PUBLIC (公開)
SAP BusinessObjects Business Intelligence プラットフォーム
ドキュメントバージョン: 4.3 Support Package 4 – 2023-12-07

BI セマンティックレイヤ Java SDK 開発者ガイド

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

THE BEST RUN SAP

目次

1 ドキュメント履歴 5

2 BI セマンティックレイヤ Java SDK 開発者ガイドの紹介 6

2.1 対象者 6

2.2 このガイドの表記規則 6

3 セマンティックレイヤについて 8

3.1 ユニバースについて 8

3.2 接続について 9

3.3 セキュリティプロファイルについて 9

4 BI Semantic Layer Java SDK について 10

4.1 使用事例 10

4.2 ユーザー要件 11

4.3 環境要件 11

4.4 アプリケーションの目的 11

リソースの作成 12

リソースの公開 12

リソースの取得 13

データファンデーションの使用 13

ビジネスレイヤの使用 16

セキュリティプロファイルの使用 19

接続の使用 20

ユニバースの変換 20

4.5 SDK オブジェクトモデル 20

リソース 21

データファンデーション 21

ビジネスレイヤ 22

ビジネスレイヤ項目 22

表示書式 23

値の一覧 23

パラメータとプロンプト 24

プロパティ 24

セキュリティプロファイル 25

接続 26

4.6 SDK パッケージ 26

4.7 SDK サービス 27

4.8 ユニバースを操作するための SDK メソッド 28

5 BI セマンティックレイヤ Java SDK のインストールおよびデプロイ 31
5.1 BI セマンティックレイヤ Java SDK をクライアントツールとともにインストールする 31
5.2 インストールされた内容 32
5.3 Windows において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする 33
5.4 UNIX または Linux において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする 34
5.5 非 OSGI Eclipse 設定で BI セマンティックレイヤ Java SDK をデプロイする 34
5.6 Eclipse OSGI フレームワーク設定で BI セマンティックレイヤ Java SDK をデプロイする 36

6 BI Semantic Layer Java SDK の使用 38
6.1 BI セマンティックレイヤ Java SDK のイベントフロー 38
セッションの作成 39
サービスのロード 39
ファクトリの使用 40
リソースのインスタンス化 43
オブジェクトの保存 43
オブジェクトのチェック 44
オブジェクトを解放する 46
エラーの発生 46
6.2 java.util.List メソッドの使用 47
テーブルの削除 48
ビジネス レイヤ アイテムの削除 48
ビジネスレイヤ項目の移動 49
結合の削除 50
追加テーブルの追加 50
コンテキストへの結合の追加 51
1 次キーからの列の削除 51
静的値の一覧の行への値の追加 52
カスタムナビゲーションパスへのディメンションの追加 52
6.3 デフォルト値の使用 52
6.4 データ表示書式の使用 53
6.5 データファンデーションビューの使用 55
6.6 リンクされたユニバースの使用 56
6.7 クエリスクリプトプロパティの使用 59
6.8 ビジネスフィルタ式とビジネスクエリの保存方法 61
6.9 データセキュリティプロファイルのテーブル設定の作業 66
6.10 ビジネスセキュリティプロファイルのセキュリティ保護された要素の操作 66
6.11 データファンデーションの構造の最新表示 68
6.12 整合性のチェックの実行 68
6.13 複数ソース有効ユニバースの使用 69
接続ライフサイクルの管理 70
複数ソース有効データファンデーションを作成する 71

複数ソース有効データファンデーションからのデータソースの削除. 71
複数ソース有効データファンデーションの派生テーブルの作成. 72

7 BI セマンティックレイヤ Java SDK サンプルによる開発. 74
7.1 サンプルパッケージ. 74
7.2 サンプルを使用して開発する. 75

8 BI Semantic Layer Java SDK のトラブルシューティング. 77
8.1 CreateProcess エラー. 77
8.2 csEx でサポートされない操作. 77
8.3 ドライバ不明. 78
8.4 無効なデータベーステーブル. 78

9 付録. 79
9.1 接続パラメータリファレンス. 79
ユーザー権限について. 79
接続パラメータについて. 80
RDBMS 接続. 83
SAP 接続. 92
OLAP 接続. 96
9.2 事前定義済みクエリスクリプトプロパティのリファレンス. 98
9.3 識別可能なオブジェクト参照. 99
9.4 ValueFormat 文字列パターン参照. 100
数値パターン. 100
日時パターン. 101

1 ドキュメント履歴

以下の表は、最も重要なドキュメント変更の概要です。

バージョン	日付	変更
SAP BusinessObjects Business Intelligence プラットフォーム 4.3	2020 年 6 月	初期リリース

2 BI セマンティックレイヤ Java SDK 開発者ガイドの紹介

このガイドは、SAP BusinessObjects Business Intelligence プラットフォーム 4.2 サポートパッケージ 4 リリースに関連しています。

SAP BusinessObjects BI Semantic Layer Java SDK 開発者ガイドは、SAP BusinessObjects BI セマンティックレイヤ Java SDK の機能および技術に関する情報を提供します。

- 目的と機能
- SDK をインストールおよびデプロイする方法
- コード例での API 使用

BI セマンティックレイヤ Java API のインタフェース、クラス、およびメソッドの詳細については、SAP Help Portal にある *SAP BusinessObjects BI* セマンティックレイヤ *Java API* リファレンスを参照してください。

API の基本オブジェクトモデルの詳細については、SAP Help Portal にある *SAP BusinessObjects BI Semantic Layer Java SDK Object Model Diagrams* を参照してください。

対象者 [6 ページ]

このガイドの表記規則 [6 ページ]

2.1 対象者

SAP BusinessObjects BI セマンティック レイヤ *Java SDK* 開発者ガイドは、Java 開発者を対象としています。

Java 開発者は、ユニバースおよびそのリソースに対して作成、編集、公開、セキュリティの各タスクを実行するアプリケーションを記述します。これらのアプリケーションは、内部使用のために開発することも、BI platform を統合する Business Intelligence ソリューションに埋め込むこともできます。

2.2 このガイドの表記規則

このガイドでは、`<slsdk-install-dir>` 変数が BI セマンティックレイヤ Java SDK のインストールルートパスです。 Microsoft Windows では、デフォルトの `<slsdk-install-dir>` は、`C:¥Program Files (x86)¥SAP Business Objects¥SAP BusinessObjects Enterprise XI 4.0¥SL SDK` ディレクトリを表します。

`<bip-install-dir>` 変数は、BI プラットフォームまたはクライアントツールのインストールルートパスです。MS Windows (64 ビット) では、`C:¥Program Files (x86)¥SAP Business Objects` ディレクトリを表します。

このガイド全体で、Central Management Server (CMS) の接続およびユニバースのルートフォルダは、次のディレクトリを参照します。

- `/Connections`
- `/Universes`

これは、API の `CONNECTIONS_ROOT` および `UNIVERSES_ROOT` 定数値を参照します。 *SAP BusinessObjects BI Semantic Layer Java API reference* を参照してください。

3 セマンティックレイヤについて

この節では、BI Semantic Layer Java SDK で使用されているセマンティックレイヤリソースのいくつかの定義について説明しています。

ユニバースについて [8 ページ]

接続について [9 ページ]

セキュリティプロファイルについて [9 ページ]

3.1 ユニバースについて

ユニバースとは、編成されたメタデータオブジェクトのコレクションのことで、これにより、専門用語を使わずに、ビジネスユーザが企業のデータを分析してレポートを作成できます。

ユニバースの役割は、ビジネスユーザに意味を理解できるビジネスオブジェクトを提供することです。ユーザは、基になるデータソースと構成にかかわらず、関連するビジネス用語を使用して、データを分析してレポートを作成できます。

リレーショナルユニバースを使用すると、ユーザは 1 つまたは複数のリレーショナルデータベースのデータを分析できます。これは、ビジネスレイヤ、データファンデーション、およびリレーショナルデータベースに対する 1 つ以上の接続で作成されます。多次元のユニバースを使用すると、ユーザーは OLAP キューブのデータを分析できます。このユニバースはビジネスレイヤと OLAP キューブに対するビジネスレイヤの接続で作成されます。

ビジネスレイヤ、データファンデーション、および接続はインフォメーションデザインツールで作成できます。一度作成すると、これらは結合されてユニバースとして公開され、SAP BusinessObjects レポートツールで使用できます。

ユニバースはローカルのファイルシステムまたは CMS リポジトリに公開できるため共有可能で、CMS リポジトリのセキュリティの利点を活用できます。

ユニバースは、アグリゲート認識をサポートできます。これは、事前集計されたデータを含むデータベーステーブル (集計テーブル) を活用する、リレーショナルユニバースの機能です。アグリゲート認識を設定すると、処理するファクト数と集計する行数が減り、クエリの実行が加速します。

CMS リポジトリに公開された単一ソースリレーショナル UNX ユニバースは、1 つ以上のユニバース (リンクされたユニバース) のコアユニバースとして機能します。インフォメーションデザインツールでエンドユーザがリンクされたユニバースを作成した場合、コアユニバースのデータファンデーション内のテーブル、結合、値の一覧、およびプロンプトはリンクされたデータファンデーションに表示されますが、読み取り専用になります。コアユニバースのビジネスレイヤ内のフォルダ、オブジェクト、値の一覧、およびプロンプトはリンクされたビジネスレイヤに表示されますが、読み取り専用になります。ステータスのみを変更することができます。

ユニバース、ビジネスレイヤ、およびデータファンデーションの詳細については、インフォメーションデザインツールユーザガイドを参照してください。

3.2 接続について

接続とは、いくつかの SAP BusinessObjects アプリケーションがリレーショナルデータベースまたは OLAP データベースにアクセスする方法を定義する一連のパラメータです。接続は、ローカルファイルか、インフォメーションデザインツールではローカルショートカットとして参照される CMS リポジトリ内のリモートオブジェクトになります。

接続には、ローカル接続とセキュリティ接続があります。ローカル接続は、ローカル ファイル システムの独立オブジェクトとして .cnx ファイルに保存されます。ビジネスユーザーは、ローカル接続を CMS リポジトリに公開するか、または CMS リポジトリ内に接続を直接作成することによってセキュリティ接続を作成できます。

接続をユニバース定義の一部または個別リソースとして CMS リポジトリに公開すると、その接続はセキュリティ接続として保存されます。セキュリティ接続を参照するには、接続のショートカットをローカルで作成する必要があります。

ローカル接続とセキュリティ接続の詳細については、インフォメーションデザインツールユーザーガイドを参照してください。

3.3 セキュリティプロファイルについて

セキュリティプロファイルは、リポジトリで公開されたユニバースに適用される、セキュリティ設定グループです。この設定は、表示されるデータを制御し、データファンデーションまたはビジネスレイヤ、あるいはその両方で定義されているパラメータを変更します。

プロファイルには、次の 2 種類があります。

- データセキュリティプロファイルには、データファンデーション内および接続上で定義されたセキュリティ設定があります。
- ビジネスセキュリティプロファイルには、ビジネスレイヤで定義されたセキュリティ設定があります。

> ⓘ 注記
>
> 4.2 SP04 以降、ユニバースのセキュリティビジネスプロファイルおよびデータプロファイルをダウンロードし、各プロファイルを個別に編集するのではなくローカルで 1 回の操作でこれらを編集できるようになりました。詳細については、ビジネスおよびデータセキュリティプロファイルの編集 [25 ページ] を参照してください。

セキュリティの詳細については、インフォメーションデザインツールユーザガイドを参照してください。

4 BI Semantic Layer Java SDK について

BI Semantic Layer Java SDK は、インフォメーションデザインツールの機能の開発用フレームワークとして使用する API を提供します。

この API では、インフォメーションデザインツールを使用しなくても、ユニバースを使用したり 1 つ以上のデータソースとのセキュリティ保護された接続を使用できます。 たとえば、ローカルなビジネスレイヤからローカルな .unx ユニバースを作成してリポジトリに公開するサービスや、セキュリティ保護された接続をリポジトリから取得してワークスペースに渡すサービスを実装できます。

SDK は、API のインタフェース、クラス、およびメソッドを使用するのに必要なすべてのリソースと情報を提供し、ビジネスに必要な機能を実装できるようにします。

SDK には、Java サンプルとそのソースコードが用意されています。 これらによって、IDE におけるアプリケーション開発が容易になり、ユーザは独力で開発できます。

SDK は、SAP BusinessObjects Business Intelligence プラットフォーム 4.0 Feature Pack 3 より前に配信された CMS リポジトリと組み合わせて使用することはできません。

使用事例 [10 ページ]

ユーザー要件 [11 ページ]

環境要件 [11 ページ]

アプリケーションの目的 [11 ページ]

SDK オブジェクトモデル [20 ページ]

SDK パッケージ [26 ページ]

SDK サービス [27 ページ]

ユニバースを操作するための SDK メソッド [28 ページ]

4.1 使用事例

BI セマンティックレイヤ Java SDK を使用して、以下の要件に対応できます。

- プログラムでのユニバースリソースのカスタマイズ
  たとえば、顧客データベースの技術要件にユニバースリソースを適合させるためアプリケーションでデータファンデーションテーブルの所有者と修飾子を変更し、リソースを CMS に公開できます。
- プログラムでのビジネスオブジェクトのカスタマイズ
  たとえば、アプリケーションでユニバースに含まれる一部のビジネスアイテムの名前を変更して、ユニバースリソースを CMS に公開することができます。 また、データセキュリティプロファイルまたはビジネスセキュリティプロファイルを作成して、ユーザグループに割り当てることもできます。
- プログラムでのユニバースリソースの作成および公開
- .unx ユニバースへの .unv ユニバースのバルク変換の実行

4.2 ユーザー要件

アプリケーションを開発するには、次の知識と経験が必要です。

- Eclipse を使用した Java アプリケーションの開発およびテスト（推奨）（OSGI モード）
- BusinessObjects Enterprise（BOE）Java SDK の概要の理解
- インフォメーションデザインツールに関する知識
- メタデータの設計概念の概要 (データファンデーション、ビジネスレイヤ、接続、ユニバース)
- ユニバースの作成と公開に関するライフサイクルの知識
- ローカル接続およびセキュリティ接続の概念の概要
- データ セキュリティ プロファイルとビジネス セキュリティ プロファイルの概念の概要

4.3 環境要件

アプリケーションを開発および構築するには、以下の環境が必要です。

- Java 開発環境を実行できるコンピュータハードウェア
- Eclipse 3.8.2 以上
- 64 ビット Java Runtime Environment 1.6 以上

このリリースでは、BI セマンティックレイヤ Java SDK は、Microsoft Windows および UNIX フレーバでサポートされています。

関連情報

インストールされた内容 [32 ページ]

4.4 アプリケーションの目的

BI セマンティックレイヤ Java SDK を使用すると、以下の節で説明されている一部の機能を提供できるアプリケーションを記述することができます。SDK は、これらの節に記載されている機能のみを提供します。

リソースの作成 [12 ページ]

リソースの公開 [12 ページ]

リソースの取得 [13 ページ]

データファンデーションの使用 [13 ページ]

ビジネスレイヤの使用 [16 ページ]

セキュリティプロファイルの使用 [19 ページ]

接続の使用 [20 ページ]
ユニバースの変換 [20 ページ]

4.4.1 リソースの作成

- ローカルでのリレーショナル接続または OLAP 接続の作成
- 公開された接続からローカルでの接続ショートカットの作成
- ローカルでの単一ソースデータファンデーションの作成
- ローカルでの複数ソース有効データファンデーションの作成
- ローカルでのビジネスレイヤの作成
- リンクされたデータファンデーションまたはビジネスレイヤのローカルでの作成

> **注記**
>
> SDK では、リンクされたユニバースを以下の場合に作成することはできません。
>
> - リンクされたデータファンデーションが複数ソース対応である。
> - リンクされたビジネスレイヤが多次元である。
> - コアユニバースがリンクされたユニバースである。
> - コアユニバースが UNV ユニバースである。

関連情報

複数ソース有効ユニバースの使用 [69 ページ]

4.4.2 リソースの公開

- 単一ソースユニバースのローカルファイルシステムへの公開
- 単一ソースユニバースまたは複数ソース有効ユニバースの CMS リポジトリへの公開
- ローカル接続 (.cnx) の CMS リポジトリへの公開
- リンクされたユニバースの CMS リポジトリへの公開

> **注記**
>
> SDK では、リンクされたユニバースをローカルに公開することはできません。

4.4.3 リソースの取得

- ローカルのビジネスレイヤ、ローカルの単一ソースユニバースからのデータファンデーションと接続の取得
- 公開されたユニバースを取得して関連付けられたリソースの編集
- CMS リポジトリに公開されたセキュリティ接続への接続ショートカットの作成
- CMS リポジトリに保存されている .unx ユニバースの改訂番号の取得
- CMS リポジトリでの CUID からの .unx または .unv ユニバースまたはセキュリティ保護された接続のパスの取得
- CMS リポジトリでのパスからの .unx または .unv ユニバースまたはセキュリティ保護された接続の CUID の取得

4.4.4 データファンデーションの使用

コンテキスト

- コンテキストの含まれる結合の一覧と除外結合の一覧の取得
- コンテキストの結合の一覧の編集 (結合の追加または削除)

カスタムプロパティ

- データファンデーションのカスタムプロパティの作成
- データファンデーションのカスタムプロパティの一覧の取得と編集 (カスタムプロパティの追加または削除)

データファンデーション

- データファンデーションの名前、説明、およびパスの編集
- データファンデーションの SQL オプションの設定
- データファンデーションのテーブル、列、結合、コンテキスト、および 1 次キーの作成
- 計算列の作成
- 派生テーブルの作成
- 静的な値の一覧および SQL ベースの値の一覧の作成と、データファンデーションへの添付
- データファンデーションにおける値の一覧の一覧の取得と編集 (値の一覧の追加または削除)
- パラメータの作成とデータファンデーションへの添付
- データファンデーションのパラメータの一覧の取得と編集 (パラメータの追加または削除)
- データファンデーションの構造の最新表示
- データファンデーションでの整合性チェックの実行
- データファンデーションの接続の設定

データファンデーションビュー

- データファンデーションビューの作成
- データファンデーションビュー一覧の取得
- データファンデーションビューのテーブル一覧の取得
- データファンデーションビューの説明の設定および取得
- ビューでのテーブルの位置、幅、および表示状態の設定および取得
- データファンデーションビューへのコメントの挿入とそのテキストおよび表示の編集 (フォント、色、および不透明度)

