます。

数値マスク・ポリシー

数値マスク・ポリシーを使用することで、ランダム値を生成して数値データをマスクします。このポリシーには、ルール・ベースと JavaScript のオプションが含まれます。

ガウス・ランダム・ダブル

「ガウス・ランダム・ダブル」ポリシーは、ランダムな倍精度浮動小数点数を生成します。生成される数値は、釣鐘型のガウス曲線に基づいています。

ガウス分布では、一様分布の乱数とは異なり、平均から離れた数よりも平均に近い数が選択される確率が高くなります。1 から 10 までの一様分布の乱数の場合、生成される 1 の数は、生成される 5 や 10 の数とほぼ同じになります。平均が 6 で標準偏差が 2 のガウス分布の場合、3 や 9 よりも 5 や 7 の方が多く生成されます。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.gaussianRandomDouble()` で次のようにして使用できます。

```
ScrambleMask.gaussianRandomDouble(<arg:mean>, <arg:standardDeviation>)
```

例えば、平均が 50.5、標準偏差が 10.00 で値を生成する場合、次のように入力します。

```
ScrambleMask.gaussianRandomDouble('50.5', '10.00')
```

引数	説明
mean	ガウス分布の平均値
standardDeviation	ガウス分布の標準偏差

「ガウス・ランダム・ダブル」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「ガウス・ランダム・ダブル」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「ガウス・ランダム・ダブル」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 釣鐘曲線の中央を設定する平均値、および曲線の幅（平均に対する相対的な範囲で、大部分の値がそこに含まれる）を決定する標準偏差値を指定する必要があります。

ガウス・ランダム整数

「ガウス・ランダム整数」ポリシーは、ランダムな整数を生成します。生成される数値は、釣鐘型のガウス曲線に基づいています。

ガウス分布では、一様分布の乱数とは異なり、平均から離れた数よりも平均に近い数が選択される確率が高くなります。1 から 10 までの一様分布の乱数の場合、生成される 1 の数は、生成される 5 や 10 の数とほぼ同じになります。平均が 6 で標準偏差が 2 のガウス分布の場合、3 や 9 よりも 5 や 7 の方が多く生成されます。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.gaussianRandomInteger()` で次のようにして使用できます。

```
ScrambleMask.gaussianRandomInteger(<arg:mean>, <arg:standardDeviation>)
```

例えば、平均が 100、標準偏差が 20 で値を生成する場合、次のように入力します。

```
ScrambleMask.gaussianRandomInteger('100', '20')
```

引数	説明
mean	ガウス分布の平均値
standardDeviation	ガウス分布の標準偏差

「ガウス・ランダム整数」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「ガウス・ランダム整数」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「ガウス・ランダム整数」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「**データ・アクセス計画**」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「**ポリシーの追加**」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 釣鐘曲線の中央を設定する平均値、および曲線の幅（平均に対する相対的な範囲で、大部分の値がそこに含まれる）を決定する標準偏差値を指定する必要があります。

範囲内一様ランダム・ダブル

「範囲内一様ランダム・ダブル」ポリシーは、指定された範囲内でランダムな倍精度浮動小数点数を生成します。生成される数値は、一様分布に基づいています。

1 から 10 までの一様分布の乱数の場合、生成される 1 の数は、生成される 5 や 10 の数とほぼ同じになります。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.uniformRandomDoubleInRange()` で次のようにして使用できます。

```
ScrambleMask.uniformRandomDoubleInRange(<arg:rangeLowerBound>, <arg:rangeUpperBound>)
```

例えば、.01 から 99.99 までの値を生成するには、次のように入力します。

```
ScrambleMask.uniformRandomDoubleInRange('.01', '99.99')
```

引数	説明
<code>rangeLowerBound</code>	範囲の始まり。生成される最小の値。
<code>rangeUpperBound</code>	範囲の終わり。生成される最大の値。

「範囲内一様ランダム・ダブル」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「範囲内一様ランダム・ダブル」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「範囲内一様ランダム・ダブル」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 範囲の始めと終わりを示すため、倍精度浮動小数点数を指定する必要があります。始めと終わりの数は範囲に含まれます。