テーブルと列

- データファンデーションのテーブルの一覧の取得
- データファンデーションのテーブルの一覧の編集 (テーブルの追加または削除)
- テーブルの名前、所有者、修飾子からのテーブルの完全修飾名の取得
- テーブルの完全修飾名からのテーブルの名前、所有者、修飾子の取得
- テーブルの名前、所有者、修飾子、列の一覧の編集
- 計算列の SQL 式の設定と取得
- 列プロパティ (Null 値を使用できる、符号付き、表示) の設定
- コアユニバースとそこに含まれるオブジェクト識別子 (存在する場合) の取得

結合

- データファンデーションの結合の一覧の取得
- データファンデーションの結合の一覧の編集 (結合の追加または削除)
- コアユニバースとそこに含まれるオブジェクト識別子 (存在する場合) の取得

キー

- テーブルの 1 次キーの取得
- テーブルの 1 次キーの編集 (列の追加または削除)
- テーブルの 1 次キーの削除

値の一覧

- 値の一覧の名前、説明の編集
- SQL ベースの値の一覧のオプション (クエリオプション) の設定
- 静的な値の一覧の列と行の作成と編集
- SQL ベースの値の一覧の列の検出
- 静的な値の一覧または SQL ベースの値の一覧の列のプロパティ (列名、オブジェクト名、キー列、データ型、非表示) の設定
- コアユニバースとそこに含まれるオブジェクト識別子 (存在する場合) の取得

複数ソース有効データファンデーション

- 複数ソース有効データファンデーションのデータソースの一覧の取得
- 複数ソース有効データファンデーションのデータソースの一覧の編集 (ソースの追加または削除)

パラメータ

- パラメータの名前、説明の編集
- パラメータオプション (プロンプトオプション、値の一覧、およびデフォルト値) の設定
- パラメータのカスタムプロパティの作成と編集
- パラメータの回答と値の作成と編集
- コアユニバースとそこに含まれるオブジェクト識別子 (存在する場合) の取得

クエリスクリプトプロパティ

- データファンデーションへの事前定義済みまたはカスタムクエリスクリプトプロパティの追加
- データファンデーションのクエリスクリプトプロパティの一覧の編集 (クエリスクリプトプロパティの追加または削除)
- データファンデーションの事前定義済みまたはカスタムクエリスクリプトプロパティの値の設定
- データファンデーションのクエリスクリプトプロパティの一覧のリセット
- データファンデーションの事前定義済みおよびカスタムクエリスクリプトプロパティの一覧の取得

関連情報

java.util.List メソッドの使用 [47 ページ]
データファンデーションの構造の最新表示 [68 ページ]

4.4.5 ビジネスレイヤの使用

アグリゲート認識の非互換性

- アグリゲート認識の非互換性の作成
- アグリゲート認識の非互換性リストの取得

ビジネスレイヤ

- ビジネスレイヤの名前、説明、およびパスの編集
- ビジネスレイヤのクエリオプションの設定
- ビジネスレイヤのコメントの編集 (追加または削除)
- フォルダとビジネスレイヤ項目 (メジャー、ディメンション、属性、およびフィルタ) の作成
- ビジネスフィルタの作成
- ビジネスレイヤ式の取得、編集、および検証
- 値の一覧 (静的、SQL ベース、階層、およびビジネスオブジェクトで作成されたクエリに基づく) の作成とビジネスレイヤへの添付
- ビジネスレイヤにおける値の一覧の一覧の取得と編集 (値の一覧の追加または削除)
- パラメータの作成とビジネスレイヤへの添付
- ビジネスレイヤのパラメータの一覧の取得と編集 (パラメータの追加または削除)
- フォルダとビジネスレイヤ項目の移動
- ビジネスレイヤフォルダの参照とフォルダに含まれるビジネスレイヤ項目の読み込み
- フォルダとビジネスレイヤ項目の削除
- ビジネスレイヤでの整合性チェックの実行

> **注記**
>
> - SDK では、多次元のビジネスレイヤ項目はサポートされません。
> - SDK では、アグリゲート認識の非互換性を自動で検出することはできません。

ビジネスレイヤ項目

- ビジネスレイヤ項目 ID の取得
- ビジネスレイヤ項目の名前、説明、状態、および見通し関数の取得と編集
- 完全パスからのビジネスレイヤ項目の取得
- ビジネスレイヤ項目の完全パスの取得
- ビジネスレイヤ項目の SQL 定義 (SELECT、WHERE、および追加テーブル) の取得と編集
- ビジネスレイヤ項目の詳細プロパティ (アクセスレベル、使用オブジェクト、値の一覧) の取得と編集
- ビジネスオブジェクトのソース情報 (技術情報、マッピング、およびリネージ) の取得と編集

- ビジネスレイヤ項目の暗黙のテーブルの取得
- アグリゲート認識の非互換オブジェクトの一覧の取得
- ビジネスレイヤ項目のカスタムプロパティの作成
- ビジネスレイヤ項目のカスタムプロパティの取得と編集 (カスタムプロパティの追加または削除)
- ビジネスレイヤの数値および日時ビジネスオブジェクト用の定義済み表示書式およびカスタム表示書式の作成
- 定義済み表示書式およびカスタム表示書式の取得と編集
- コアユニバースとそこに含まれるオブジェクト識別子 (存在する場合) の取得

ビジネスレイヤビュー

- ビジネスレイヤビューの作成と編集
- ビジネスレイヤビューに含まれるオブジェクトの一覧の定義
- マスタビューの非表示

カスタムプロパティ

- ビジネスレイヤのカスタムプロパティの作成
- ビジネスレイヤのカスタムプロパティの一覧の取得と編集 (カスタムプロパティの追加または削除)

リンクされたユニバース

- リンクされたユニバースの作成
- ビジネスレイヤにリンクされたコアユニバースの一覧の編集 (追加または削除)
- ビジネスレイヤにリンクされたコアユニバースの一覧の取得
- リンクされたローカルユニバースへのコアユニバースコンポーネントの追加
- CMS リポジトリ内でのリンクされたユニバースのコアユニバースコンポーネントの最新バージョンとの同期
- リンクされたユニバースで使用されているコアユニバースのパスおよび名前の更新

値の一覧

- 値の一覧の名前、説明の編集
- 値の一覧のオプションの設定
- 静的な値の一覧の列と行の作成と編集
- SQL ベースの値の一覧の列の検出
- 静的な値の一覧または SQL ベースの値の一覧の列のプロパティ (列名、オブジェクト名、キー列、データ型、非表示) の設定

- ビジネスクエリに基づく値の一覧の列の検出
- 値の一覧のビジネスクエリの取得、編集、および検証
- 値の一覧とビジネスオブジェクト (ディメンション、メジャー、または属性) の関連付け
- 値の一覧とパラメータの関連付け

> ⚠ 制限
>
> ビジネスオブジェクトには、ビジネスレイヤで作成された値の一覧のみを関連付られます。

ナビゲーションパス

- ビジネスレイヤのカスタムナビゲーションパスの作成
- カスタムナビゲーションパスの取得および編集 (ディメンションの追加または削除)
- ビジネスレイヤに使用するカスタムナビゲーションパスの設定および取得

パラメータ

- パラメータの名前、説明の編集
- パラメータオプション (プロンプトオプション、値の一覧、およびデフォルト値) の設定
- パラメータのカスタムプロパティの作成と編集
- パラメータのカスタムプロパティの取得と編集 (カスタムプロパティの追加または削除)
- パラメータの回答と値の作成と編集
- ビジネスオブジェクト (ディメンション、属性) とパラメータの関連付け

クエリスクリプトプロパティ

- ビジネスレイヤへの事前定義済みまたはカスタムクエリスクリプトプロパティの追加
- ビジネスレイヤのクエリスクリプトプロパティの一覧の編集 (クエリスクリプトプロパティの追加または削除)
- ビジネスレイヤの事前定義済みまたはカスタムクエリスクリプトプロパティの値の設定
- ビジネスレイヤのクエリスクリプトプロパティの一覧のリセット
- ビジネスレイヤの事前定義済みおよびカスタムクエリスクリプトプロパティの一覧の取得

関連情報

java.util.List メソッドの使用 [47 ページ]

4.4.6 セキュリティプロファイルの使用

データセキュリティプロファイル

- "行" 設定によるデータセキュリティプロファイルの作成とユニバースへの添付
- "テーブル" 設定によるデータセキュリティプロファイルの作成とユニバースへの添付
- "接続" 設定によるデータセキュリティプロファイルの作成とユニバースへの添付
- ユニバースに添付されたデータセキュリティプロファイルの取得
- ユニバースからのデータセキュリティプロファイルの切り出し
- ユーザまたはグループへのデータセキュリティプロファイルの割り当て、または割り当て解除

> ⓘ 注記
>
> 4.2 SP04 以降、(データおよびビジネスの) セキュリティプロファイルを 1 回の操作で取得し、ローカルですべての設定を編集してから公開できるようになりました。これにより、設定を 1 つずつ取得する必要がなくなります。

ビジネスセキュリティプロファイル

- "クエリの作成" 設定によるビジネスセキュリティプロファイルの作成とユニバースへの添付
- "データの表示" 設定によるビジネスセキュリティプロファイルの作成とユニバースへの添付
- ユニバースに添付されたビジネスセキュリティプロファイルの取得
- ユニバースからのビジネスセキュリティプロファイルの切り出し
- ユーザまたはグループへのビジネスセキュリティプロファイルの割り当て、または割り当て解除

> ⓘ 注記
>
> 4.2 SP04 以降、(データおよびビジネスの) セキュリティプロファイルを 1 回の操作で取得し、ローカルですべての設定を編集してから公開できるようになりました。これにより、設定を 1 つずつ取得する必要がなくなります。

関連情報

ビジネスセキュリティプロファイルのセキュリティ保護された要素の操作 [66 ページ]
データセキュリティプロファイルのテーブル設定の作業 [66 ページ]
ビジネスおよびデータセキュリティプロファイルの編集 [25 ページ]

4.4.7 接続の使用

- ローカル接続のテスト
- 接続ドライバの変更
- CMS リポジトリに保存されているセキュリティ接続の一部のパラメータの更新
- 単一ソースユニバースに追加されているセキュリティ接続の CMS リポジトリ内の他の接続との置換
- 複数ソース有効ユニバースに追加されている複数のセキュリティ接続の CMS リポジトリ内の他のユニバースとの置換
- 接続での整合性チェックの実行

> ⓘ 注記
>
> - SDK では、インフォメーションデザインツールと同じ接続タイプがサポートされています。
> - SDK では、リンクされたユニバースに追加された接続を CMS リポジトリ内の別の接続に置換することはできません。

> → 注意
>
> CMS で .cnx 接続を作成できません。接続は、ローカルで作成して保存した後に CMS に公開する必要があります。

4.4.8 ユニバースの変換

- ローカルの .unv ユニバースの .unx ユニバースへの変換
- セキュリティ保護された .unv ユニバースから CMS リポジトリの .unx ユニバースへの変換

4.5 SDK オブジェクトモデル

一連のモデルによって SDK で使用されるオブジェクトを定義します。次の節では、これらのオブジェクトに関する詳細を説明します。詳細については、SAP Help Portal の「BI プラットフォームでの開発」ページで、*SAP BusinessObjects BI Semantic Layer Java SDK Object Model Diagrams* を参照してください。

リソース [21 ページ]

データファンデーション [21 ページ]

ビジネスレイヤ [22 ページ]

ビジネスレイヤ項目 [22 ページ]

表示書式 [23 ページ]

値の一覧 [23 ページ]

パラメータとプロンプト [24 ページ]

プロパティ [24 ページ]

セキュリティプロファイル [25 ページ]
接続 [26 ページ]

4.5.1 リソース

リソースはユニバースを構成するオブジェクトを定義します。 該当するオブジェクトは、DataFoundation、BusinessLayer、および Connection です。 リソースのルートオブジェクトは SlResource です。

4.5.2 データファンデーション

DataFoundation オブジェクトは SDK オブジェクトモデルのデータファンデーションを表しています。 MonoSourceDataFoundation オブジェクトは 1 つのデータソース (1 つの接続) に基づくデータファンデーションを指定しますが、MultiSourceDataFoundation オブジェクトは、複数のデータソース (複数の接続) に基づくデータファンデーションを指定します。

Table オブジェクトはデータファンデーションテーブルを定義し、Column オブジェクトはテーブル列を定義します。 テーブルは、データベーステーブル、派生テーブル、エイリアステーブル、または連合テーブルのいずれかにすることができます。 DatabaseTable オブジェクトはデータソーステーブルを定義し、データベース情報 (所有者および修飾子) を提供します。 DerivedTable オブジェクトは、データファンデーションの仮想テーブルを定義します。 AliasTable オブジェクトは、データファンデーションの標準テーブルまたは派生テーブルへの参照を定義します。

CalculatedColumn および InputColumn オブジェクトは、計算列と入力列を明示的に定義します。 以下の種類の列は、SAP システムへの接続など特定の接続に使用されます。

Context オブジェクトはデータファンデーションのコンテキストを定義し、SQLJoin オブジェクトはデータファンデーションテーブル間の結合を定義します。

PrimaryKey オブジェクトは、DatabaseTable の 1 次キーを定義します。

DataFoundationView オブジェクトは、SDK オブジェクトモデルでのマスタビューおよびカスタムデータファンデーションビューを表します。 TableView オブジェクトは、ビュー内のデータファンデーションテーブルの特定のビューを表します。 ビューは、連合テーブルを含め、任意のテーブルを参照することができます。

FederatedTable オブジェクトは、インフォメーションデザインツールのデータファンデーションエディタで作成された連合テーブルを表します。

関連情報

データファンデーションビューの使用 [55 ページ]

4.5.3 ビジネスレイヤ

BusinessLayer オブジェクトは SDK オブジェクトモデルのビジネスレイヤを表します。 RootFolder オブジェクトは、BusinessLayer オブジェクトのルートコンテナです。 RelationalBusinessLayer オブジェクトは、BusinessLayer から継承され、リレーショナルデータソースのビジネスレイヤを表します。

BusinessLayerView オブジェクトは、SDK オブジェクトモデルのビジネスレイヤのビジネスレイヤビューを表します。 ビジネスレイヤビューには、ビジネスレイヤの一連のビジネスオブジェクトが含まれています。

AggregateIncompatibility オブジェクトは、ビジネスレイヤアイテムとデータベースの集計テーブル間の非互換性の関係を表します。 このオブジェクトを作成して、ビジネスレイヤにアグリゲート認識を定義します。

NavigationPath オブジェクトは、ビジネスレイヤのカスタムナビゲーションパスを表します。 ナビゲーションパスは、ドリル可能なビジネスオブジェクトの一覧です。SAP BusinessObjects Web Intelligence などのレポートツールのユーザは、ディメンションタイプのビジネスオブジェクトにドリルダウンできます。 ナビゲーションパスには、表示されるディメンションのみが含まれます。 ビジネスレイヤのカスタムナビゲーションパスを定義する NavigationPath オブジェクトを作成します。 デフォルトナビゲーションパスは、ビジネスレイヤ内でビジネスオブジェクトの階層構成によって定義されます。 これらは作成、編集、または削除することができません。 デフォルトナビゲーションパスを取得するには、オブジェクト階層をドリルスルーする必要があります。

4.5.4 ビジネスレイヤ項目

ビジネスレイヤは、BlItem オブジェクトのコレクションです。BlItem は、ビジネスレイヤの任意のオブジェクト (ディメンション、メジャー、属性、フォルダ、ビジネスフィルタ、およびネイティブフィルタ) を表します。

BusinessObject は、ビジネスレイヤの任意のメタデータオブジェクト (ディメンション、メジャー、および属性) を表します。これらのオブジェクトには、クエリでロールがあります。

BlContainer オブジェクトは、BlItem オブジェクトのコンテナと考えることもできる一部の BlItem オブジェクトのコンテナです。ディメンションおよびメジャーには属性があります。フォルダにはメジャーおよびディメンションが含まれます。

CustomProperty オブジェクトは、ビジネスレイヤとビジネスレイヤ項目 (ディメンション、メジャー、属性、フォルダ、およびフィルタ) のカスタムプロパティを定義します。このプロパティは、ビジネスレイヤ項目のソース情報 (技術情報、マッピング、およびリネージ) に依存しません。

BusinessFilter オブジェクトは、ビジネスレイヤのビジネスオブジェクトに基づいてフィルタを定義します。Filter オブジェクトから継承したメソッドを利用します。

NativeRelationalFilter オブジェクトは、リレーショナルビジネスレイヤに添付されているネイティブタイプの定義済みフィルタを定義します。ネイティブフィルタは、SQL 式を使用した場合にのみ定義できます。

RelationalBinding オブジェクトは、BusinessObject または NativeRelationalFilter の SQL 定義パーツ (SELECT、WHERE、および追加テーブル) を保持します。NativeRelationalFilter オブジェクトは、SELECT を使用しません。

4.5.5 表示書式

DisplayFormat オブジェクトは、SDK オブジェクトモデルにおけるビジネスオブジェクトの表示書式を表します。このリリースではデータ表示書式のみがサポートされています。DataFormat オブジェクトは、データ表示書式を表します。

> → 注意
>
> このオブジェクトモデルでは、その他の表示書式 (配置、罫線、網掛け、フォント) はサポートされていません。

このモデルでは、DataFormat から継承される DateTimeFormat および NumberFormat インタフェースを通して、ビジネスオブジェクトの日時と数値の表示書式がサポートされています。同様に、以下のインタフェースを通して、日時と数値のカスタム書式および定義済み書式がサポートされています。

DateTimeFormat オブジェクト	**NumberFormat** オブジェクト
CustomDateTimeFormat	CustomNumberFormat
PredefinedDateTimeFormat	PredefinedNumberFormat

日時と数値の定義済み書式は以下の列挙に定義されています。

- PredefinedDateTimeFormatType (SHORT_DATE、LONG_TIME など)
- PredefinedNumberFormatType (NUMERIC、PERCENT、SCIENTIFIC、および BOOLEAN)

ValueFormat オブジェクトは、カスタム書式の値を表します。以下の部分で構成されます。

- 書式を定義する文字列。デフォルトでは空の文字列が使用されます。
- 色 (FormatColor 列挙の値)。デフォルト値は、色が定義されていないことを意味する "Automatic" です。

> ⓘ 注記
>
> 表示書式の詳細については、インフォメーションデザインツールユーザガイドを参照してください。

関連情報

データ表示書式の使用 [53 ページ]

4.5.6 値の一覧

Lov オブジェクトは SDK オブジェクトモデルでの値の一覧を表しています。以下のタイプの値の一覧がサポートされています。

- StaticLov は、値を明示的に定義している一覧です。
- SQLQueryLov は、カスタム SQL 式に基づく値の一覧を表しています。
- BusinessQueryLov は、ビジネスオブジェクトで作成されたクエリを基にした値の一覧を表しています。

- BusinessHierarchicalLov は、ディメンションのカスタム階層を基にした値の一覧を表しています。

LovColumn オブジェクトは値の一覧の列を表しています。このモデルでは各タイプの値の一覧の列が定義されます。

- StaticLovColumn
- SQLQueryLovColumn
- BusinessQueryLovColumn
- BusinessHierarchicalLovColumn

StaticLovRow オブジェクトでは静的な値の一覧の行が定義されます。行には、すべての値の一覧の列に対して定義された値のセットが含まれます。

4.5.7 パラメータとプロンプト

Parameter オブジェクトは、データファンデーションおよびビジネスレイヤのユニバースパラメータを表します。

Answer オブジェクトはパラメータに渡される回答を表します。 次の種類の回答が使用できます。

- SingleValueAnswer は 1 つの値による回答を示しています
- IntervalAnswer は間隔で定義された値のセットを示しています
- MultipleValueAnswer は値のセットを示しています

TypedValue オブジェクトはパラメータへの回答として渡される値の一覧を示しています。 値はフラット (FlatValue) または階層 (HierarchicalValue) のいずれかになります。 階層値は平坦な値リストになります。 使用できる平坦な値は次のとおりです。

- 数値に対する NumericValue
- 文字列値に対する StringValue
- 日付に対する DateValue

4.5.8 プロパティ

Property インタフェースでは、SDK オブジェクトモデル内のカスタムプロパティおよびクエリスクリプトプロパティを管理します。カスタムプロパティは CustomProperty によって表され、クエリスクリプトプロパティは QueryScriptProperty によって表されます。

プロパティは {キー, 値} のペアとして定義されます。データファンデーションおよびビジネスレイヤの事前定義済みクエリスクリプトプロパティのキーは、DataFoundationQueryScriptPropertyKey および BusinessLayerQueryScriptPropertyKey で enum として指定されます。

DataFoundation および BusinessLayer インタフェースは、クエリスクリプトプロパティをサポートするために QueryScriptCustomizable を拡張します。

> → 注意
>
> クエリスクリプトプロパティは、インフォメーションデザインツールで指定されるクエリスクリプトパラメータです。詳細については、インフォメーションデザインツールユーザガイドを参照してください。

関連情報

事前定義済みクエリスクリプトプロパティのリファレンス [98 ページ]

4.5.9 セキュリティプロファイル

SecurityProfile オブジェクトは、ビジネスレイヤで定義されたセキュリティプロファイルを表します。

現在のリリースでは、データセキュリティプロファイルとビジネスセキュリティプロファイルの両方に対応しています。 DataSecurityProfile オブジェクトと BusinessSecurityProfile オブジェクトが用意されています。

データセキュリティプロファイル設定の範囲は、"行"、"テーブル"、および "接続" に制限されます。

ビジネスセキュリティプロファイル設定の範囲は、"クエリの作成" と "データの表示" に制限されます。

SecuredElements オブジェクトは、ビジネスセキュリティプロファイルで許可または拒否として定義されたビジネスレイヤアイテムおよびビジネスレイヤビューの一覧を表します。

4.5.9.1 ビジネスおよびデータセキュリティプロファイルの編集

以前は、セキュリティプロファイル設定を編集するためには、設定を 1 つずつ取得する必要がありました。現在は、1 つのメソッドですべてのデータおよびビジネスプロファイル設定を取得し、ローカルに編集してから、その新しい設定を公開できます。これまでのメソッドも引き続き使用できますが、この新しいアプローチにすると、パフォーマンスが大幅に向上します。

メソッド	アクション
getUniverseSecurityCache	これを使用すると、ユニバースのセキュリティプロファイル設定を取得し、設定をローカルに編集できます。
getBusinessSecurityProfiles	このメソッドを使用すると、ビジネスセキュリティプロファイルをローカルに取得できます。
getDataSecurityProfiles	このメソッドを使用すると、データセキュリティプロファイルをローカルに取得できます。
getPrincipals	このメソッドを使用すると、プリンシパル設定をローカルに取得できます。
getUniversePath	これは情報提供のみを目的としています。このメソッドを使用すると、ローカルに作業中のユニバースセキュリティ設定ファイルのユニバースパスを取得できます。
commit	このメソッドを使用すると、セキュリティプロファイル設定をユニバースに公開できます。

メソッド	アクション
close	UniverseSecurityCache は、いったん閉じると使用できなくなります。

4.5.10 接続

Connection はローカル接続およびセキュリティ接続を表すベースオブジェクトです (.cnx)。DatabaseConnection オブジェクトはローカル接続のみを表し、接続パラメータ情報を提供します。

RelationalConnection、OlapConnection、DataFederatorConnection オブジェクトは、すべて DatabaseConnection から継承され、それぞれ、リレーショナルデータソース、OLAP データソースへの接続、および各データフェデレーションサービスに基づいた接続を表します。

ConnectionShortcut は、セキュリティ接続へのローカル .cns ショートカットを表します。

4.6 SDK パッケージ

BI Semantic Layer Java SDK は、以下のパッケージで構成されています。

パッケージ	説明
com.sap.sl.sdk.authoring.businesslayer	ビジネスレイヤ、ビジネスレイヤアイテム、およびビジネスレイヤビューの操作に使用するインタフェース
com.sap.sl.sdk.authoring.checkintegrity	整合性チェックの実行に使用するインタフェース
com.sap.sl.sdk.authoring.cms	CMS リポジトリに保存されたユニバースリソースの操作に使用するインタフェース
com.sap.sl.sdk.authoring.commons	モデルのすべてのリソースオブジェクトのルートオブジェクトである SlResource オブジェクトを定義します。オブジェクトの識別、命名、カスタマイズ、および継承が可能なインタフェースが用意されています。SDK バージョン番号を取得するためのインタフェースが用意されています。
com.sap.sl.sdk.authoring.connection	接続および接続ショートカットの作成と管理に使用するインタフェース
com.sap.sl.sdk.authoring.datafoundation	データファンデーションとコンテンツの操作に使用するインタフェース (テーブル、結合、コンテキスト、1 次キー、値の一覧)
com.sap.sl.sdk.authoring.local	ローカルリソースの操作に使用するインタフェース

パッケージ	説明
`com.sap.sl.sdk.authoring.security`	データおよびビジネスセキュリティプロファイルの作成と操作に使用するインタフェース
`com.sap.sl.sdk.framework`	セマンティックレイヤ SDK の例外を扱うクラスとインタフェースを定義します
`com.sap.sl.sdk.framework.cms`	BI プラットフォーム Java SDK から CMS セッションを取得するために使用するインタフェース

パッケージの内容の詳細については、*SAP BusinessObjects BI Semantic Layer Java API reference* を参照してください。

com.sap.sl.sdk.authoring.commons パッケージ

名前、ID、およびカスタムプロパティは、ほぼすべての SDK オブジェクトに共通です。そのため、次の目的で新しいクラスが導入されました。

- コードをファクトライズし、SDK の実装を簡略化する
- これらの共通プロパティを一般的な方法で管理する

新しいクラスには、次のものがあります。

- オブジェクト ID を取得する `Identifiable`
- オブジェクト名を保存、取得、および設定する `Nameable`
- カスタムプロパティの一覧を保存および取得する `Customizable`
- コアユニバースのオブジェクトをリンクされたユニバースに継承することを許可する `Inheritable`

`Customizable` クラスを使用すると、カスタムプロパティのサポートをデータファンデーションとパラメータに拡張できます。

このパッケージには `SdkService` サービスも追加されています。

関連情報

BI セマンティックレイヤ Java SDK のイベントフロー [38 ページ]

4.7 SDK サービス

以下の表に、BI セマンティックレイヤ Java SDK が提供するサービスを示します。

サービス	説明
`BusinessLayerFactory`	ビジネスレイヤ、ビジネスレイヤ項目、ビジネスレイヤビュー、値の一覧、パラメータ、回答、カスタムおよびクエリスクリプトプロパティ、集計の非互換性、ナビゲーションパス、およびデータ表示書式の作成に使用されます。
`BusinessLayerService`	ビジネスレイヤ項目の暗黙のテーブルの取得に使用されます。SQL に基づく値の一覧の検出に使用されます。
`CmsResourceService`	セキュリティで保護されたユニバースリソースの管理に使用されます。
`CmsSecurityService`	リソースへのセキュリティプロファイルの割り当てに使用されます。
`CmsSessionService`	CMS セッションとそのコンテキストの取得に使用されます。
`ConnectionFactory`	接続の作成に使用されます。
`ConnectionService`	接続のテストおよび接続ドライバの変更に使用されます。
`DataFederatorService`	Data Federator クエリサーバに保存されている接続の管理に使用されます。複数ソース有効ユニバースによる作業で使用されます。
`DataFoundationFactory`	単一ソースのデータファンデーションと複数ソース対応のデータファンデーション、テーブル、列、結合、コンテキスト、1 次キー、ビュー、値の一覧、パラメータ、回答、カスタムおよびクエリスクリプトプロパティの作成に使用されます。
`DataFoundationService`	データファンデーションテーブルの完全修飾名の構築または分解に使用されます。データファンデーション構造の最新表示に使用されます。SQL に基づく値の一覧の検出に使用されます。
`LocalResourceService`	ローカルユニバースリソースの管理に使用されます。
`SdkService`	BI Semantic Layer Java SDK に共通のサービスの管理に使用されます。
`SecurityFactory`	データセキュリティプロファイルおよびビジネスセキュリティプロファイルの作成に使用されます。

4.8 ユニバースを操作するための SDK メソッド

以下のセクションでは、BI セマンティックレイヤ Java SDK のユニバースの操作に使用可能なメソッドの詳細を説明します。 詳細については、*SAP BusinessObjects BI* セマンティックレイヤ *Java API* 参照を参照してください。

メソッドの公開および取得

メソッド	説明
`CmsResourceService.publish`	ユニバースをリポジトリに公開または再公開します。 このメソッドは `.blx` ビジネスレイヤを引数として使用しており、`.unx` ユニバースを公開します。また、このメソッドには、`.blx` ビジネスレイヤ、`.dfx` データファンデーション、および `.cnx` 接続の完全パスが含まれます。
`CmsResourceService.retrieveUniverse`	セキュリティで保護されたユニバースを CMS リポジトリからユーザワークスペースに取得します。 このメソッドは `.unx` ユニバースを引数として使用しており、リソースのすべてをローカルに作成します。 リソースは、リレーショナル単一ソースユニバースと複数ソース有効ユニバースの場合は `.bfx`、`.dfx`、および `.cns` です。 OLAP ユニバースの場合は `.bfx` および `.cns` です。 作成されたリソースのオプション暗号化レベルが提供されます。
`CmsResourceService.createShortcut`	CMS に格納されているセキュリティで保護された接続 (`.cnx`) から、接続ショートカット (`.cns`) を作成します。
`LocalResourceService.publish`	ユーザワークスペースにリソースを公開します。 このメソッドは `.blx` ビジネスレイヤを引数として使用しており、`.unx` ユニバースをローカルに作成します。
`LocalResourceService.retrieve`	ローカル `.unx` からユニバースリソースを取得します。

メソッドのロードおよび保存

メソッド	説明
`CmsResourceService.loadConnection`	CMS のパスから接続をメモリにロードします。 セキュリティで保護された CMS への接続を更新するには、`CmsResourceService.saveConnection(DatabaseConnection)` を使用します。
`LocalResourceService.load`	任意の種類のローカルリソース (`.blx`、`.dfx`、`.cnx`、または `.cns`) をメモリにロードします。
`LocalResourceService.save`	リソース (`.blx`、`.dfx`、または `.cnx`) をファイルとしてローカルに保存します。

.unv から **.unx** への変換メソッド

メソッド	説明
`CmsResourceService.convertUniverse`	CMS に格納された `.unv` ユニバースを CMS の `.unx` ユニバースに変換します。

メソッド	説明
`LocalResourceService.convertUniverse`	ローカルの `.unv` ユニバースをローカルの `.unx` ユニバースに変換します。 このリリースでは、ローカルユニバースの変換に CMS セッションは必要ありません。

5 BI セマンティックレイヤ Java SDK のインストールおよびデプロイ

BI セマンティックレイヤ Java SDK は、SAP BusinessObjects Business Intelligence プラットフォーム 4.2 リリースの一部として提供されます。Windows、UNIX または Linux での SAP BusinessObjects Business Intelligence プラットフォームのインストールの実行時に、デフォルトでインストールされます。SAP BusinessObjects Business Intelligence プラットフォームクライアントツールのインストールに実行時にも、インストールすることができます。詳細については、*SAP BusinessObjects Business Intelligence* プラットフォームインストールガイド *(Windows* 版*)*と*(Unix* 版*)*を参照してください。

この SDK では、接続ドライバとの依存関係は作成されません。SDK のインストールとは別に、プラットフォームとともにドライバをインストールする必要があります。

BI セマンティックレイヤ Java SDK は Eclipse 環境またはスタンドアロンプロジェクトのいずれかにデプロイできます。

サポートされている設定は次のとおりです。

- Eclipse を使用する OSGI
- Eclipse を使用する非 OSGI
- Eclipse を使用しない非 OSGI

開発プロジェクトで BI セマンティックレイヤ Java SDK リソースをデプロイするには、以下の手順を実行します。サポートされている Eclipse のバージョンは 3.8.2 以上です。

以下の手順は Eclipse 3.8.2 Classic に適用されます。その他のバージョンの Eclipse で SDK をデプロイする場合は、手順が異なる可能性があります。

BI セマンティックレイヤ Java SDK をクライアントツールとともにインストールする [31 ページ]

インストールされた内容 [32 ページ]

Windows において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする [33 ページ]

UNIX または Linux において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする [34 ページ]

非 OSGI Eclipse 設定で BI セマンティックレイヤ Java SDK をデプロイする [34 ページ]

Eclipse OSGI フレームワーク設定で BI セマンティックレイヤ Java SDK をデプロイする [36 ページ]

5.1 BI セマンティックレイヤ Java SDK をクライアントツールとともにインストールする

> ⓘ 注記
>
> Microsoft Windows でのインストールでは、使用されるアカウントが Windows の Administrators グループのメンバーであること、Administrators グループに割り当てられるデフォルトの権限が変更されていないことが必要です。

1. SAP BusinessObjects Business Intelligence プラットフォームクライアントツールのインストールプログラムを検索します。
2. `setup.exe` ファイルを実行します。
3. インストールする機能の一覧を確認し、開発者用コンポーネントで *SAP BusinessObjects セマンティックレイヤ Java SDK* を選択します。
4. 開発プロジェクト用の SDK リソースとともにサンプルをインストールするには、サンプルをチェックしたままにします。 サンプルをインストールしない場合は、サンプルのチェックを解除します。
5. 残りのインストールステップを実行し、SDK リソースをマシンにインストールします。

SDK リソースを `C:¥Program Files (x86)¥SAP Business Objects¥SAP BusinessObjects Enterprise XI 4.0¥SL SDK` ディレクトリにインストールしました。

> **注記**
>
> BI クライアントツールインストーラのコマンドラインを使用して、SDK のサイレントインストールを実行することもできます。 詳細については、*SAP BusinessObjects Business Intelligence プラットフォームインストールガイド*を参照してください。

インストールの後、開発環境内で BI セマンティックレイヤ Java SDK をデプロイする必要があります。

5.2 インストールされた内容

Windows の場合

`<slsdk-install-dir>` ディレクトリにインストールされたリソースは、次のとおりです。

- `eclipse¥plugins` フォルダ。Eclipse での OSGI および非 OSGI デプロイメントに必要なすべてのプラグインと JAR リソースが含まれています。 このフォルダには、BOE Java SDK に必要なライブラリ (`com.businessobjects.boesdk`) を特別に埋め込みます。
  このフォルダには、Eclipse OSGI フレームワークで SDK をデプロイする場合にターゲットプラットフォームで選択する必要のない一連の Eclipse プラグインも含まれています。
- `java¥sl_sdk.jar`。これには、スタンドアロンデプロイメントの `CLASSPATH` を定義するマニフェストファイルが含まれています。
- `javadoc`。これには、ZIP ファイル (`sl_sdk_javadoc.zip`) として *SAP BusinessObjects BI Semantic Layer Java API reference* が含まれています。
- `SDK Samples` フォルダ。次のアーカイブが含まれています。
  - `com.sap.sl.sdk.authoring.samples.jar`
    テストに使用可能なサンプルのコンパイルされたバージョンが含まれています。
  - `com.sap.sl.sdk.authoring.samples.source.jar`
    これには、サンプルのソースコードが含まれています。

UNIX および Linux の場合

`<slsdk-install-dir>` ディレクトリにインストールされたリソースは、次のとおりです。

- eclipse/plugins フォルダ。JAR リソースが含まれています。 このフォルダには、BOE Java SDK に必要なライブラリ (com.businessobjects.boesdk) を特別に埋め込みます。
- java/sl_sdk.jar。これには、スタンドアロンデプロイメントの CLASSPATH を定義するマニフェストファイルが含まれています。
- javadoc
- SDK Samples フォルダ

5.3 Windows において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする

1. SDK リソースを使用して、<my_application.java> などのアプリケーションを開発します。
2. sl_sdk.jar ファイルで提供されたマニフェストファイルを使用して、CLASSPATH 環境変数を設定します。

   たとえば、次のコマンドを実行します。

   ```
   set CLASSPATH=<bip-install-dir>¥SAP BusinessObjects Enterprise XI 4.0¥SL
   SDK¥java¥sl_sdk.jar;
   ```

3. PATH 環境変数を、Connection Server バイナリディレクトリに設定します。

   たとえば、次のコマンドを実行します。

   ```
   set PATH=<bip-install-dir>¥SAP BusinessObjects Enterprise XI
   4.0¥win32_x86;%path%
   ```

4. 次のコマンドを実行して、プログラムをコンパイルします。

   ```
   javac my_application.java
   ```

5. 次のコマンドを実行して、プログラムを実行します。

   ```
   java -Dbusinessobjects.connectivity.directory="<bip-install-dir>¥SAP
   BusinessObjects
    Enterprise XI 4.0¥dataAccess¥connectionServer" my_application
   ```