範囲内一様ランダム・ロング

「範囲内一様ランダム・ロング」ポリシーは、指定された範囲内でランダムな長整数を生成します。生成される数値は、一様分布に基づいています。

1 から 10 までの一様分布の乱数の場合、生成される 1 の数は、生成される 5 や 10 の数とほぼ同じになります。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.uniformRandomLongInRange()` で次のようにして使用できます。

```
ScrambleMask.uniformRandomLongInRange(<arg:rangeLowerBound>, <arg:rangeUpperBound>)
```

例えば、2000000000 から 3000000000 までの値を生成するには、次のように入力します。


```
ScrambleMask.uniformRandomLongInRange('2000000000', '3000000000')
```

引数

rangeLowerBound)

rangeUpperBound

説明

範囲の始まり。生成される最小の値。

範囲の終わり。生成される最大の値。

「範囲内一様ランダム・ロング」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「範囲内一様ランダム・ロング」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「範囲内一様ランダム・ロング」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - 範囲の始めと終わりを示すため、長整数を指定する必要があります。始めと終わりの数は範囲に含まれます。

スクランブル・マスク・ポリシー

スクランブル・モデル・ポリシーは、文字および数値データ・タイプをマスクするために使用します。このポリシーには、ルール・ベースと JavaScript のオプションが含まれます。

反復可能な置換

「反復可能な置換」ポリシーは、置換される文字の各々のタイプに一致する文字でストリングをマスクします。例えば、数字は数字に、小文字は小文字に置き換えられます。マスクングに使用される文字は指定の文字セットから取得されます。このポリシーは、文字セットの一部となっている文字のみをマスクします。

以下のマスク・メソッドを使用できます。

CRC 巡回冗長検査 (CRC) メソッドは、各ストリングを反復可能な方法でマスクします。ただし、CRC メソッドは各ストリングを固有のストリングでマスクするとは限りません。

ハッシュ

ハッシュ・メソッドは、各ストリングを反復可能な方法でマスクします。ただし、ハッシュ・メソッドは各ストリングを固有のストリングでマスクするとは限りません。

マップ マップ・メソッドは、各ストリングを反復可能な方法で、そして固有なストリングでマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.repeatableReplacement()` で次のようにして使用できます。

```
ScrambleMask.repeatableReplacement(record.getItem('<arg:inputAttribute>'),  
'<arg:scrambleType>','<arg:language>')
```

例えば、CRC メソッドを使用して英語文字セットの文字でストリングをマスクするには、次のように入力します。

```
ScrambleMask.repeatableReplacement(record.getItem('/DEMO/ORDERS/ORDER_ID'),  
'CRC_BASE','English')
```

引数

inputAttribute

scrambleType

language

説明

マスクするストリングを含んだ属性。

CRC_BASE、HASH_BASE、または MAP_BASE のいずれかのマスク・メソッド。

マスキングに使用される文字を提供する、文字セットの言語。 文字セットが指定されていない場合、または文字セットがサポートされていない場合は、英語の文字セットが使用されます。サポートされる文字セットのリストについては、70 ページの『スクランブル・マスク・ポリシーでサポートされる言語文字セット』を参照してください。

「反復可能な置換」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「反復可能な置換」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「反復可能な置換」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。 「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスキングに使用される文字を提供する文字セットを選択する必要があります。
 - 以下のマスク・メソッドのいずれかを選択する必要があります。

CRC 巡回冗長検査 (CRC) メソッドは、各ストリングを反復可能な方法でマスクします。ただし、CRC メソッドは各ストリングを固有のストリングでマスクするとは限りません。

ハッシュ

ハッシュ・メソッドは、各ストリングを反復可能な方法でマスクします。ただし、ハッシュ・メソッドは各ストリングを固有のストリングでマスクするとは限りません。

マップ マップ・メソッドは、各ストリングを反復可能な方法で、そして固有なストリングでマスクします。

正規表現による反復可能な置換

「正規表現による反復可能な置換」ポリシーは、置換される文字の各々のタイプに一致する文字でストリングをマスクします。例えば、数字は数字に、小文字は小文字に置き換えられます。このポリシーは、ストリング内のどの文字をマスクするかを決定するために、正規表現を使用します。マスクングに使用される文字は指定の文字セットから取得されます。このポリシーは、文字セットの一部となっている文字のみをマスクします。