   仮想マシンの引数 -Dbusinessobjects.connectivity.directory により、Connection Server のインストールディレクトリが指定されます。 これは、Connection Server との間で確立された接続を使用するために必要です。

アプリケーションで実装されたタスクを実行しました。

関連情報

このガイドの表記規則 [6 ページ]

5.4 UNIX または Linux において BI セマンティックレイヤ Java SDK をスタンドアロンでデプロイする

必要なサードパーティミドルウェアをインストールおよび設定しておきます。 LD_LIBRARY_PATH や PATH などの環境変数が、サードパーティミドルウェアパスとともに適切に設定されていることを確認します。

1. SDK リソースを使用して、<my_application.java> などのアプリケーションを開発します。
2. env.sh ファイルを使用して、SDK で使用される環境変数を設定します。

   このファイルは、<bip-install-dir>/setup フォルダにあります。 次のコマンドを実行します。

   ```
   source <bip-install-dir>/setup/env.sh
   ```

3. 次のコマンドを実行して、プログラムをコンパイルします。

   ```
   javac my_application.java
   ```

4. 次のコマンドを実行して、プログラムを実行します。

   ```
   java -Dbusinessobjects.connectivity.directory="<bip-install-dir>/
   enterprise_xi40/
   dataAccess/connectionServer" -jar my_application.jar
   ```

   仮想マシンの引数 -Dbusinessobjects.connectivity.directory により、Connection Server のインストールディレクトリが指定されます。 これは、Connection Server との間で確立された接続を使用するために必要です。

アプリケーションで実装されたタスクを実行しました。

関連情報

このガイドの表記規則 [6 ページ]

5.5 非 OSGI Eclipse 設定で BI セマンティックレイヤ Java SDK をデプロイする

この手順に従って、SDK 開発プロジェクトを作成、開発および実行できます。

1. PATH 環境変数を、Connection Server バイナリディレクトリに設定します。

   たとえば、次のコマンドを実行します。

   ```
   set PATH=<bip-install-dir>¥SAP BusinessObjects Enterprise XI
   4.0¥win32_x86;%path%
   ```

2. eclipse.exe ファイルをダブルクリックして Eclipse を起動し、ワークスペースを選択します。
3. Java プロジェクトの作成:

a. *File* > *New* > *Java Project* を選択します。
New Java Project ダイアログボックスが開きます。
b. プロジェクト名を入力し、プロジェクト JRE として *jre6* を選択して、*Finish* を選択します。
Package Explorer ビューにプロジェクトフォルダが表示されます。

4. SDK JAR ファイルのプロジェクトへの追加:
   a. プロジェクトを右クリックし、*Build Path* > *Configure Build Path* を選択します。
   Properties ダイアログボックスが開きます。
   b. *Libraries* タブを選択し、*Add External JARs* を選択します。
   c. `<slsdk-install-dir>¥java` ディレクトリに格納された `sl_sdk.jar` ファイルを参照し、*Open* を選択します。
   d. ユニットテストを実行する場合は、JUnit JAR ファイルを追加します。
      1. *Add Library* を選択します。
      2. JUnit を選択し、*Next* を選択します。
      3. JUnit バージョンを選択して、*Finish* を選択します。
5. 開発環境に javadoc を挿入します。
   a. ライブラリ一覧で `sl_sdk.jar` ノードを展開します。
   b. *Javadoc location* エントリを選択し、*Edit* を選択します。
   c. アーカイブパス `'C:¥Program Files¥SAP BusinessObjects¥SAP BusinessObjects Enterprise XI 4.0¥SL SDK¥javadoc¥sl_sdk_javadoc.zip'` を入力し、*OK* を選択します。
6. *OK* を選択して *Properties* ダイアログを閉じます。
   Package Explorer ビューで、該当プロジェクトの *Referenced Libraries* に SDK リソースパッケージが表示されます。
7. SDK パッケージを使用して、独自のアプリケーションを開発します。
8. 実行設定の作成:
   a. *Run* > *Run Configurations* を選択します。
   b. 実行設定を選択し、*Arguments* タブを選択します。
   c. *VM arguments* に以下の行を追加し、Connection Server をポイントします。

```
-Dbusinessobjects.connectivity.directory="<bip-install-dir>¥SAP
BusinessObjects Enterprise XI 4.0¥dataAccess¥connectionServer"
```

9. *Run* を選択してアプリケーションを実行します。
10. アプリケーションの結果をインフォメーションデザインツールで確認します。

アプリケーションで実装されたタスクを実行しました。

関連情報

このガイドの表記規則 [6 ページ]

5.6 Eclipse OSGI フレームワーク設定で BI セマンティックレイヤ Java SDK をデプロイする

この手順に従って、SDK 開発プロジェクトを作成、開発および実行できます。

1. PATH 環境変数を、Connection Server バイナリディレクトリに設定します。

   たとえば、次のコマンドを実行します。

   ```
   set PATH=<bip-install-dir>¥SAP BusinessObjects Enterprise XI
   4.0¥win32_x86;%path%
   ```

2. eclipse.exe ファイルをダブルクリックして Eclipse を起動し、ワークスペースを選択します。
3. SDK 開発のターゲットプラットフォームを作成します。
   a. *Window* > *Preferences* を選択し、*Plug-in Development* を展開して、*Target Platform* を選択します。
   b. *Add* を選択します。
      New Target Definition ウィンドウが表示されます。
   c. *Default: Default target for the running platform* にチェックを入れて、*Next* を選択します。.
   d. **SL SDK** などのターゲット名を入力します。
   e. *Locations* タブを選択し、*Add* を選択します。
      Add Content ウィンドウが表示されます。
   f. *Directory* を選択し、*Next* をクリックします。
   g. <slsdk-install-dir>¥eclipse¥plugins を参照し、*OK* を選択して、*Next*、*Finish* を選択します。
   h. *Locations* タブを選択し、*Add* を選択します。
      Add Content ウィンドウが表示されます。
   i. *Directory* を選択し、*Next* をクリックします。
   j. Eclipse 配布プラグインをターゲットプラットフォームで参照するには、*OK* を選択して、*Next* および *Finish* を選択します。
   k. *New Target Definition* ウィンドウで *Content* タブを選択し、*Group by* ドロップダウンメニューから *Location* を選択します。
   l. Eclipse 配布からすべてのプラグインのチェックを解除し、SDK プラグインとの競合を回避します。
   m. Eclipse 配布で、org.hamcrest.core および org.junit プラグインを選択します。
   n. *Finish* を選択し、ターゲットプラットフォームを選択して有効化します。
4. プラグインプロジェクトの作成:
   a. *File* > *New* > *Project* を選択します。
      New Project ダイアログボックスが表示されます。
   b. *Plug-in Development* を展開し、*Plug-in Project* を選択して、*Next* を選択します。
   c. プロジェクト名を入力し、*Target Platform* で Eclipse バージョン 3.4 を選択して、*Next* を選択します。
   d. 実行環境として Java 1.6 を選択し、プラグインに対する適切なオプションを選択して、*Finish* を選択します。

   Package Explorer ビューにプロジェクトフォルダが表示されます。
5. manifest.mf ファイルを開き、*Dependencies* をクリックして、以下のプラグインを *Required Plug-ins* 領域に追加します。
   - com.businessobjects.boesdk
   - com.sap.sl.sdk.authoring

> → ヒント
>
> また、ユニットテストを実行するために `org.junit` を追加することもできます。

6. SDK パッケージを使用して、独自のアプリケーションを開発します。
7. 実行設定の作成:
   a. *Run* > *Run Configurations* を選択します。
   b. 左ペインで、*JUnit Plug-in Test* をダブルクリックして現在のテストクラスの実行設定を作成します。
   c. *Main* タブで *Run an application* を選択し、*No Application - Headless Mode* または UI モードで実行するアプリケーションを選択します。

      実行時環境では JRE 1.6 を使用してください。
   d. *Arguments* タブで、*VM arguments* に以下の行を追加し、Connection Server をポイントします。

```
-Dbusinessobjects.connectivity.directory="<bip-install-dir>¥SAP
BusinessObjects Enterprise XI 4.0¥dataAccess¥connectionServer"
```

8. *Run* を選択してアプリケーションを実行します。
9. アプリケーションの結果をインフォメーションデザインツールで確認します。

アプリケーションで実装されたタスクを実行しました。

関連情報

このガイドの表記規則 [6 ページ]

6 BI Semantic Layer Java SDK の使用

この節では、BI Semantic Layer Java API の最適な使用方法についての詳細な技術情報と適切な実例を示します。

BI セマンティックレイヤ Java SDK のイベントフロー [38 ページ]

java.util.List メソッドの使用 [47 ページ]

デフォルト値の使用 [52 ページ]

データ表示書式の使用 [53 ページ]

データファンデーションビューの使用 [55 ページ]

リンクされたユニバースの使用 [56 ページ]

クエリスクリプトプロパティの使用 [59 ページ]

ビジネスフィルタ式とビジネスクエリの保存方法 [61 ページ]

データセキュリティプロファイルのテーブル設定の作業 [66 ページ]

ビジネスセキュリティプロファイルのセキュリティ保護された要素の操作 [66 ページ]

データファンデーションの構造の最新表示 [68 ページ]

整合性のチェックの実行 [68 ページ]

複数ソース有効ユニバースの使用 [69 ページ]

6.1 BI セマンティックレイヤ Java SDK のイベントフロー

次の手順に、セマンティックレイヤのオブジェクトを編集するすべての BI セマンティックレイヤ Java SDK サンプルに共通するイベントワークフローを示します。これは、SDK を使用するすべての開発プロジェクトに適用する必要がある基本的で不可欠なワークフローでもあります。

1. SDK コンテキストを作成します。
2. セッションが CMS リポジトリに作成され、コンテキストにアタッチされます。
3. コンテキストがサービスをインスタンス化します。
4. サービスを使用して必要なリソースがメモリにロードされます。
5. Java Bean で編集する場合と同様に、オブジェクトに対して操作が実行されます。
6. サービスを使用してリソースが保存されます。この手順で、ほとんどのセキュリティと整合性の制御が行われます。
7. サービスを使用してリソースが解放されます。
8. サービスを使用してコンテキストが解放されます。
9. セッションからログオフします。

セッションの作成 [39 ページ]

サービスのロード [39 ページ]

ファクトリの使用 [40 ページ]

リソースのインスタンス化 [43 ページ]
オブジェクトの保存 [43 ページ]
オブジェクトのチェック [44 ページ]
オブジェクトを解放する [46 ページ]
エラーの発生 [46 ページ]

6.1.1 セッションの作成

BI セマンティックレイヤ Java SDK が CMS へのアクセスを必要とするサービスをロードする場合は、事前に BI プラットフォーム Java SDK を使用してセッションが作成されます。BI セマンティックレイヤ Java SDK は、CMS 上のセッションを取得します。このセッションは、以下にアクセスできます。

- CMS にリンクされているリソースに対するすべてのサービスの実行
- CMS オブジェクト（ユーザーの接続権限に依存する）

例

1. まず、SDK コンテキストを作成します。

```
SlContext context = SlContext.create();
```

2. BI プラットフォーム Java SDK を使用して `IEnterpriseSession` オブジェクトを作成します。

```
IEnterpriseSession enterpriseSession =
CrystalEnterprise.getSessionMgr().logon(cmsUserName, cmsPassword, cmsHost,
cmsAuthMode);
```

3. `CmsSessionService` にセッションを登録して、SDK コンテキストにアタッチします。

```
context.getService(CmsSessionService.class).setSession(enterpriseSession);
```

4. すべてのサービスの実行が完了したら、`SlContext` オブジェクトを解放します。

```
context.close();
```

5. セッションをログオフします（コンテキストを閉じても自動的にログオフされません）。

```
enterpriseSession.logoff();
```

6.1.2 サービスのロード

`getService()` メソッドを使用すると、以下のようにサービスをロードすることができます。

```
<Service Class Name> service = context.getService(<Service Class Name>.class);
```

例

LocalResourceService をインスタンス化し、セッションを使用して LocalResourceService.class を取得できます。

```
LocalResourceService service = context.getService(LocalResourceService.class);
```

関連情報

SDK サービス [27 ページ]

6.1.3 ファクトリの使用

ほとんどのサービスは、Factory インタフェースによって作成できるオブジェクトを使用します。次の表に、このリリースで使用できる Factory インタフェースを示します。

インタフェース	説明
BusinessLayerFactory	次のメソッドを提供します。 • createRelationalBusinessLayer • createBlItem • createCustomProperty • createQueryScriptProperty • createAggregateIncompatibility • createNavigationPath • createBusinessLayerView • createSQLQueryLov • createStaticLov、createStaticLovColumn、および createStaticLovRow • createBusinessHierarchicalLov および createBusinessHierarchicalLovColumn • createBusinessQueryLov • createParameter • createSingleValueAnswer、createMultipleValueAnswer、および createIntervalAnswer • createHierarchicalValue • createNumericValue、createStringValue、および createDateValue • createPredefinedDateTimeFormat および createCustomDateTimeFormat • createPredefinedNumberFormat および createCustomNumberFormat
ConnectionFactory	次のメソッドを提供します。 • createRelationalConnection • createOlapConnection

インタフェース	説明
DataFoundationFactory	次のメソッドを提供します。 • createMonoSourceDataFoundation • createMultiSourceDataFoundation • createDatabaseTable • createDerivedTable • createAliasTable • createColumn • createCalculatedColumn • createSqlJoin • createContext • createDataFoundationView • createTableView • createDataFederatorSourceInfo • createPrimaryKey • createNativeRelationalFilter • createCustomProperty • createQueryScriptProperty • createSQLQueryLov • createStaticLov、createStaticLovColumn、および createStaticLovRow • createParameter • createSingleValueAnswer、createMultipleValueAnswer、および createIntervalAnswer • createHierarchicalValue • createNumericValue、createStringValue、および createDateValue
SecurityFactory	次のメソッドを提供します。 • createBusinessSecurityProfile • createDataSecurityProfile • createConnectionMapping • createRowRestriction • createTableMapping

例

SecurityFactory オブジェクトをサービスとしてインスタンス化して、SecurityFactory.class を取得できます。

```
final SecurityFactory securityFactory =
context.getService(SecurityFactory.class);
```

これで、インタフェースのメソッドを呼び出してタスクを実行できます。

```
DataSecurityProfile dataSecurityProfile =
securityFactory.createDataSecurityProfile();
```

6.1.4 リソースのインスタンス化

リソースに対してタスクを実行するメソッドを呼び出す前に、リソースをインスタンス化する必要があります。以下のメソッドを使用して、SlResource オブジェクトをインスタンス化することができます。

メソッド	実行できる操作
BusinessLayerFactory.createRelationalBusinessLayer(String, String)	メモリでの RelationalBusinessLayer オブジェクトの作成
DataFoundationFactory.createMonoSourceDataFoundation(String, String)	メモリでの MonoSourceDataFoundation オブジェクトの作成
DataFoundationFactory.createMultiSourceDataFoundation(String)	メモリでの MultiSourceDataFoundation オブジェクトの作成
LocalResourceService.load(String)	ユーザマシンにローカルに保存されているリソースのメモリでのロード。 リソースは、接続 (.cnx)、データファンデーション (.dfx)、またはビジネスレイヤ (.blx) です。DataFederatorConnectionShortcut オブジェクトなどの接続ショートカットをリソースとして使用することもできます。
CmsResourceService.loadConnection(String)	CMS リポジトリ内に保存されている接続のユーザマシンのメモリでのロード
ConnectionFactory.createRelationalConnection(String, String, String)	メモリでの RelationalConnection オブジェクトの作成
ConnectionFactory.createOlapConnection(String, String, String)	メモリでの OlapConnection オブジェクトの作成

6.1.5 オブジェクトの保存

接続（.cnx)、データファンデーション（.dfx)、ビジネスレイヤ（.blx）を編集したら、リソースを明示的に保存する必要があります。これは、リソースを公開する前に LocalResourceService.save(SlResource, String, boolean) メソッドを使用して行います。これにより、リソースはユーザー マシン上のファイルとして維持されます。自動保存機能はありません。

リソースの保存を実装しない場合、パブリケーションはメモリ内のオブジェクトからではなく、マシン上のリソース ファイルのパスからローカルに行われます。

接続を編集した後に、`CmsResourceService.saveConnection(String)` メソッドを使用して接続をリポジトリに保存することもできます。

> → 注意
>
> 構造を最新表示した後に、データファンデーションを明示的にロードしたり保存したりする必要はありません。この処理は `refreshStructure(String)` メソッドによって実行されます。

関連情報

データファンデーションの構造の最新表示 [68 ページ]

6.1.6 オブジェクトのチェック

4.1 SP5 より前にリリースされた SDK では、データファンデーションまたはビジネスレイヤの作成または編集時に一部でテストが実行されていました。 4.1 SP5 以降、リソースの保存時にすべての妥当性テストが実行されるようになりました。

次の表は、リソースの作成または編集時に発生したエラーの一覧です。現在、これらのエラーは保存時に検出されます。

リソース	エラー
データファンデーション	• 複数のテーブルに同じ修飾子名、所有者名、およびテーブル名が使用される。 • エイリアステーブルが、データベーステーブルや派生テーブル以外のデータファンデーションに属さないテーブルを参照する。 • 1 次キーが空白列または存在しない列を参照する。 • いくつかの列が、エイリアステーブルに含まれる。 • 派生テーブルの列またはデータベーステーブルに、同じ名前の列が含まれる。 • 2 つの結合に同じ式が含まれる。 • 結合で参照されるテーブルが存在しない。
複数ソース有効データファンデーション	• 無効な修飾子 • 接続がデプロイされない。 • 2 つの接続が同じショート名を共有している。 • 接続が CMS リポジトリ内の無効なパスを指定している。 • 複数ソース有効データファンデーションで使用される接続が CMS リポジトリ内に存在しない。

リソース	エラー
ビジネスレイヤ	• マスタビューが表示されず、他の表示ビューも存在されない。 • ビューに空白名が含まれている。 • 2 つのビューに同じ式が含まれている。 • ビューにサポートされていないオブジェクトが含まれている。 • ビューに重複するオブジェクトが含まれている。 • 同じフォルダ内で、2 つの同じ種類のオブジェクトが同じ名前を持っている。 • ビジネスレイヤ項目の名前が正しくない。 • 2 つのオブジェクトに同じ識別子が含まれている。 • オブジェクトタイプが不明です。 • ディメンションまたはメジャーに子として属性以外のオブジェクトがある。 • リレーショナルフィルタの WHERE 句が空になる。 • カスタムプロパティに空白キーがある。 • 同じオブジェクトの 2 つのカスタムプロパティが同じ名前である。
静的値の一覧	• 空白の列名 • 列が別の値の一覧から移動される • 重複列 • 自動参照 • 不明な列参照 • 値の一覧以外の行内でのその他の列
SQL クエリに基づく値の一覧	• 列が別の値の一覧から移動される • 自動参照 • 不明な列参照 • 不正なタイムアウト • 不正な最大行数 • 不正または NULL の SQL 式
階層型の値の一覧	• 階層に同じディメンションが 2 回含まれる。 • 不明なディメンション • 列が別の値の一覧から移動される • 不正なタイムアウト • 不正な最大行数
ビジネスクエリに基づく値の一覧	• 列が別の値の一覧から移動される
プロンプト	• 2 つのプロンプトに同じ名前が含まれる。 • プロンプトに空白名が含まれる。 • プロンプトに空白テキストが含まれる。 • 関連する値の一覧がサポートされていない。 • 関連付けられている値の一覧の列が表示されない。
プロンプト値	• 予期しないタイプ (フラット/階層) • 回答の値が NULL になる。 • 回答の値がプロンプトで予期されたものと同じタイプではない。

リソース	エラー
クエリ	• 警告: SQL クエリに基づく値の一覧の SQL 式が無効 • 警告: 派生テーブルの SQL 式が無効 • 警告: 計算列の SQL 式が無効 • 警告: 結合の SQL 式が無効 • 警告: AggregateIncompatibility の NULL テーブル名 • 警告: AggregateIncompatibility の不明なテーブル • 警告: AggregateIncompatibility の フォルダ • 警告: AggregateIncompatibility の不明な項目 • 例外: エイリアステーブルから存在しないテーブルを参照 • 例外: コンテキストを含むかコンテキストから除外する結合、または複数回含むか除外する結合 • 例外: コンテキストにデータファンデーションに属さない結合を含む • 例外: 欠落した接続 • 例外: 欠落したデータファンデーション • 例外: 見つからないか、または追加テーブルの重複

6.1.7 オブジェクトを解放する

`SlContext` に加えて、`SlResource` のオブジェクトも使用後に解放してメモリ リークを防止する必要があります。

以下のメソッドを使用して、対応するリソースを解放できます。

- `LocalResourceService.close(SlResource)`
- `CmsResourceService.close(SlResource)`
- `ConnectionFactory.close(SlResource)`

6.1.8 エラーの発生

BI Semantic Layer Java SDK の 4.0 FP3 および SP4 リリースには、各種カテゴリの例外を作成するためのメソッドが多数用意されています。4.0 SP5 リリース以降、4.1、および 4.2 リリースでは、API メソッドによってスローされる例外はすべて `SlException` オブジェクトとして管理されます。

エラーがワークフローのどの段階で発生しても、`SlException` が発生します。例外は以下の情報で構成されます。

- 3 つの文字と数字から成るコード (`SLS 10000` など)
- エラーメッセージ

各 `SlException` オブジェクトでは、`IStatus` オブジェクトを利用して、コード、メッセージ、重大度、原因などのエラー詳細へのアクセスを提供することができます。以下はその例です。

```
exception.getStatus().getCode();
```

6.2 java.util.List メソッドの使用

BI セマンティックレイヤ Java SDK では、データファンデーションやビジネスレイヤのオブジェクトで次のような多数のタスクを実行できます。

- データファンデーションテーブルの作成、追加、削除
- ビジネスレイヤアイテム (フォルダやビジネスオブジェクト) の作成、追加、移動、および削除
- 値の一覧およびパラメータの一覧の作成
- データファンデーションへの結合の追加、結合の取得および削除
- コンテキストへの結合の追加、結合の取得および削除
- 1 次キーへの列の追加、列の削除
- 静的な値の一覧への列の追加、列の削除
- 静的な値の一覧への行の追加、行の削除
- 静的な値の一覧への値の追加、行の削除
- 複数値が含まれる回答への値の追加
- 追加テーブルの取得、追加、および削除
- ビジネスセキュリティプロファイルへの許可および拒否されたオブジェクトの追加、オブジェクトの削除

SDK では SDK オブジェクトを作成するために特定のメソッドが提供されます。 たとえば、`DataFoundationFactory.createSqlJoin(String, String)` を使用して、`SQLJoin` オブジェクトを作成します。

その他の処理の場合、`java.util.List` メソッドを使用する必要があります。 オブジェクトモデルで、`BlContainer` オブジェクトは `BlItem` オブジェクトの一覧として実装されます。 `BlItem` オブジェクトに追加する、このオブジェクトを移動または削除するには、`BlContainer` インタフェースを使用します。

- `BlItem` オブジェクトを追加するには、このオブジェクトを一覧の最後または一覧の特定の位置に追加します。
- `BlItem` オブジェクトを削除するには、このオブジェクトを付加先の一覧から削除します。
- `BlItem` オブジェクトを移動するには、このオブジェクトを付加先の一覧から削除して別の一覧に追加するか、付加先の一覧内で位置を変更します。

一覧は次のルールを適用します。

- 厳密なオブジェクト関係: オブジェクトは 2 つの異なる一覧に属することができません。 1 つの一覧のみに保存する必要があります。 たとえば、結合が属することができるのは 1 つのデータファンデーション (`DataFoundation.getJoins()`) のみです。
- 低いオブジェクト関係: オブジェクトは複数の一覧に属することができます。 たとえば、結合は複数のコンテキスト (`Context.getIncludedJoins()`) に属することができます。
- 一覧には同一オブジェクトを 2 回含めることはできません。
- 一覧は並べ替えられます。

次の節では、BI セマンティックレイヤ Java SDK のオブジェクトで実行できるいくつかの操作について説明します。 標準的な `List` メソッドの説明は下記のリンクを参照してください。

テーブルの削除 [48 ページ]

ビジネス レイヤ アイテムの削除 [48 ページ]

ビジネスレイヤ項目の移動 [49 ページ]

結合の削除 [50 ページ]
追加テーブルの追加 [50 ページ]
コンテキストへの結合の追加 [51 ページ]
1 次キーからの列の削除 [51 ページ]
静的値の一覧の行への値の追加 [52 ページ]
カスタムナビゲーションパスへのディメンションの追加 [52 ページ]

関連情報

インタフェースリスト (Java Platform SE 6)

6.2.1 テーブルの削除

`DataFoundation.getTables()` と `java.util.List.remove()` メソッドのいずれかを使用して、データファンデーションテーブルを削除できます。

例

```
MonoSourceDataFoundation dataFoundation = ...;
Table table = ...;
dataFoundation.getTables().remove(table);
```

6.2.2 ビジネス レイヤ アイテムの削除

`BlContainer.getChildren()` と `Java.util.List.remove()` メソッドのいずれかを使用して、ビジネスレイヤアイテムを削除できます。

例

メジャーを取得するには、インデックス（`0`）と親を決定するか、

```
Folder folder = ...;
Measure measure = (Measure) folder.getChildren().get(0);
folder.getChildren().remove(measure);
```

BlItem.getParent() を使用して親を取得します。

```
measure.getParent().getChildren().remove(measure);
```

6.2.3 ビジネスレイヤ項目の移動

フォルダ間またはフォルダ内で、BlItem オブジェクトを移動することができます。

各項目は 1 つの親フォルダにのみ属すことができます。項目をコンテナに追加すると、その項目が属しているコンテナから自動的に削除されます。

同じコンテナ内で項目を移動する場合は、まずコンテナから項目を削除してから、再度コンテナに追加する必要があります。いずれかの java.util.List.add() メソッドを使用します。

例: フォルダ間でのメジャーの移動

一覧の末尾にメジャーを追加することができます。一覧にすでにメジャーが含まれている場合は、そのメジャーは変更されず、メソッドから false が返されます。

```
Folder folder1 = ...;
Folder folder2 = ...;
Measure measure = (Measure) folder1.getChildren().get(0);
folder2.getChildren().add(measure);
```

一覧の 3 の位置にメジャーを追加することができます。一覧にすでにメジャーが含まれている場合は、例外がスローされます。

```
folder2.getChildren().add(3, measure);
```

例: メジャーの位置の変更

一覧の末尾にメジャーを追加することができます。

```
Folder folder = ...;
Measure measure = (Measure) ...
folder.getChildren().remove(0);
folder.getChildren().add(measure);
```

一覧の 3 の位置にメジャーを追加することができます。

```
folder.getChildren().add(3, measure);
```

6.2.4 結合の削除

`DataFoundation.getJoins()` と `java.util.List.remove()` メソッドのいずれかを使用して、データファンデーションから結合を削除できます。

例

```
MonoSourceDataFoundation dataFoundation = ...;
SQLJoin join = (SQLJoin) ...
dataFoundation.getJoins().remove(0);
```

6.2.5 追加テーブルの追加

追加テーブルは、ビジネスオブジェクトに値を返すときに、SQL 式の結合によってクエリに含められます。 追加テーブルのリストは、インフォメーションデザインツールに表示されるテーブルのサブセットです。

`RelationalBinding.getExtraTables()` により、ビジネスオブジェクトの追加テーブルのリストを取得することができます。

このリストをメモリ内で取得した後に、`java.util.List.add()` メソッドのいずれかを使用してテーブルをリストの最後に追加できます。

次の書式設定ルールが `List` インタフェースメソッドに渡すことができる `qualifer.owner.table` 文字列に適用されます。

- 修飾子、所有者またはテーブル名に、ピリオド (`.`)、二重引用符 (`"`)、スペースが含まれる場合、引用符で囲む必要があります。
- 修飾子、所有者またはテーブル名内で二重引用符は二重にする必要があります。
- 二重引用符はバックスラッシュ ¥ を使用してエスケープする必要があります。

例

次の例は、`qualifier.owner.table` 文字列を `add()` メソッドの引数として渡すために必要な方法を示しています。 次のコードにより、`City` という追加テーブルをリストの末尾に追加します。

コード	説明
`getExtraTables().add("City")`	修飾子と所有者は null です。
`getExtraTables().add("Warehouse.dbo.City")`	修飾子は `Warehouse`、所有者は `dbo` です。

コード	説明
`getExtraTables().add("Warehouse."".City")`	修飾子は Warehouse、所有者はサポートされていません。
`getExtraTables().add("¥"Ware house¥".dbo.City")`	修飾子は "Ware house"、所有者は dbo です。
`getExtraTables().add("¥"Ware¥"¥"house¥".dbo.¥"Ci.ty¥"")`	修飾子は "Ware"house"、所有者は dbo で、テーブル名は Ci.ty です。

DataFoundationService.getTableFullName(String, String, String) を使用して、テーブルの完全修飾名を作成することができます。

6.2.6 コンテキストへの結合の追加

データファンデーションに結合を追加するとき、その結合はデータファンデーションのコンテキストには追加されません。結合を特定のコンテキストに明示的に追加する必要があります。

Context.getIncludedJoins() または Context.getExcludedJoins() メソッドを使用して、コンテキストに含まれる結合または除外される結合のリストを取得できます。

メモリからこのリストを取得すると、java.util.List.add() メソッドのいずれかを使用して、リストの末尾に結合を追加できます。

例

```
Context context = ...;
SQLJoin join = ...;
context.getExcludedJoins().add(join);
```

6.2.7 1 次キーからの列の削除

Key.getColumns() と java.util.List.remove() メソッドのいずれかを使用して、データファンデーションキーを構成する列のリストから列を削除できます。

例

```
DatabaseTable table = ...;
PrimaryKey key = (PrimaryKey) table.getPrimaryKey();
```

```
Column column = ...;
key.getColumns().remove(column);
```

6.2.8 静的値の一覧の行への値の追加

`StaticLovRow.getValues()`、および `java.util.List.add()` メソッドの 1 つを使用して、ビジネスレイヤの静的値の一覧の行に値を追加することができます。

例

```
StaticLovRow staticLovRow = ...;
staticLovRow.getValues().add(businessLayerFactory.createDateValue(Date.valueOf("2
014-02-03")));
```

6.2.9 カスタムナビゲーションパスへのディメンションの追加

`NavigationPath.getDimensions()`、および `java.util.List.add()` メソッドの 1 つを使用して、ビジネスレイヤのカスタムナビゲーションパスにディメンションを追加することができます。

例

```
BusinessLayer businessLayer = ... ;
NavigationPath navigationPath = ...;
Dimension dimension = (Dimension) businessLayerService.getBlItem(businessLayer,
"Dimperiod¥¥Date", true);
navigationPath.getDimensions().add(dimension);
```

6.3 デフォルト値の使用

SDK リソースは次のルールを適用します。

- データファンデーションの作成時、SQL オプションのデフォルト値を自動的に設定する。
- ビジネスレイヤの作成時、クエリオプションのデフォルト値を自動的に設定する。
- ビジネスオブジェクトの作成時、デフォルトの値の一覧に自動的に関連付ける。 デフォルトの値の一覧は、このオブジェクトのために計算された値の一覧です。

デフォルト値の詳細については、*SSAP BusinessObjects BI Semantic Layer Java API reference* を参照してください。

6.4 データ表示書式の使用

データ書式の作成

`BusinessLayerFactory` インタフェースの以下の "create" メソッドのいずれかを使用してデータ表示書式を設定し、関連するディメンション、メジャー、または属性に追加します。

- `createPredefinedDateTimeFormat(BusinessObject)`
- `createCustomDateTimeFormat(BusinessObject)`
- `createPredefinedNumberFormat(BusinessObject)`
- `createCustomNumberFormat(BusinessObject)`

例

```
Dimension requiredDateDimension = (Dimension)
businessLayerService.getBlItem(businessLayer, "CustOrder¥¥RequiredDate");
CustomNumberFormat customNumberFormat =
businessLayerFactory.createCustomNumberFormat(requiredDateDimension);
...
```

カスタムデータ書式の編集

以下のメソッドを使用します。

- `ValueFormat` インタフェースの `setFormat` および `setColor` メソッド (ビジネスオブジェクトのカスタムデータ書式を定義)
- `CustomDateTimeFormat` および `CustomNumberFormat` インタフェースのメソッド (`ValueFormat` オブジェクトに格納されている値を取得)

カスタムデータ書式は、`ValueFormat` 値が以下のルールに従っている場合にのみ有効です。

- PositiveFormat および DateTimeFormat は null にも空にもすることができません。
- NegativeFormat、ZeroFormat、UndefinedFormat の各値は null または空にすることができます。

複数のビジネスオブジェクトの書式を一括で変更することはできません。オブジェクトごとに明示的に変更する必要があります。

例

```
...
customNumberFormat.getPositiveFormat().setFormat("E+");
customNumberFormat.getPositiveFormat().setColor(FormatColor.BLUE);
customNumberFormat.getNegativeFormat().setFormat("#");
```

```
customNumberFormat.getNegativeFormat().setColor(FormatColor.RED);
customNumberFormat.getZeroFormat().setFormat("'nothing'");
customNumberFormat.getZeroFormat().setColor(FormatColor.AUTOMATIC);
customNumberFormat.getUndefinedFormat().setFormat("'??'");
customNumberFormat.getUndefinedFormat().setColor(FormatColor.BLACK);
```

文字列パターン

BI Semantic Layer Java SDK には、データ表示書式の文字列部分の作成に役立つ一連の数値および日時パターンが用意されています。

不適切な書式はありません。認識される文字列パターンは適切な値に置き換えられます。認識されないパターンは単なる文字列とみなされます。以下に例を示します。

- パターン `hour` は "*[1 から 12 の 1 桁または 2 桁で表される時間]*]ou*[元号]*" と解釈されます。
- パターン `xx/MM/yyyy` は "`xx`/*[01 から 12 の 2 桁で表される月]*/*[0000 から 9999 の 4 桁で表される年]*" と解釈されます。

文字列パターンの文字をエスケープするには、その文字を単一引用符で囲みます。したがって、単一引用符をエスケープする場合は、引用符を二重にします。以下に例を示します。

- パターン `'hour'` は "`hour`" と解釈されます。
- パターン `d'd'` は "*[1 から 31 の 1 桁または 2 桁で表される日付]*`d`" と解釈されます。
- パターン `''d''` は "`'`*[1 から 31 の 1 桁または 2 桁で表される日付]*`'`" と解釈されます。

書式に使用する色

`FormatColor` 列挙には、カスタムデータ書式に使用できる一連の色 (`Black`、`White`、`Red`、`Green`、`Blue`、`Yellow`、`Magenta`、`Cyan`、`Dark Red`、`Dark Green`、`Dark Blue`、`Dark Yellow`、`Dark Magenta`、`Dark Cyan`、`Light Gray`、`Gray`) が定義されています。

カスタム書式の一覧の取得

インフォメーションデザインツールには、カスタム書式の一覧が表示されます。BI Semantic Layer Java SDK では、ビジネスレイヤの各オブジェクトの表示書式を取得する前に、その書式がカスタム書式かどうかをチェックする必要があります。

関連情報

ValueFormat 文字列パターン参照 [100 ページ]

6.5 データファンデーションビューの使用

マスタビューの編集

マスタビューが、デフォルトのデータファンデーションビューです。 これには、データファンデーションのすべてのテーブルのビューが含まれています (1 つのテーブルに 1 つのビュー)。 マスタビューは作成や削除はできませんが、既存のビューを変更するために編集することができます。

データファンデーションに新しいテーブルが追加されると、マスタビューにすでに存在する場合を除いて、関連ビューが自動的にマスタビューに追加されます。 データファンデーションからテーブルが削除されると、すでに削除されている場合を除いて、関連ビューも自動的にマスタビューから削除されます。

> → 注意
>
> 次の変更はマスタビューの-保存時に取り消されます。
>
> - マスタビューからの既存のテーブルのビューの削除
> - マスタビューにすでにビューがあるテーブルに関連しているビューの追加

カスタムデータファンデーションビューの作成

`createDataFoundationView` メソッドを使用して、カスタムデータファンデーションビューを作成し、それを関連データファンデーションに追加します。

例

```
DataFoundation dataFoundation = ... ;
DataFoundationView dataFoundationView =
dataFoundationFactory.createDataFoundationView("MyView", dataFoundation);
dataFoundationView.setDescription("My description");
```

カスタムデータファンデーションビューへのテーブルの追加

カスタムデータファンデーションビューにテーブルを追加するには、次を行う必要があります。

- 追加対象のテーブルおよびターゲットビューに `TableView` オブジェクトのインスタンスを作成します。 `TableView` は、ビューでのテーブルの表示方法を定義するために使用されます。
- `TableView` のプロパティ (X 軸および Y 軸の位置、幅、表示状態) を設定します。