以下のマスク・メソッドを使用できます。

CRC 巡回冗長検査 (CRC) メソッドは、各ストリングを反復可能な方法でマスクします。ただし、CRC メソッドは各ストリングを固有のストリングでマスクするとは限りません。

ハッシュ

ハッシュ・メソッドは、各ストリングを反復可能な方法でマスクします。ただし、ハッシュ・メソッドは各ストリングを固有のストリングでマスクするとは限りません。

マップ マップ・メソッドは、各ストリングを反復可能な方法で、そして固有なストリングでマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.repeatableReplacementByRegularExpression()` で次のようにして使用できます。

```
ScrambleMask.repeatableReplacementByRegularExpression(record.getItem(
  '<arg:inputAttribute>'), '<arg:regularExpression>', '<arg:scrambleType>',
  '<arg:language>')
```

例えば、a から h までの小文字を、CRC メソッドを使用して英語文字セットの文字でマスクするには、次のように入力します。

```
ScrambleMask.repeatableReplacementByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)', 'CRC_BASE', 'English')
```

引数

`inputAttribute`

`regularExpression`

`scrambleType`

`language`

説明

マスクするストリングを含んだ属性。

入力ストリングでマスクする文字を示す正規表現。

CRC_BASE、HASH_BASE、または MAP_BASE のいずれかのマスク・メソッド。

マスクングに使用される文字を提供する、文字セットの言語。文字セットが指定されていない場合、または文字セットがサポートされていない場合は、英語の文字セットが使用されます。サポートされる文字セットのリストについては、70 ページの『スクランブル・マスク・ポリシーでサポートされる言語文字セット』を参照してください。

「正規表現による反復可能な置換」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「正規表現による反復可能な置換」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「正規表現による反復可能な置換」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。 「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスクする文字を指定する正規表現を提供する必要があります。
 - マスキングに使用される文字を提供する文字セットを選択する必要があります。
 - 以下のマスク・メソッドのいずれかを選択する必要があります。

CRC 巡回冗長検査 (CRC) メソッドは、各ストリングを反復可能な方法でマスクします。ただし、CRC メソッドは各ストリングを固有のストリングでマスクするとは限りません。

ハッシュ

ハッシュ・メソッドは、各ストリングを反復可能な方法でマスクします。ただし、ハッシュ・メソッドは各ストリングを固有のストリングでマスクするとは限りません。

マップ マップ・メソッドは、各ストリングを反復可能な方法で、そして固有なストリングでマスクします。

文字の置換

「文字の置換」ポリシーは、ストリングの各々の文字を、置換される文字のタイプに一致する、ランダムに生成された文字でマスクします。例えば、数字は数字に、小文字は小文字に置き換えられます。マスキングに使用される文字は指定の文字セットから取得されます。 このポリシーは、文字セットの一部となっている文字のみをマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.replaceCharacters()` で次のようにして使用できます。

```
ScrambleMask.replaceCharacters(record.getItem('<arg:inputAttribute>'),'<arg:language>')
```

例えば、ストリング中の値を、英語の文字セットの文字で置換するには、次のように入力します。

```
ScrambleMask.replaceCharacters(record.getItem('/DEMO/ORDERS/ORDER_ID'),'English')
```

引数	説明
<code>inputAttribute</code>	マスクするストリングを含んだ属性。
<code>language</code>	マスキングに使用される文字を提供する、文字セットの言語。 文字セットが指定されていない場合、または文字セットがサポートされていない場合は、英語の文字セットが使用されます。サポートされる文字セットのリストについては、70 ページの『スクランブル・マスク・ポリシーでサポートされる言語文字セット』を参照してください。

「文字の置換」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「文字の置換」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「文字の置換」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスキングに使用される文字を提供する文字セットを選択する必要があります。

正規表現による文字の置換

「正規表現による文字の置換」ポリシーは、ストリングの各々の文字を、置換される文字のタイプに一致する、ランダムに生成された文字でマスクします。例えば、数字は数字に、小文字は小文字に置き換えられます。このポリシーは、ストリング内のどの文字をマスクするかを決定するために、正規表現を使用します。マスキングに使用される文字は指定の文字セットから取得されます。このポリシーは、文字セットの一部となっている文字のみをマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.replaceCharactersByRegularExpression()` で次のようにして使用できます。

```
ScrambleMask.replaceCharactersByRegularExpression(record.getItem(
  '<arg:inputAttribute>'), <arg:regularExpression>, <arg:language>)
```