データファンデーションビューに応じて、さまざまな位置、幅、または表示状態をテーブルに設定することができます。

例

```
Table table = dataFoundation.getTables().get(0);
TableView tableView = dataFoundationFactory.createTableView(table,
dataFoundationView);
tableView.setX(123);
tableView.setY(456);
tableView.setTableState(TableState.JOINS_ONLY);
```

カスタムデータファンデーションビューへの連合テーブルの追加

ビューで連合テーブルを参照できるようにするために、FederatedTable クラスが API に追加されました。 このクラスは、Table を継承しています (名前、説明、列)。 連合テーブルを作成、編集、あるいは削除することはできませんが、データファンデーションテーブルのリスト内のある位置から別の位置へ連合テーブルを移動することができます。

このリリースでは、TableView オブジェクトを使用することによって、データファンデーションビューでの連合テーブルの追加および位置の設定を行うことができます。

例

```
FederatedTable table = ... ;
dataFoundation.getTables().remove(0);
dataFoundation.getTables().add(2, table); // moves the table to position 2
dataFoundation.getTables().get(2);
TableView tableView = dataFoundationFactory.createTableView(table,
dataFoundationView);
```

関連情報

オブジェクトの保存 [43 ページ]

6.6 リンクされたユニバースの使用

コアユニバースの取得

以下のメソッドを使用します。

- DataFoundation.getCoreUniverseReferences。データファンデーションによって参照されるコアユニバースの一覧を取得します。

- BusinessLayer.getCoreUniverseReferences。ビジネスレイヤによって参照されるコアユニバースの一覧を取得します。

コアユニバース情報の取得

コアユニバースの CUID、パス、および改訂番号は、リンクされたデータファンデーションまたはビジネスレイヤで参照情報として使用されます。この情報を取得するには、UniverseReference クラスの getCUID、getPath、および getRevisionNumber メソッドを使用します。

コアユニバースから継承されるオブジェクトの取得

データファンデーションおよびビジネスレイヤは、BlItem、Table、Join、Lov、および Parameter オブジェクトをコアユニバースから継承することができます。これらのオブジェクトは Inheritable インタフェースを実装するため、Inheritable.getInheritedData メソッドを使用して、それらのコアユニバースおよび識別子を取得することができます。

InheritedData クラスによって、以下のメソッドが提供されます。

- getCoreReference。オブジェクトの継承元のコアユニバースに関する参照情報を取得します。
- getCoreItemIdentifier。コアユニバース内のオブジェクトの識別子を取得します。

例

```
//BlItem
BlItem item = (BlItem) ...
String coreBlItemIdentifier = item.getInheritedData().getCoreItemIdentifier();
UniverseReference universeReference = item.getInheritedData().getCoreReference();
//Join
SQLJoin sqlJoin = (SQLJoin) ...
String coreJoinIdentifier = sqlJoin.getInheritedData().getCoreItemIdentifier();
UniverseReference universeReference =
sqlJoin.getInheritedData().getCoreReference();
```

リンクされたユニバースの作成および編集

BusinessLayerService インタフェースの以下のいずれかのメソッドを使用して、リンクされたユニバースを作成または編集します。

メソッド	実行できる操作
addCoreUniverses	リンクされたユニバースへのコアユニバースの追加。 コアユニバースの接続がデータファンデーションの接続と異なる場合は、警告が表示されます。
removeCoreUniverses	リンクされたユニバースからのコアユニバースの削除。
includeCoreUniverses	すべてのコアユニバースコンテンツの、リンクされたビジネスレイヤおよびその基になるデータファンデーションへの統合 (フォルダ、オブジェクト、テーブル、結合、ビューなど)。 コアユニバースへのリンクが削除され、そのオブジェクトはリンクされたユニバースの書き込み可能オブジェクトになります。コアユニバースの一覧からコアユニバースが削除されます。
synchronizeCoreUniverses	CMS リポジトリ内での、リンクされたユニバースのコアユニバースコンテンツの最新バージョンとの同期 (フォルダ、オブジェクト、テーブル、結合、ビューなど)。 同期とは、ローカルのリンクされたビジネスレイヤおよびデータファンデーションからのコアユニバースコンポーネントの更新、追加、または削除を意味します。コアユニバースは、その改訂番号が変更されている場合、つまり、これらのユニバースの新しいバージョンが公開されている場合に同期されます。新しいバージョンは、そのコアユニバースの前のバージョンであることも、後のバージョンであることもあります。BlItem オブジェクトのステータスは同期時に更新されません。
refreshUniverseReferences	コアユニバースのパスまたは名前が CMS リポジトリ内で移動または変更されている場合の、リンクされたユニバースでのそれらの更新。 **注記** インフォメーションデザインツールではこの機能は提供されません。

これらのメソッドは、コアユニバース識別子の一覧を表す引数として java.util.List<String> メソッドを指定するため、複数のコアユニバースを一度に管理することができます。受け入れられるユニバース識別子はユニバースのパスおよび CUID です。ユニバースのパスは /Universes/ で始まります。

これらのメソッドは、それ自身の操作を実行する前に、ビジネスレイヤおよびその基になるデータファンデーションをロードし、後でそれらを保存します。

> **注記**
>
> EditLinkedUniverseTest.java サンプルを使用して、リンクされたユニバースの作成および編集をテストすることができます。

リンクされたビジネスレイヤまたはデータファンデーションの保存

LocalResourceService.save メソッドを使用して、他のユニバースで実行する場合と同様に、リンクされたビジネスレイヤまたはデータファンデーションを保存します。

コアユニバースから継承されたコンテンツをリンクされたユニバースで変更することはできないため、コンテンツが更新されている場合、保存時にエラーが発生します。

`BlItem` オブジェクトのステータスのみを変更することができます。`isStatusOverloaded` メソッドを使用して、項目のステータスが、コアユニバースでの値と比べて変更されているかどうかを確認します。

`BlItem.setState(ItemState)` メソッドが使用されている場合、`isStatusOverloaded` 戻り値は自動的に `true` に設定されます。変更された `State` 値は保存時に保持されます。ビジネスレイヤのロード時に、CMS リポジトリ内のコアユニバースオブジェクトの `State` 値によって、以前に変更された値が上書きされることはありません。

リンクされたユニバースの公開

`CmsResourceService.publish` メソッドを使用して、他のユニバースで実行する場合と同様に、リンクされたビジネスレイヤまたはデータファンデーションを CMS リポジトリに公開します。

> → 注意
>
> リンクされたユニバースを公開する前に、コアユニバースを同期する必要があります。

関連情報

BI セマンティックレイヤ Java SDK サンプルによる開発 [74 ページ]

6.7 クエリスクリプトプロパティの使用

> → 注意
>
> クエリスクリプトプロパティは、インフォメーションデザインツールで指定されるクエリスクリプトパラメータです。

クエリスクリプトプロパティについて

クエリスクリプトプロパティは、以下のように提供されます。

- データファンデーションまたはビジネスレイヤで事前定義済みで、SDK に enum 値として付属します。
- ユーザにより作成され、データファンデーションまたはビジネスレイヤに動的に追加されます。

事前定義済みクエリスクリプトプロパティは、論理値、文字列、または整数 (正または負) のいずれかです。ただし、オブジェクトモデルでは {キー, 値} のペアとして定義され、この値は常に文字列として保存されるか、返されます。そのため、ご使用のコードで期待されるデータ型を文字列にキャストする必要があります。事前定義済みクエリスクリプトプロパティのリファレンス [98 ページ]でデータ型を参照してください。

事前定義済みクエリスクリプトプロパティは、ビジネスレイヤまたはデータファンデーションで必須の場合とオプションの場合とがあります。データファンデーションまたはビジネスレイヤが作成されると、すべての必須クエリスクリプトプロパティが自動的に追加され、削除することはできません。

クエリスクリプトプロパティの使用

メソッド	実行できる操作
• `BusinessLayerFactory.createQueryScriptProperty` • `DataFoundationFactory.createQueryScriptProperty`	オプションの事前定義済みクエリスクリプトプロパティまたは新しいプロパティ (カスタム) のビジネスレイヤまたはデータファンデーションへの追加。 このメソッドでは、クエリスクリプトプロパティがクエリスクリプトプロパティの一覧に追加されます。新しいクエリスクリプトプロパティのキーまたは名前は一意である必要があります。
`QueryScriptCustomizable.getQueryScriptProperties`	ビジネスレイヤまたはデータファンデーションの事前定義済みおよびカスタムクエリスクリプトプロパティの一覧の取得
`QueryScriptCustomizable.resetQueryScriptProperties`	一覧に対する以下のタスクの同時実行: 1. 一覧に追加されたオプションの事前定義済みおよびカスタムクエリスクリプトプロパティの削除 2. 必須の事前定義済みクエリスクリプトプロパティを一覧に残す 3. これらのプロパティのデフォルト値の設定
`Java.util.List.remove`	一覧からのクエリスクリプトプロパティの削除。 必須の事前定義済みクエリスクリプトプロパティを一覧から削除することはできません。
`Property.setKey`	オプションの事前定義済みまたはカスタムクエリスクリプトプロパティのキーの変更。 クエリスクリプトプロパティのキーまたは名前は一意である必要があります。
`Property.getKey`	オプションの事前定義済みまたはカスタムクエリスクリプトプロパティの取得
`Property.setValue`	事前定義済み (必須またはオプション) またはカスタムクエリスクリプトプロパティの値の変更
`Property.getValue`	事前定義済み (必須またはオプション) またはカスタムクエリスクリプトプロパティの値の取得

例

```
// Gets the predefined and custom query script properties of the data foundation
List<QueryScriptProperty> dataFoundationProperties =
dataFoundation.getQueryScriptProperties();
// Creates a custom query script property
```

```
QueryScriptProperty dataFoundationQueryScriptProperty =
dataFoundationFactory.createQueryScriptProperty("A_KEY", "A_VALUE",
dataFoundation);
// Sets a new value to the custom property
dataFoundationQueryScriptProperty.setValue("A_NEW_VALUE");

// Removes the custom property
dataFoundationProperties.remove(dataFoundationQueryScriptProperty);
```

ビジネスレイヤまたはデータファンデーションの保存

以下の場合は、保存時にエラーが発生します。

- 2 つのクエリスクリプトプロパティの名前が同じ場合。
- 必須のクエリスクリプトプロパティが一覧から削除された場合。

> **注記**
>
> 以下の 2 つの必須クエリスクリプトプロパティは、以前の 4.x リリースに追加されました。データファンデーションがユニバースデザインツールで作成されたユニバースから変換された場合などは、データファンデーションにこれらのプロパティが存在しないことがあります。
>
> - `REPLACE_COMMA_BY_CONCAT`
> - `USE_ENHANCED_QUERY_STRIPPING`
>
> BI セマンティックレイヤ Java SDK によってロードされたデータファンデーションにこれらのクエリスクリプトプロパティが存在しない場合は、それらのプロパティなしでデータファンデーションを保存することができます。

6.8 ビジネスフィルタ式とビジネスクエリの保存方法

ビジネスフィルタの式および値の一覧のビジネスクエリを作成または編集すると、フィルタ式またはビジネスクエリとして使用される XML 仕様が解析され、実際に保存される前に正しい形式に変更されます。式の意味は同じです。

XML 仕様は BI セマンティックレイヤ RESTful Web サービス SDK に含まれ、*Web Intelligence* および *BI* セマンティックレイヤ向け *SAP BusinessObjects RESTful Web* サービス *SDK* ユーザガイドで説明されています。以下の表で、この仕様の更新内容を説明します。

クエリオプション

要素	動作	式

`<queryOptions>`	• 以下のクエリオプションが存在しない場合に追加されます。
	  • `duplicatedRows`
	  • `maxRetrievalTimeInSeconds`
	  • `maxRowsRetrieved`
	  • `samplingResultSetSize`
	  • `samplingResultSetFixed`
	• 有効な属性を持たないクエリオプションがある場合、`activated="true"` が追加されます。
	• `samplingResultSetFixed` の `activated` および `value` 属性の値が異なる場合、これらがいずれも `false` に設定されます。
		ビジネスクエリ

例

```
<querySpecification version="1.0">
    <queryOptions>
        <queryOption name="duplicatedRows" value="true"/>
        <queryOption name="maxRetrievalTimeInSeconds" activated="false"
value="600"/>
        <queryOption name="maxRowsRetrieved" activated="true" value="5000"/>
        <queryOption name="samplingResultSetSize" activated="false" value="200">
        <queryOption name="samplingResultSetFixed" activated="false"
value="false">
    </queryOptions>
    ...
```

0、1、または 2 個の右オペランドを持つ比較フィルタ

要素	動作	式
`<answerValue>`	`<answerValue dataType="xxx">yyy</answerValue>` これが以下になります `<value>   <caption type="xxx">yyy</caption> </value>` • `dataType` 属性が指定されていない場合、`captiontype` 属性が String に設定されます。 • `Date` 型が `DateTime` になり、`T00:00:00.000Z` が値に追加されます。	• ビジネスフィルタ • ビジネスクエリ

例

```
<filterPart>
    <comparisonFilter operator="EqualTo" path="Time|folder¥Calendar|
folder¥CalendarYear|dimension"
        id="_IBo8FLIhEeCk0Ylv-tlF2Q">
        <constantOperand>
            <answerValue dataType="String">2011</answerValue>
        </constantOperand>
    </comparisonFilter>
</filterPart>
```

右オペランドに値の一覧を持つ比較フィルタ

要素	動作	式
`<value>` または `<level>` の `<caption>`	指定されていない場合、`type` 属性が `string` に設定されます。	• ビジネスフィルタ • ビジネスクエリ

例

```
<filterPart>
    <comparisonFilter operator="InList" id="_zPwYENK-EeSNS_-8mYpikg" path="Store
Names|folder¥Cascading_Associate_LOV_City|dimension">
        <constantOperand>
            <value>
                <caption type="String">Yancheng</caption>
```

```
            </value>
            <value>
                <caption type="String">London</caption>
                    <path>
                        <level>
                            <caption type="String">England</caption>
                        </level>
                        <level>
                            <caption type="String">London</caption>
                        </level>
                    </path>
            </value>
        </constantOperand>
    </comparisonFilter>
</filterPart>
```

パラメータを持つ比較フィルタ

要素	動作	式
`<parameter>`	属性 `constrained="true"`、`keepLastValues="true"`、`optional="false"`、および `promptWithLov="false"` が指定されていない場合、これらが設定されます。	• ビジネスフィルタ • ビジネスクエリ

例

```
<filterPart xmlns="http://www.sap.com/rws/sl/universe">
    <comparisonFilter id="_6zk08A-8EeO1RP--CtxScg" operator="EqualTo"
path="Custorder¥Orderid">
        <parameterOperand>
            <parameter constrained="true" keepLastValues="true" optional="false"
promptWithLov="true">
                <question>Enter Orderid</question>
            </parameter>
        </parameterOperand>
    </comparisonFilter>
</filterPart>
```

順位フィルタ

要素	動作	式
`<rankedByDimension>`	`path` 属性が削除されます。	• ビジネスフィルタ • ビジネスクエリ

例

```
<filterPart>
            <rankingFilter level="3" function="Top">
                <prompt>Enter ranking level :</prompt>
                <dimension path="Dimcustomer|folder¥Customer Name|dimension"
id="_7Zkd8A-8EeO1RP--CtxScg"/>
                <basedOnMeasure path="Custorderline|folder¥Quantity|measure"
id="_60Bg4g-8EeO1RP--CtxScg"/>
                <rankedByDimensions>
                    <rankedByDimension path="Dimcustomer|folder¥Regionname|
dimension" id="_60xHwQ-8EeO1RP--CtxScg"/>
                    <rankedByDimension path="Dimcustomer|folder¥Countryname|
dimension" id="_60xHwQ-8EeO1RP--CtxScg"/>
                </rankedByDimensions>
            </rankingFilter>
        </filterPart>
```

サブクエリフィルタ

要素	動作	式
<subQueryFilter>	指定されていない場合、correlationType 属性が None に設定されます。	• ビジネスフィルタ • ビジネスクエリ
<filterObject>	path 属性が削除されます。	

例

```
<filterPart>
    <subQueryFilter operator="EqualTo" correlationType="Any">
        <filterObjects>
            <filterObject path="Customer|folder¥Geography|folder¥Continent|
dimension"
            id="_IBo8M7IhEeCk0Ylv-tlF2Q"/>
            <filterObject path="Customer|folder¥Geography|folder¥Country|
dimension"
            id="_IBo8NrIhEeCk0Ylv-tlF2Q"/>
        </filterObjects>
        <queryData>
        <resultObjects>
            <resultObject path="Customer|folder¥Geography|folder¥Continent|
dimension"
            id="_IBo8M7IhEeCk0Ylv-tlF2Q"/>
            <resultObject path="Customer|folder¥Geography|folder¥Country|
dimension"
            id="_IBo8NrIhEeCk0Ylv-tlF2Q"/>
        </resultObjects>
        <filterPart>
        ...
```

6.9 データセキュリティプロファイルのテーブル設定の作業

`securityFactory.createDataSecurityProfile()` メソッドにより、空のデータセキュリティプロファイルを作成します。 `createRowRestriction`、`createTableMapping`、および `createConnectionMapping` メソッドを使用して、プロファイル設定 ("行"、"テーブル"、または "接続") を作成する必要があります。

"テーブル" の設定では、`setOriginalTable(String)` を使用して設定する元のテーブルは、データファンデーションに属するテーブルにする必要があります。

テーブル名は、以下のいずれかの名前になります。

- 修飾子または所有者のないテーブル名 (デフォルトの修飾子または所有者があるテーブルの場合、またはデータベースで修飾子または所有者がサポートされていない場合)。
- `"qualifier"."owner"."table"` の形式の完全修飾テーブル名。 データベースで修飾子または所有者がサポートされていない場合でも、パスする文字列にはすべての情報が含まれている必要があります。 たとえば、所有者がサポートされていない場合、文字列は `"qualifier"""."table"` にする必要があります。

`DataFoundationService.getTableName(String, String, String)` を使用して、テーブルの完全修飾名を作成することができます。 また、この文字列を `RowRestriction.setTable(String)` の引数として使用することもできます。

6.10 ビジネスセキュリティプロファイルのセキュリティ保護された要素の操作

`securityFactory.createBusinessSecurityProfile()` メソッドは空のビジネスセキュリティプロファイルを作成します。ビジネスレイヤのビューやアイテムを明示的にプロファイルに追加し、プロファイル設定を作成する必要があります。