例えば、a から h までの小文字を、英語文字セットの文字で置換するには、次のように入力します。

```
ScrambleMask.replaceCharactersByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)', 'English')
```

引数

inputAttribute

regularExpression

language

説明

マスクするストリングを含んだ属性。

入力ストリングでマスクする文字を示す正規表現。

マスキングに使用される文字を提供する、文字セットの言語。 文字セットが指定されていない場合、または文字セットがサポートされていない場合は、英語の文字セットが使用されます。サポートされる文字セットのリストについては、70 ページの『スクランブル・マスク・ポリシーでサポートされる言語文字セット』を参照してください。

「正規表現による文字の置換」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「正規表現による文字の置換」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「正規表現による文字の置換」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスクする文字を指定する正規表現を提供する必要があります。
 - マスキングに使用される文字を提供する文字セットを選択する必要があります。

文字のスクランブル

「文字のスクランブル」ポリシーは、ストリング内の文字の順序をランダムに変更することで、ストリングをマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.scrambleCharacters()` で次のようにして使用できます。

```
ScrambleMask.scrambleCharacters(record.getItem('<arg.inputAttribute>'))
```

例えば、次のようにします。

```
ScrambleMask.scrambleCharacters(record.getItem('/DEMO/ORDERS/ORDER_ID'))
```

引数

`inputAttribute`

説明

マスクするストリングを含んだ属性。

「文字のスクランブル」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「文字のスクランブル」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「文字のスクランブル」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。 ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。

- 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。

正規表現による文字のスクランブル

「正規表現による文字のスクランブル」ポリシーは、ストリング内の文字の順序をランダムに変更することで、ストリングをマスクします。このポリシーは、ストリング内のどの文字をマスクするかを決定するために、正規表現を使用します。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.scrambleCharactersByRegularExpression()` で次のようにして使用できます。

```
ScrambleMask.scrambleCharactersByRegularExpression(record.getItem(
  '<arg.inputAttribute>'), <arg.regularExpression>)
```

例えば、a から h までの小文字をスワップするには、次のように入力します。

```
ScrambleMask.scrambleCharactersByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)')
```

引数	説明
<code>inputAttribute</code>	マスクするストリングを含んだ属性。
<code>regularExpression</code>	入力ストリングでマスクする文字を示す正規表現。

「正規表現による文字のスクランブル」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「正規表現による文字のスクランブル」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「正規表現による文字のスクランブル」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。
 - マスクする文字を指定する正規表現を提供する必要があります。

単純な文字のスクランブル

「単純な文字のスクランブル」ポリシーでは、ストリングの中の文字どうしをランダムにスワップすることで、ストリングをマスクします。このポリシーは反復可能な方法でストリングをマスクします。

JavaScript ポリシーの構文

このポリシーは、関数 `ScrambleMask.simpleScramble()` で次のようにして使用できます。

```
ScrambleMask.simpleScramble(record.getItem('<arg.inputAttribute>'))
```

例えば、次のようにします。

```
ScrambleMask.simpleScramble(record.getItem('/DEMO/ORDERS/ORDER_ID'))
```

引数

説明

`inputAttribute`

マスクするストリングを含んだ属性。

「単純な文字のスクランブル」ポリシーの作成

「ポリシーの追加」ウィザードを使用して、データ・アクセス計画に「単純な文字のスクランブル」ポリシーを作成できます。

選択ポリシーを含んだデータ・アクセス計画にポリシーを追加できます。

「単純な文字のスクランブル」ポリシーを作成するには、次のようにします。

1. Data Project Explorer で、論理データ・モデル・パッケージの「データ・アクセス計画」フォルダーを拡張します。
2. データ・アクセス計画を右クリックしてポリシーを組み込んでから、「開く」をクリックします。ポリシー・エディターが開きます。
3. 「ポリシーの追加」をクリックします。「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。
 - 入力値を提供するためのソース属性を選択する必要があります。
 - 入力値のすべてのインスタンスがマスクされる追加属性を選択することができます。
 - 入力値の中から保持する値を選択できます。例えば、保持される特定の値がソース属性に含まれる場合、その値はマスクされません。

スクランブル・マスク・ポリシーでサポートされる言語文字セット

文字の置換、正規表現による文字の置換、反復可能な置換、および正規表現による反復可能な置換ポリシーでは、マスキングに複数の言語文字セットがサポートされます。

JavaScript ポリシーに文字セットを入力するときには、以下の値を使用してください。

英語、アフリカーンス語、アルバニア語、アラビア語、アルメニア語、アッサム語、アゼルバイジャン語、ベラルーシ語、ベンガル語、ブルガリア語、カタロニア語、中国語 (簡体字)、中国語 (繁体字)、クロアチア語、チェコ語、デンマーク語、オランダ語、エストニア語、フランス語、フィンランド語、ドイツ語、グルジア語、グジャラート語、インドネシア語、ヘブライ語、ヒンディ語、ハンガリー語、ギリシャ語、アイスランド語、イタリア語、日本語、カナダ語、カザフ語、コンカニー語、韓国語、ラトビア語、リトアニア語、マケドニア語、マレー語、マラヤーラム語、マルタ語、マラーティー語、ネパール語、ノルウェー語、オリヤー語、ポルトガル語 (ブラジル)、ポルトガル語 (ポルトガル)、パンジャブ語、ポーランド語、ルーマニア語、ロシア語、セルビア語 (キリル文字)、セルビア語 (ローマ字)、シンハラ語、スロバキア語、スロベニア語、スペイン語、スワヒリ語、スウェーデン語、タミール語、テルグ語、タイ語、トルコ語、ウクライナ語、ウルドゥー語、ベトナム語、ウェールズ語

JavaScript ポリシー

JavaScript ポリシーは、データ管理サービスのソース属性に適用され、サービスの実行時に実行されます。サービス計画を使用することで、JavaScript ポリシーをサービスに追加します。

JavaScript ポリシーには、サービスの属性に適用されるデータ変換を記述した JavaScript ファイルが含まれます。JavaScript ファイルは、エンティティ内の複数の属性で使用できますが、各属性を関連付けできるのは 1 つのファイルのみとなります。サービスには、複数の JavaScript ポリシーを含めることができます。

JavaScript を使用して、属性のデータ変換を定義します。JavaScript を使用することで、数値をランダム値でマスクする、サブストリングを抽出する、エンティティ値を連結する、JavaScript で使用可能な他のデータ変換を実行するなどの処理が行えます。JavaScript 関数を使用して、日付マスク、身元マスク、数値マスク、およびスクランブル・マスクのポリシーを適用することも可能です。

JavaScript ポリシーで定義された変換は、Optim がソース・データに対してルックアップまたはルール・ベースの変換を実行した後に行われます。

ソース値の取得

「record」オブジェクトを使用してソース論理データ・モデルを参照し、getItem() メソッドを使用してソース属性にアクセスします。リレーショナル・データの場合、ソース項目は ('/schema/entity/attribute') の形式になります。

例えば DEMO スキーマの CUSTOMERS エンティティのソース属性 ADDRESS を取り出すには、次の構文を使用します。

```
record.getItem('/DEMO/CUSTOMERS/ADDRESS')
```

ストリングの連結

ストリングを連結するには、concat() 関数ではなく「+」演算子を使用します。

ソース値の処理

record.getItem() メソッドでソース値を処理する際、値は JavaScript の処理で Java™ データ・タイプに変換されます。JavaScript の処理の後、値はターゲット・データベースのデータ・タイプに変換されます。

JavaScript の処理におけるソース・データ・タイプの変換方法については、以下の表から判別してください。

ソース・データ・タイプ	Java タイプ
文字	java.lang.String
可変長文字	java.lang.String
国別文字	java.lang.String
可変長国別文字	java.lang.String
文字ラージ・オブジェクト	byte[] (IBM DB2 では java.sql.Clob) (Oracle では char[])
国別文字ラージ・オブジェクト	byte[] (IBM DB2 では java.sql.Clob) (Oracle では char[])
バイナリー	byte[]
可変長バイナリー	byte[]
バイナリー・ラージ・オブジェクト	java.sql.Blob
ブール値	java.lang.Boolean
日付	java.util.Calendar