ビジネスセキュリティプロファイルの設定は `SecuredElements` オブジェクトを使用して作成します。 `SecuredElements` オブジェクトは、プロファイルで許可または拒否されるビジネスレイヤビューまたはビジネスレイヤアイテムのリストを定義します。`SecuredElement` オブジェクトは、ビジネスレイヤのオブジェクトへのパスの文字列によって識別されます。

この文字列は次のルールに基づいて作成します。

- オブジェクトがビジネスレイヤビューの場合、名前で識別されます。マスタビューは空の文字列 (`""`) で識別されます。
- オブジェクトがビジネスレイヤアイテムの場合、文字列はオブジェクトの完全パスを表します。次のルールが適用されます。
  - パス内の各ノードはビジネスレイヤアイテム名とそのタイプで構成され、"<名前>|<タイプ>" のように `|` で区切ります。
  - `|` および ~ のエスケープ文字は ~ です。
  - パスのノードは `"Age Group|folder¥Age Max|dimension"` のように ¥ で区切ります。
  - ¥ および § のエスケープ文字は § です。

例

命名規則の例を次に示します。

Path	オブジェクト	String
Root folder	"CustomerName" ディメンション	"`CustomerName|dimension`"
"Customer" フォルダ	"Name" ディメンション	"`Customer|folder¥Name|dimension`"
"Contact" フォルダの "Customer" ディメンション	"Name" 属性	"`Contact|folder¥Customer|dimension¥Name|attribute`"
"Country|US" フォルダの "Customer¥Large" ディメンション	"First~Name" 属性	"`Country~|US|folder¥Customer§¥Large|dimension¥First~~Name|attribute`"

ビジネスセキュリティプロファイルでは次のタイプのビジネスオブジェクトを使用できます。

- `folder`
- `dimension`
- `measure`
- `attribute`
- `filter`

`BusinessSecurityProfile` および `SecuredElements` インタフェースで提供されるメソッドと、`java.util.List.add()` メソッドのいずれかを使用して、許可または拒否されるオブジェクトのリストをビジネスセキュリティプロファイルに追加できます。すべてのオブジェクトを許可または拒否に設定するには、`SecuredElements.setAllElementsStatus()` メソッドを使用できます。

例

```
//Creates an empty profile and adds its name
BusinessSecurityProfile bsp = securityFactory.createBusinessSecurityProfile();
bsp.setName(bspName);
//In the "Create Query" settings, adds all views of the business layer as
granted, except the master view that is set to Denied
bsp.getCreateQuerySecuredViews().setAllElementsStatus( SecurityStatus.Granted );
bsp.getCreateQuerySecuredViews().getDeniedElements().add ("");
//In the "Create Query" settings, adds all objects of the business layer as
granted, except the object CompanyName that is set to Denied
bsp.getCreateQuerySecuredObjects().setAllElementsStatus( SecurityStatus.Granted )
;
bsp.getCreateQuerySecuredObjects().getDeniedElements().add("Customer|
folder¥CompanyName|dimension");

//In the "Display Data" settings, adds all objects of the business layer as
granted, except the object CompanyName that is set to Denied
bsp.getDisplayDataSecuredObjects().setAllElementsStatus( SecurityStatus.Granted )
;
```

```
bsp.getDisplayDataSecuredObjects().getDeniedElements().add("Customer|
folder¥CompanyName|dimension");
```

6.11 データファンデーションの構造の最新表示

`DataFoundationService.refreshStructure(String)` を使用して、データベースのメタデータとデータファンデーションのメタデータを同期できます。

インフォメーションデザインツールと異なり、このメソッドでは更新するオブジェクトのタイプを選択できません。データベース内で変更されたすべてのオブジェクトをデータファンデーション内で最新表示します。

`refreshStructure(String)` メソッドは次の処理を実行します。

- データファンデーションのロード
- データファンデーション (結合、テーブルなど) の最新表示
- データファンデーションの保存

> → 注意
>
> このメソッドを実行する前にデータファンデーションをロードした場合、データファンデーションは更新されません。

6.12 整合性のチェックの実行

`LocalResourceService.createCheckIntegrityRunner(String)` を使用して、ローカルに保存するデータファンデーションまたはビジネスレイヤに、コンテキスト、結合、接続、パラメータ、および値の一覧に関する問題が含まれていないかを確認することができます。 また、このメソッドを使用して、接続へのパスを引数として渡すことによってローカル接続をチェックすることもできます。

インフォメーションデザインツールとは異なり、このメソッドではチェック対象のオブジェクトのタイプを選択できません。 ルールは同時に実行されます。

整合性チェックは、保存されているリソース上でのみ実行されます。 チェックにより、エラーの重要度、メッセージ、コード、および各ルールのステータスを含む `IStatus` オブジェクトが返されます。 ルールの結果ステータスは `RuleStatus` オブジェクトで管理されます。 そのため、`IStatus` オブジェクトは `RuleStatus` や `IStatus` オブジェクトなどから構成されるツリーとして表されます。

```
ERROR Check integrity // IStatus
-->OK Rule "Prompts validity rules" execution succeeded. // RuleStatus
-->OK Rule "Check Business Object Name " execution succeeded.
-->OK Rule "Query validity rules" execution succeeded.
-->OK Rule "Broken dependencies" execution succeeded.
-->OK Rule "Check Connection" execution succeeded.
-->OK Rule "LOV validity rules" execution succeeded.
-->OK Rule "Business filter validity" execution succeeded.
-->ERROR Business object validity (SQL/MDX) // RuleStatus
---->ERROR The business object 'Dimperiod¥Order Year' return data type is not
correct. (CIM 01404) // IStatus
java.lang.RuntimeException: Wrong data type: String
```

```
        at ...
        at com.sap.checkintegrity.core.AbstractRuleExecutor.execute(Unknown
Source)
        at
com.sap.checkintegrity.core.internal.IntegrityThread.runChecked(Unknown Source)
        at com.sap.checkintegrity.core.internal.IntegrityThread.run(Unknown
Source)
-->OK Rule "Business object validity (Binding)" execution succeeded.
```

処理に時間がかかりすぎる場合は、キャンセルすることができます。 キャンセル操作を実行する前に、現在のタスクの処理が完了します。 IStatus オブジェクトでは、操作をキャンセルする前に処理されたルールの結果のみが表示されます。

整合性のチェックでは、オブジェクト上で行われるすべてのテストは実行されません。 テストの大部分はオブジェクトの保存時に実行されます。

例

```
public void checkIntegrityOnLoadedObject() throws Exception {
    //...
    LocalResourceService localResourceService =
context.getService(LocalResourceService.class);
    final CheckIntegrityRunner checkIntegrityRunner =
localResourceService.createCheckIntegrityRunner("...");
        Thread thread = new Thread() {
            public void run() {
                IStatus status = checkIntegrityRunner.start();
                //...
            }
        };
        thread.start();
        //...
        checkIntegrityRunner.cancel();
    }
```

関連情報

オブジェクトのチェック [44 ページ]

6.13 複数ソース有効ユニバースの使用

単一ソースユニバースに適用される機能の大部分は、複数ソース有効ユニバースにも適用されますが、一部のワークフローは複数ソース有効ユニバースに固有です。

- 複数ソース有効ユニバースのデータソースライフサイクルの管理
- 複数ソース有効データファンデーションの作成
- 複数ソース有効データファンデーションからのデータソースの削除

- 複数ソース有効データファンデーションの派生テーブルの作成

接続ライフサイクルの管理 [70 ページ]
複数ソース有効データファンデーションを作成する [71 ページ]
複数ソース有効データファンデーションからのデータソースの削除 [71 ページ]
複数ソース有効データファンデーションの派生テーブルの作成 [72 ページ]

6.13.1 接続ライフサイクルの管理

複数ソース有効ユニバースの場合、`DataFederatorService` を使用して接続ライフサイクルを管理します。 複数ソース有効データファンデーションを作成するには、次の手順に従います。

1. `getCatalog(ConnectionShortcut)` を使用して、接続に関連するデータソースのカタログを取得します。
2. `deploy(ConnectionShortcut, String)` を使用して、データフェデレーション Query Server に接続を公開します。

> → 注意
>
> 複数ソース有効ユニバースに関連する任意の操作で接続を使用する場合、この手順は必須です。

また、このサービスでは次のメソッドも提供されます。

- `undeploy(String catalog)`
  これにより、データフェデレーションクエリサーバからユニバース接続が削除されます。
- `closeServerConnection()`
  これにより、データフェデレーションクエリサーバへの接続を閉じます。

> ⓘ 注記
>
> `SlContext.close()` を呼び出すと、サーバ接続は自動的に閉じられます。

例

```
//Instantiates the resource
ConnectionShortcut connectionShortcut = (ConnectionShortcut) ...;
//Loads the service in memory
DataFederatorService dataFederatorService =
context.getService(DataFederatorService.class);
//Retrieves the catalog
String catalog = dataFederatorService.getCatalog(connectionShortcut);
//Deploys the connection to the Query Server if no catalog is returned
if (catalog == null)
 catalog = ...;
 dataFederatorService.deploy(connectionShortcut, catalog);
```

関連情報

セッションの作成 [39 ページ]

6.13.2 複数ソース有効データファンデーションを作成する

DataFoundationFactory を使用して、CMS 上に複数ソース有効データファンデーションを作成します。次の手順に従います。

1. createMultiSourceDataFoundation(String) を使用して、接続がない複数ソース有効データファンデーションを作成します。
2. createDataFederatorSourceInfo(String connectionPath, String shortName, MultiSourceDataFoundation dataFoundation) を使用して DataFederatorSourceInfo オブジェクトを作成し、指定されたデータファンデーションにアタッチします。
   このオブジェクトは接続ショートカットへのパスを保持します。DataFederatorSourceInfo を作成することにより、データソースを複数ソース有効データファンデーションに追加しました。shortName 引数はデータファンデーション内の関連付けられた接続の一意の識別子です。

> ⚠ 警告
>
> DataFederatorSourceInfo オブジェクトを最初に作成していない場合、複数ソース有効データファンデーションを保存できません。

例

```
factory = context.getService(DataFoundationFactory.class);
MultiSourceDataFoundation dataFoundation =
factory.createMultiSourceDataFoundation("new dfo");
factory.createDataFederatorSourceInfo(connectionShortcut, "DFSourceInfo_1",
dataFoundation);
```

6.13.3 複数ソース有効データファンデーションからのデータソースの削除

MultisourceDataFoundation.getSourceInfo() と java.util.List.remove() メソッドのいずれかを使用して、複数ソース有効データファンデーションにアタッチされた接続のリストから接続を削除できます。このリストは並べ替えられていません。

例

```
MultiSourceDataFoundation dataFoundation = ...;
DataFederatorSourceInfo source = ...;
dataFoundation.getSourceInfo().remove(source);
```

`java.util.List.clear()` を使用して、データファンデーションからすべての接続を削除できます。

例

```
MultiSourceDataFoundation dataFoundation = ...;
dataFoundation.getSourceInfo().clear();
```

データファンデーションからデータソースを削除することにより、データソースに関連付けられているテーブル、結合、フィルタも削除されるわけではありません。データソースを削除した後、これらを明示的に削除する必要があります。

例

```
MultiSourceDataFoundation dataFoundation = ...;
DataFederatorSourceInfo source = ...;
dataFoundation.getSourceInfo().remove(source);
Table table = ...;
dataFoundation.getTables().remove(table);
SQLJoin join = (SQLJoin) ...
dataFoundation.getJoins().remove(join);
```

6.13.4 複数ソース有効データファンデーションの派生テーブルの作成

`DataFoundationFactory.createDerivedTable(String, String, DataFoundation)` を使用して、単一ソースまたは複数ソース有効データファンデーションの派生テーブルを作成できます。複数ソース有効データファンデーションの場合、このメソッドはデフォルトで ANSI SQL-92 標準構文をサポートする派生テーブルを作成します。

`DataFoundationFactory.createDerivedTable(String name, String sql, String shortName, MultiSourceDataFoundation dataFoundation)` を使用して、複数ソース有効データファンデーションのデータベース固有の派生テーブルを作成できます。`sql` 引数は指定されたデータソースの SQL 構文に準拠します。使用される特定のデータソースの名前は `shortName` 引数で渡されます。

`DerivedTable.getSourceShortName()` を使用して、派生テーブルの作成に使用されるデータソースのショート名を取得できます。

データファンデーションが単一ソースの場合、このメソッドは空の文字列を返します。複数ソース有効の場合は、次のようになります。

- 派生テーブルが複数データソースのテーブルから作成される場合、このメソッドは空の文字列を返します。
- 派生テーブルがデータベース固有の SQL 構文を使用する 1 つのデータソースのテーブルから作成される場合、このメソッドは `DataFederatorSourceInfo.getShortName()` の結果を返します。

`DerivedTable.setSourceShortName(String)` を使用して、派生テーブルの作成に使用されるデータソースを変更できます。

> ⚠ 警告
>
> SAP ではこのメソッドを使用してデータベース固有のテーブルを ANSI SQL-92 標準テーブルに変換するのではなく、ANSI SQL-92 テーブルを直接作成することをお勧めします。

7 BI セマンティックレイヤ Java SDK サンプルによる開発

サンプルは、BI セマンティックレイヤ Java SDK の使用方法を示すコードスニペットです。 サンプルにより、Java アプリケーションの実装方法を理解することができます。 Java 開発者として、これらのサンプルを基礎として使用すれば、独自のプログラミングを容易にし、促進することができます。 ブレークポイントをスニペットコードに置いて、閲覧することもできます。

これらのサンプルは、アーカイブ `<slsdk-install-dir>¥SDK Samples¥com.sap.sl.sdk.authoring.samples.source.jar` に提供されています。 このリリースでは、セルフドキュメント化されています。 サンプルコードファイルにおけるサンプルの動作についての説明を参照してください。

サンプルパッケージ [74 ページ]

サンプルを使用して開発する [75 ページ]

7.1 サンプルパッケージ

以下の表で、SDK サンプルについて説明します。

クラス	説明
`AssignProfileTest`	データセキュリティプロファイルをユーザまたはグループに割り当てたり、割り当てを解除したりします。
`ChangeUniverseConnectionsTest`	複数ソース有効ユニバースに追加されている複数の接続を CMS リポジトリ内の他のユニバースと置換します。
`ChangeUniverseConnectionTest`	単一ソースユニバースに追加されている接続を CMS リポジトリ内の他のユニバースと置換します。
`CreateBusinessSecurityProfileTest`	ビジネスセキュリティプロファイルを "クエリの作成" および "データの表示" 設定で作成し、このプロファイルをユニバースに追加します。
`CheckIntegrityTest`	ローカルに保存されたリソースで整合性のチェックを実行します。
`CreateUniverseTest` (4.2 SP3 で更新)	ユニバースのデータファンデーションとビジネスレイヤを作成し、CMS リポジトリに公開します。
`DataSecurityProfileTest`	データセキュリティプロファイルを "接続"、"行" および "テーブル" 設定で作成し、このプロファイルをユニバースに追加します。`CreateRowRestrictionTest` サンプルの代わりになります。

クラス	説明
`DisplayLinkedUniverseTest`	ユニバースを取得し、コアユニバースを持つかどうかをチェックします。
`EditLinkedUniverseTest` (4.2 SP3 で導入)	コアユニバースの機能 (追加、削除、最新表示、同期、および含める) を示すようにリンクされたユニバースを編集します。
`EditLocalBusinessLayerTest`	ローカルビジネスレイヤのビジネスレイヤ項目のフィールドを編集します。
`EditLocalUniverseTest`	接続名、データファンデーション、およびローカルユニバースのビジネスレイヤを編集したり、更新されたユニバースをローカルに公開したりします。
`GetUniverseConnectionPathsTest`	CMS リポジトリのユニバースに追加された、セキュリティで保護されている接続のパスを取得します。
`LovAndParameterTest` (4.2 SP3 で更新)	値の一覧を作成して、パラメータと関連付けます。
`PublishResourceTest`	ユニバースを CMS リポジトリに公開または再公開します。
`QueryScriptPropertyTest` (4.2 SP3 で導入)	データファンデーションおよびビジネスレイヤのクエリスクリプトプロパティを設定および取得します。
`RefreshStructureTest`	データベースメタデータとデータファンデーションを同期します。
`RetrieveResourceSaveForAllTest`	ユニバースとその関連接続をオプションの暗号化を使用して CMS リポジトリからローカルフォルダに取得します。
`UnvConversionTest`	`.unv` ユニバースを `.unx` ユニバースに変換します。
`UpdateConnectionTest`	CMS リポジトリの接続パラメータ (ユーザ、パスワードなど) を編集します。

ユーティリティパッケージ

`com.sap.sl.sdk.samples.util` パッケージには、以下のクラスが提供されています。

- `AuthenticationMode`: サンプルで使用される認証モードを定義します。
- `SamplesUtilities`: 主にファクトライズされたメソッドを提供して、サンプルコードを簡素化します。

7.2 サンプルを使用して開発する

1. このガイドで説明されている、いずれかのデプロイメント手順を実行します。
2. IDE へのサンプルソースアーカイブファイルのインポート:

a. プロジェクトフォルダ src を右クリックします。
b. インポート > 全般 > アーカイブファイル を選択します。
c. `<slsdk-install-dir>¥SDK samples¥com.sap.sl.sdk.authoring.samples.source.jar` を選択して [*OK*] を選択します。

これで、アーカイブファイルがプロジェクトソースフォルダにインポートされました。

3. 使用するサンプルを開き、サンプルで示される機能を埋め込むコードスニペットをコピーして、アプリケーションソースファイルに貼り付けます。

> → 注意
>
> パラメータの編集は、サンプルファイルに指定されているとおりに行ってください。

4. アプリケーションを開発します。

SDK サンプルを使用してアプリケーションを開発しました。

関連情報

非 OSGI Eclipse 設定で BI セマンティックレイヤ Java SDK をデプロイする [34 ページ]
Eclipse OSGI フレームワーク設定で BI セマンティックレイヤ Java SDK をデプロイする [36 ページ]

8 BI Semantic Layer Java SDK のトラブルシューティング

この節では、BI Semantic Layer Java SDK とそのソリューションで発生する可能性があるいくつかの問題について説明しています。

CreateProcess エラー [77 ページ]

csEx でサポートされない操作 [77 ページ]

ドライバ不明 [78 ページ]

無効なデータベーステーブル [78 ページ]