ソース・データ・タイプ	Java タイプ
時刻	java.util.Calendar
タイム・スタンプ	java.sql.Timestamp (Oracle では java.lang.Object)
数値	java.math.BigDecimal
10 進数	java.lang.String
倍精度	java.lang.Double (Oracle では java.lang.String)
実数	java.lang.Double
浮動小数点	java.lang.Double (Oracle では java.math.BigDecimal)
短精度整数	java.lang.Short
整数	java.lang.Integer
64 ビット整数	java.lang.Long
インターバル	java.lang.Object
XML	java.lang.Object
データ・リンク	java.lang.Object

JavaScript ポリシーの作成

「ポリシーの追加」ウィザードを使用して、サービス計画に JavaScript ポリシーを作成できます。

JavaScript ポリシーを作成するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを展開して、ポリシーを追加するサービス計画を含むサービス要求を開きます。
2. 「サービス・プラン」ノードをダブルクリックします。 サービス計画エディターが開きます。
3. 「ポリシーの追加」をクリックします。 「ポリシーの追加」ウィザードが開きます。
4. ウィザードのステップを完了します。

「ポリシー・セクター」ページで「JavaScript ポリシー (JavaScript Policy)」を選択します。

JavaScript ポリシーへの JavaScript ファイルの追加

「JavaScript ファイルの追加 (Add JavaScript File)」ウィザードを使用して、JavaScript ポリシーに JavaScript ファイルを追加できます。 JavaScript ファイルは、サービスのソース・エンティティの属性に適用されます。ファイルを追加した後で、エディターを使用してファイルに式を追加できます。

JavaScript ポリシーに JavaScript ファイルを追加するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを展開して、ポリシーが使用されているサービス計画が含まれるサービス要求を開きます。
2. 「サービス・プラン」ノードをダブルクリックします。 サービス計画エディターが開きます。
3. 「ターゲット・ポリシー (Target Policies)」リストで、JavaScript ポリシーを選択します。 JavaScript ポリシー・エディターが開きます。
4. 「JavaScript の追加 (Add JavaScript)」をクリックします。 「JavaScript ファイルの追加 (Add JavaScript File)」ウィザードが開きます。
5. ウィザードのステップを完了します。

JavaScript ファイルを適用する属性を選択し、ファイル名を入力する必要があります。 ウィザードを終了すると、エディターが開きます。

6. エディターを使用して、ファイルに JavaScript 式を入力します。
7. 「ファイル」 → 「保存」をクリックして、ファイルを保存します。ファイルが、JavaScript ポリシー・エディターにリストされます。

JavaScript ポリシーの JavaScript ファイルの編集

JavaScript ポリシーの JavaScript ファイルを編集することができます。

JavaScript ポリシーの JavaScript ファイルを編集するには、次のようにします。

1. Data Project Explorer で「サービス」フォルダーを展開して、ポリシーが使用されているサービス計画が含まれるサービス要求を開きます。
2. 「サービス・プラン」ノードをダブルクリックします。サービス計画エディターが開きます。
3. 「ターゲット・ポリシー (Target Policies)」リストで、JavaScript ポリシーを選択します。JavaScript ポリシー・エディターが開きます。
4. JavaScript ファイルを選択して、「編集」をクリックします。JavaScript ファイルがエディターに開きます。
5. ファイルを編集します。「ファイル」 → 「保存」をクリックして、ファイルを保存します。

JavaScript 式の例

以下の例は、一般的な JavaScript 式を示しています。

サブストリング

CITY 属性値の最初の 15 文字のサブストリングを抽出するには、次の構文を使用します。

```
record.getItem('/DEMO/CUSTOMERS/CITY').substr(0,15)
```

連結

ADDRESS 属性の値を CITY および STATE 属性の値と連結して各値をスペースで分離するには、次の構文を使用します。

```
record.getItem('/DEMO/CUSTOMERS/ADDRESS')+ ' ' +record.getItem('/DEMO/CUSTOMERS/CITY')+ ' ' +record.getItem('/DEMO/CUSTOMERS/STATE')
```

if-else ステートメント

エラーを回避するために if-else 文を使用し、属性の値の長さがサブストリングの長さより短い場合はサブストリング・メソッドを無視するようにします。次の構文では、CUSTNAME の値が 8 文字以下の場合にはサブストリング・メソッドは使用されません。

```
var maxLength = 8 if ( record.getItem('/DEMO/CUSTOMERS/CUSTNAME').toString().length() > maxLength ) { record.getItem('/DEMO/CUSTOMERS/CUSTNAME').substr( 0, maxLength ) } else { record.getItem('/DEMO/CUSTOMERS/CUSTNAME') }
```

日付操作

(java.sql.Date クラスで使用するために) YYYY-MM-DD 形式のランダム日付を返すには、Date() メソッドを使用して現在日付を取得し、setDate() メソッドを使用してその日付に日数の乱数 (0 から 365) を追加し

ます。次に `getFullYear()`、`getMonth()`、および `getDate()` メソッドが返す値を連結して `YYYY-MM-DD` 形式で新しい日付を返します。以下の構文を使用してください。

```
var dob=new Date(); dob.setDate(dob.getDate()+Math.floor(Math.random()*365))
dob.getFullYear()+'-'+dob.getMonth()+'-'+dob.getDate()
```

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

以下は、International Business Machines Corporation の米国およびその他の国における商標です。

IBM
IBM LOGO
DB2
AIX
Informix
Optim

Adobe®、Adobe ロゴ、PostScript®、および Adobe ベースのすべての商標は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Microsoft、Windows、Windows NT® および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc.の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

以前のリリースの Optim 定義

移行 6

概要 6

[カ行]

クレジット・カードのポリシー

概要 48

クレジット・カード固有 48

すべての提供者からのクレジット・カード番号をマスクする 49

作成 49

提供者名に基づいてすべての提供者からのクレジット・カード番号をマスクする 50

作成 50

American Express のクレジット・カード番号のマスク 48

Diners Club のクレジット・カード番号のマスク 48

Discover のクレジット・カード番号のマスク 48

JCB のクレジット・カード番号のマスク 48

MasterCard のクレジット・カード番号のマスク 48

VISA のクレジット・カード番号のマスク 48

国民 ID のポリシー

イタリアの Fiscal Code Number のマスク 52

英国の National Insurance Number のマスク 53

概要 50

各国特有の国民 ID のポリシー 50

作成 50

カナダ社会保険番号のマスク 51

国名または国別コードに基づいて国民識別番号をマスクする 55

作成 55

スペインの Fiscal Identification Number のマスク 53

国民 ID のポリシー (続き)

フランスの National Institute for Statistics and Economic Studies Number のマスク 52

米国社会保障番号のマスク 54

[サ行]

サービス計画

概要 23

更新ポリシーの編集 26

サービス診断ポリシー 26

制約を使用不可にする、ポリシー 25

ソースからターゲットへのマッピング 23

編集 23

サービス診断ポリシー

概要 26

作成 26

サンプル・データベース

概要 6

ルックアップ表 7

EXTENDED_LOOKUP 表 6

数値マスク・ポリシー

概要 60

ガウス・ランダム整数 61

作成 61

ガウス・ランダム・ダブル 60

作成 60

範囲内一様ランダム・ダブル 61

作成 62

範囲内一様ランダム・ロング 62

作成 63

スクランブル・マスク・ポリシー

概要 63

サポートされる文字セット 70

正規表現による反復可能な置換 65

作成 65

正規表現による文字のスクランブル 69

作成 69

正規表現による文字の置換 67

作成 68

単純な文字のスクランブル 69

作成 70

反復可能な置換 63

作成 64

文字のスクランブル 68

作成 68

文字の置換 66

作成 67

制約を使用不可にする、ポリシー

概要 25

作成 25

すべての制約を使用可能/使用不可にする 25

選択した制約を使用可能/使用不可にする 25

選択基準

定義 19

選択ポリシー

エンティティの除去 19

エンティティの追加 19

開始エンティティの変更 19

関連エンティティおよび参照エンティティの定義 19

作成 17

選択基準 19

選択ポリシー・エディター 17

編集 16

ソースからターゲットへのマッピング

エンティティの除去 24

エンティティの追加 23

概要 23

自動マッピングのリストア 24

ターゲット論理データ・モデルの変更 24

[タ行]

データ管理サービス

概要 21

作成の概要 21

サブセット・サービスの作成 21

実行 28

データ変換サービスの作成 22

テストの概要 27

編集 23

ローカル Optim 管理データ・ストア 27

Optim Executor の構成 27

Optim リポジトリ 29

データ・アクセス計画

概要 16

選択ポリシー 16

追加 16

ポリシーの編集 16

データ・プライバシー・ポリシー

概要 43

日付マスク・ポリシー 43

身元マスク 46

ルックアップ 56

データ・プライバシー・ポリシー (続き)
JavaScript ポリシー 71
データ・モデル
概要 13
データ・アクセス計画 16
リバース・エンジニアリングに基づく
物理データ・モデル 13
Optim Database Relationship
Analyzer 14
Optim プロパティがある論理デー
タ・モデル 15

[ナ行]

ネイティブ・データ・ソース接続
概要 11
自動的な定義 11
手動での定義 11

[ハ行]

日付マスク・ポリシー
概要 43
範囲内のランダム日付 43
作成 44
日付を月に丸める 44
作成 45
日付を年に丸める 45
作成 46

[マ行]

身元マスク・ポリシー
概要 46
クレジット・カードのポリシー 48
国民 ID のポリシー 50
数値マスク・ポリシー 60
スクランブル・マスク・ポリシー 63
E メール・アドレス・ポリシー 46

[ラ行]

リバース・エンジニアリングに基づく物理
データ・モデル
概要 13
定義 13
ルックアップ・マスク・ポリシー
会社名マスク 59
概要 56
個人情報 57
作成 59
住所情報 57
ハッシュ・ルックアップ 56
ファーストネーム情報 58
ラストネーム情報 58

ルックアップ・マスク・ポリシー (続き)
ランダム・ルックアップ 56
ローカル Optim 管理データ・ストア
概要 27
構成 28
接続 28
データの表示 28

D

Database Relationship Analyzer
物理モデル 15

E

E メール・アドレス・ポリシー
概要 46
自動生成される E メール名 47
自動生成ユーザー名
作成 47
フォーマット設定された E メール名
47
作成 47

J

JavaScript ポリシー
概要 71
作成 72
例 73
JavaScript ファイルの追加 72
JavaScript ファイルの編集 73

O

Optim Database Relationship Analyzer
使用 14
接続プロファイルの定義 14
セットアップ 14
Optim for z/OS 要求モデル
概要 34
実行 36
Optim for z/OS バッチ・ホスト 35
Optim for z/OS バッチ・ホスト・エデ
ィター 35
Optim 相互運用性モデル
アーカイブ要求 31
作成 31
概要 31
削除要求 31
作成 32
挿入要求 33
作成 33
抽出要求 32
作成 32

Optim 相互運用性モデル (続き)
復元要求 34
作成 34
ロード要求 33
作成 33
Optim for z/OS 要求モデル 34
Optim ディレクトリー 37
Optim 要求モデル 36
Optim ディレクトリー
概要 37
接続プロファイルの定義 37
プロジェクトの作成 38
Optim デザイナー
アクセシビリティ機能 8
概要 1
新機能 2
データのマスキング 3
データベース接続 9
データベース・サポート 7
ヘルプ 9
JDBC 接続の詳細 9
Optim パースペクティブ
概要 4
データ・ソース・エクスプローラー 5
Data Project Explorer 4
Optim 要求モデル
概要 36
実行 40
定義のインポート 39
定義のインポート、 OEF から
OIM 39
定義のインポート、 OEF から Optim
ディレクトリー 40
定義のインポート、 Optim ディレク
トリーから OIM 40
定義のエクスポート 38
定義のエクスポート、 OIM から
OEF 38
定義のエクスポート、 OIM から
Optim ディレクトリー 38
定義のエクスポート、 Optim ディレ
クトリーから OEF 39
prOcmnd の場所 37
Optim リポジトリー
概要 29
サービス要求の公開 29
デフォルトの場所の入力 29



Printed in Japan