8.1 CreateProcess エラー

構文

```
Cannot run program "C:¥Program Files¥Java¥jre6¥bin¥Javaw.exe" (in directory
"C:¥Users¥Administrator¥SL SDK 4.1"): CreateProcess error=206, The filename or
extension is too long
```

Eclipse 3 で JUNIT テストを実行している場合に、このエラーメッセージが表示される場合があります。この問題を回避するには、Eclipse 4.2 を使用してください。

8.2 csEx でサポートされない操作

構文

```
java.lang.UnsupportedOperationException: csEX
```

このエラーメッセージが表示された場合は、Java 仮想マシンに次の引数が追加されていることを確認してください。

```
-Dbusinessobjects.connectivity.directory="D:¥SAP BusinessObjects¥SAP
BusinessObjects Enterprise XI 4.0¥dataAccess¥connectionServer"
```

8.3 ドライバ不明

構文

```
com.sap.sl.sdk.authoring.internal.AuthoringException: The connection driver
"xxxx" is unknown. (SLS 16000)
```

このエラーメッセージが表示されたら、以下の項目を確認してください。

- 32 ビット JRE をマシンにインストールしておくこと。64 ビット JRE はこのエラーの原因となります。
- Path 環境変数を <bip-install-dir>¥SAP BusinessObjects Enterprise XI 4.0¥win32_x86;%Path% に設定しておくこと。たとえば、以下のパスを設定します。

  ```
  set Path=D:¥SAP BusinessObjects¥SAP BusinessObjects Enterprise XI
  4.0¥win32_x86;%Path%
  ```

- createRelationalConnection メソッドに渡す DBMS およびネットワークレイヤの値が有効であること。各ネイティブ接続タイプの有効値については、「接続パラメータリファレンス」の章を参照してください。

8.4 無効なデータベーステーブル

構文

```
com.sap.sl.sdk.authoring.internal.AuthoringException: The database table is not
valid. (SLS 11004)
```

createDatabaseTable(String qualifier, String owner, String name, DataFoundation dataFoundation) メソッドを使用してデータベーステーブルを作成するときに、小文字の name を使用すると、このエラーメッセージが表示されることがあります。

小文字のユーザ名を使用して接続定義を作成すると、この操作では認証情報の大文字小文字が区別されないため、DBMS ではその接続でデータベースにアクセスすることができます。 しかし、テーブルを作成するときに小文字のユーザ名を指定すると、テーブルを作成するクエリは大文字小文字を区別するために問題が発生します。

この問題を解決するには、接続を定義するときとテーブルを作成するときに、大文字のユーザ名を使用します。

9 付録

接続パラメータリファレンス [79 ページ]
事前定義済みクエリスクリプトプロパティのリファレンス [98 ページ]
識別可能なオブジェクト参照 [99 ページ]
ValueFormat 文字列パターン参照 [100 ページ]

9.1 接続パラメータリファレンス

この節では、BI Semantic Layer Java SDK がサポートする接続と、この接続に関連付けられているユーザ権限について説明しています。

ユーザー権限について [79 ページ]
接続パラメータについて [80 ページ]
RDBMS 接続 [83 ページ]
SAP 接続 [92 ページ]
OLAP 接続 [96 ページ]

9.1.1 ユーザー権限について

BI セマンティックレイヤ Java SDK では、次のメソッドを使用して、接続および接続パラメータを更新できます。

- `CmsResourceService.loadConnection`
- `DatabaseConnection.getParameter`
- `ConnectionParameter.setValue`
- `ConnectionParameter.getValue`
- `CmsResourceService.saveConnection`

セキュリティ上の理由により、これらの操作は CMC の以下のユーザー権限によって保護されます。

- "Designer にログオンし CMC でこのオブジェクトを表示する"
- "オブジェクトの編集"
- "接続のローカル ダウンロード"
- “接続の作成、変更、または削除”

インフォメーション デザイン ツールにアタッチされている "CMS へのログイン" ユーザー権限は、サービスが実行される前の CMS セッションの作成時にチェックされます。

"オブジェクトの編集" および "接続のローカルダウンロード" は、CMS の接続オブジェクトにアタッチされているユーザー権限ですが、"接続の作成、変更、または削除" はインフォメーションデザインツールにアタッチされて

います。"接続のローカル ダウンロード" ユーザー権限は、リレーショナル接続にのみ関連します。Data Federator データ ソースと OLAP 接続では、これはサポートされません。

ユーザー権限は、CMC 内のユーザーまたはユーザーグループに割り当てられます。メソッドは、リソースに対する変更を保存する前に権限を検証します。

9.1.2 接続パラメータについて

セキュリティで保護された接続のほとんどのパラメータは、機密情報を含んでおり、CMS リポジトリに格納されている必要があります。 これらはプライベートパラメータと呼ばれます。 すなわち、これらのパラメータがリレーショナル接続で使用されるときには、"接続をローカルにダウンロード" 権限が付与されている場合にのみ、これらのパラメータを取得できます。 これは、`USER_NAME`、`DATASOURCE` などに当てはまります。 `loadConnection` メソッドは接続パラメータの完全な一覧を返します。ただし、この権限が拒否された場合には、プライベートパラメータ値は返されません。

`PASSWORD` は書き込み専用のプライベートパラメータです。 "接続をローカルにダウンロード" 権限が付与されている場合でも、CMS からこのパラメータの値を取得することはできません。 すなわち、`loadConnection` メソッドは `PASSWORD` の値なしで接続を返します。`getValue` メソッドをこのパラメータに使用することはできません。

CMS に格納される一部の接続パラメータはパブリックです。 "接続をローカルにダウンロード" 権限が拒否された場合でも、これらのパラメータを取得できます。 これは `AUTHENTICATION_MODE` に当てはまります。 `CONNECTIVITY_TYPE`、`DBMS`、および `NETWORK_LAYER` は、読み取り専用のパブリックパラメータです。 ただし、`changeDriver` メソッドで、これらのパラメータを間接的に変更できます。

接続を編集するには、"接続の作成、変更、または削除" 権限と "オブジェクトを編集する" 権限がユーザに付与されている必要があります。 これらの権限のうちの 1 つが拒否されると、`saveConnection` メソッドを使用できず、エラーメッセージが返されます。

接続パラメータの一覧 [80 ページ]

Connection Server 接続パラメータ [82 ページ]

Data Federator と OLAP 接続のパラメータ [83 ページ]

9.1.2.1 接続パラメータの一覧

次の表に、`DatabaseConnection.java` ファイルで定数値として定義されている接続パラメータの一覧と、各パラメータについてパブリックとプライベートの区別、読み取りと書き込みが可能かどうかを示します。 パラメータ名は定数値です。 この一覧には、リレーショナル、Data Federator、および OLAP 接続共通のほとんどのログインパラメータが含まれています。 SAP 接続パラメータも含まれています。

`DatabaseConnection.java` ファイルには、Apache Hadoop HIVE、IBM Informix、Oracle EBS などの接続の種類に固有の一部の接続パラメータに関する定数は含まれていません。

パラメータ名	アクセスの種類	読み取り可能	書き込み可能
`AUTO_RECONNECT`	プライベート	YES	YES

パラメータ名	アクセスの種類	読み取り可能	書き込み可能
AUTHENTICATION_MODE	パブリック	YES	YES
CLASS	プライベート	YES	YES
CONNECTIVITY_TYPE	パブリック	YES	changeDriver メソッドを除き NO
DATABASE	プライベート	YES	YES
DATASOURCE	プライベート	YES	YES
DBMS	パブリック	YES	changeDriver メソッドを除き NO
FETCH_SIZE	プライベート	YES	YES
LANGUAGE	プライベート	YES	YES
NETWORK_LAYER	パブリック	YES	changeDriver メソッドを除き NO
OLAP_CUBE	パブリック	YES	YES
OLAP_PATH	パブリック	YES	YES
OLAP_SERVER_NAME	パブリック	YES	YES
OLEDBPROVIDER	プライベート	YES	YES
PASSWORD	プライベート	NO	YES
PROVIDERSTRING	プライベート	YES	YES
SAP_APPLICATION_SERVER_NAME	プライベート	YES	YES
SAP_CLIENT_NUMBER	プライベート	YES	YES
SAP_GATEWAY_HOST_NAME	プライベート	YES	YES
SAP_GATEWAY_SERVICE_NAME	プライベート	YES	YES
SAP_GROUP_NAME	プライベート	YES	YES
SAP_HANA_HOST_NAME	プライベート	YES	YES
SAP_HANA_INSTANCE_NUMBER	プライベート	YES	YES

パラメータ名	アクセスの種類	読み取り可能	書き込み可能
SAP_HANA_SERVER_TYPE	プライベート	YES	YES
SAP_MESSAGE_SERVER_NAME	プライベート	YES	YES
SAP_PROGRAM_MAPPING	プライベート	YES	YES
SAP_SAVE_LANGUAGE	プライベート	YES	YES
SAP_SERVER_TYPE	プライベート	YES	YES
SAP_SSO_USE_SNC	プライベート	YES	YES
SAP_SYSTEM_ID	プライベート	YES	YES
SAP_SYSTEM_NUMBER	プライベート	YES	YES
SAP_USE_GATEWAY	プライベート	YES	YES
SAP_USE_PROGRAM_MAPPING	プライベート	YES	YES
USER_NAME	プライベート	YES	YES
USE_SSL	プライベート	YES	YES

9.1.2.2 Connection Server 接続パラメータ

この節では、Connection Server で管理される接続のみを対象としています。

以下の表は、CMC のユーザ権限に関連する接続更新のメソッドを使用する場合に適用される制限を説明しています。 メソッドを使用してパラメータを取得または設定できない場合は、例外がスローされます。 別途指定されない限り、これらのメソッドは、特定のパラメータに対して以下の権限で使用することができます。

	パブリックパラメータ	プライベートパラメータ
"オブジェクトを編集する": 許可 "接続をローカルにダウンロード": 許可	すべてのメソッドを使用できます。 パラメータが読み取り専用の場合: • saveConnection は使用できますが、パラメータ値は更新されません。 • setValue は使用できません。	すべてのメソッドを使用できます。

"オブジェクトを編集する": 拒否 "接続をローカルにダウンロード": 許可	編集が拒否されるため、`saveConnection` は使用できません。	編集が拒否されるため、`saveConnection` は使用できません。
"オブジェクトを編集する": 許可 "接続をローカルにダウンロード": 拒否	すべてのメソッドを使用できます。 パラメータが読み取り専用の場合: • `saveConnection` は使用できますが、パラメータ値は更新されません。 • `setValue` は使用できません。	`loadConnection` はすべてのパラメータを含む接続を返します。 `getValue` は空の文字列を返します。
"オブジェクトを編集する": 拒否 "接続をローカルにダウンロード": 拒否	`saveConnection` は使用できません。	`loadConnection` はすべてのパラメータを含む接続を返します。 `getValue` は空の文字列を返します。 `saveConnection` は使用できません。

9.1.2.3 Data Federator と OLAP 接続のパラメータ

このセクションでは、Data Federator データソースと OLAP 接続について説明します。 これらの接続では、"接続をローカルにダウンロード" 権限はサポートされません。

以下の制限は、CMC のユーザ権限に関連して接続更新のメソッドを使用する場合に適用されます。 メソッドを使用してパラメータを取得または設定できない場合は、例外がスローされます。 別途指定されない限り、これらのメソッドは、特定のパラメータに対して以下の権限で使用することができます。

	パブリックパラメータ	プライベートパラメータ
"オブジェクトを編集する": 許可	すべてのメソッドを使用できます。 パラメータが読み取り専用の場合: • `saveConnection` は使用できますが、パラメータは更新されません。 • `setValue` は使用できません。	すべてのメソッドを使用できます。
"オブジェクトを編集する": 拒否	`saveConnection` および `setValue` は使用できません。	`saveConnection` および `setValue` は使用できません。

9.1.3 RDBMS 接続

次の表は、Connection Server が使用される RDBMS 接続の各タイプでサポートされるパラメータの一覧です。 このパラメータ値を実装で使用して、データベースに接続できるようにします。

DB2 接続 [84 ページ]

JDBC 接続 [84 ページ]
ODBC 接続 [86 ページ]
OLE DB 接続 [88 ページ]
Oracle クライアント接続 [89 ページ]
SAP Sybase 接続 [90 ページ]
SAS 接続 [91 ページ]
テキストファイル接続 [91 ページ]

9.1.3.1 DB2 接続

次の表は、Connection Server を介した IBM DB2 データベースへの接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 値は以下のいずれかです。 • DB2 v9 • DB2 for z/OS v9 • DB2 10 for LUW • DB2 10.5 for LUW • DB2 10 for z/OS • DB2 UDB for iSeries v5 • DB2 for i v6 • DB2 for i v7
NETWORK_LAYER	値は IBM DB2 Client です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATASOURCE	エイリアス名 タイプは文字列です。

9.1.3.2 JDBC 接続

次の表は、Connection Server を介して確立される JDBC 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 値はデータベースに依存します。 下の表を参照してください。
NETWORK_LAYER	値は JDBC Drivers です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO (サポートされている場合)
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATABASE	デフォルトのデータベース名。 タイプは文字列です。 必要としないデータソースもあります。 Oracle データベース向けのネットサービスとして使用されます。
DATASOURCE	タイプは文字列です。 形式は <サーバホスト>:<ポート> です。 Generic JDBC datasource のみの JDBC URL 文字列も表します。
CLASS	JDBC ドライバクラスです。 タイプは文字列です。 Generic JDBC datasource にのみ必要です。

次の表に、データベースベンダ別の使用可能な DBMS パラメータ値を示します。

データベースベンダ	DBMS
Apache	• Apache Hadoop HIVE • Amazon EMR HIVE • Derby 10 Embedded
汎用	汎用 JDBC データソース
GreenPlum	GreenPlum 4
Hewlett Packard	• HP Neoview • HP Vertica 6.1
HSQLDB	HSQLDB 1.8 Embedded
IBM	• DB2 v9 • DB2 10 for z/OS • DB2 10 for LUW • DB2 10.5 for LUW • Informix Dynamic Server 11

データベースベンダ	DBMS
Ingres	Ingres Database 9
Microsoft	• MS SQL Server 2008 • MS SQL Server 2012
Netezza	• Netezza Server 4 • Netezza Server 5 • Netezza Server 6 • Netezza Server 7
Oracle	• MySQL 5 • Oracle 10 • Oracle 11 • Oracle Exadata
PostgreSQL	• PostgreSQL 8 • PostgreSQL 9
SAP	• MaxDB 7.7 • SAP HANA データベース 1.0
SAP Sybase	• Sybase Adaptive Server Enterprise 15.5 • Sybase IQ 15 • Sybase IQ 16 • Sybase SQL Anywhere 11 • Sybase SQL Anywhere 12 • Sybase SQL Anywhere 16
Teradata Corporation	• Teradata 12 • Teradata 13 • Teradata 14 • Teradata 14.1

9.1.3.3 ODBC 接続

次の表は、Connection Server を介して確立される ODBC 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 値はデータベースに依存します。 下の表を参照してください。
NETWORK_LAYER	値は ODBC Drivers です。

パラメータ名	説明
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO (サポートされている場合)
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATABASE	デフォルトのデータベース名。 タイプは文字列です。 必要としないデータソースもあります。
DATASOURCE	タイプは文字列です。 値はデータソース名です (DSN)。

次の表に、データベースベンダ別の使用可能な DBMS パラメータ値を示します。

データベースベンダ	DBMS
汎用	• 汎用 ODBC データソース • 汎用 ODBC 3 データソース
GreenPlum	GreenPlum 4
Hewlett Packard	• HP Neoview • HP Vertica 6.1
IBM	• DB2 UDB for iSeries v5 • DB2 for i v6 • DB2 for i v7 • Informix Dynamic Server 11
Ingres	Ingres Database 9
Microsoft	• MS Access 2007 • MS Access 2010 • MS Access 2013 • MS Excel 2007 • MS Excel 2010 • MS Excel 2013 • MS SQL Server 2008 • MS SQL Server 2012 • テキストファイル

データベースベンダ	DBMS
Netezza	• Netezza Server 4 • Netezza Server 5 • Netezza Server 6 • Netezza Server 7
Oracle	MySQL 5
PostgreSQL	• PostgreSQL 8 • PostgreSQL 9
salesforce.com	OpenAccess for Salesforce
SAP	• MaxDB 7.7 • SAP HANA データベース 1.0
SAP Sybase	• Sybase IQ 15 • Sybase IQ 16 • Sybase SQL Anywhere 11 • Sybase SQL Anywhere 12 • Sybase SQL Anywhere 16
Teradata Corporation	• Teradata 12 • Teradata 13 • Teradata 14 • Teradata 14.1

9.1.3.4 OLE DB 接続

次の表は、Connection Server を介して確立される OLE DB 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 値はデータベースに依存します。 下の表を参照してください。
NETWORK_LAYER	値は OLE DB Providers です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO (サポートされている場合)

パラメータ名	説明
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATABASE	デフォルトのデータベース名。 タイプは文字列です。
DATASOURCE	サーバ名。 タイプは文字列です。
OLEDBPROVIDER	OLE DB プロバイダ名。 タイプは文字列です。
PROVIDERSTRING	OLE DB プロバイダ文字列。

次の表に、データベースベンダ別の使用可能な DBMS パラメータ値を示します。

データベースベンダ	DBMS
汎用	汎用 OLEDB プロバイダ
Microsoft	• MS SQL Server 2008 • MS SQL Server 2012

9.1.3.5 Oracle クライアント接続

次の表は、Connection Server を介して確立される Oracle OCI 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 使用可能な値は以下のとおりです。 • Oracle 10 • Oracle 11 • Oracle EBS • Oracle Exadata
NETWORK_LAYER	値は Oracle Client です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO
USER_NAME	タイプは文字列です。

パラメータ名	説明
PASSWORD	タイプは文字列です。
DATASOURCE	タイプは文字列です。 値は Oracle SID または Oracle 接続文字列のいずれかです。

例: データソース

以下の例の場合、データソース値は ORA11 または ORA11 と同等の式のいずれかです。

```
ORA11 =
 (DESCRIPTION =
   (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
   )
   (CONNECT_DATA =
      (SERVICE_NAME = ORA11)
   )
 )
```

9.1.3.6 SAP Sybase 接続

次の表は、Connection Server を介した SAP Sybase データベースへの接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	タイプは文字列です。 値は Sybase Adaptive Server Enterprise 15.5 です。
NETWORK_LAYER	値は Sybase Open Client です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATABASE	デフォルトのデータベース名。 タイプは文字列です。
DATASOURCE	タイプは文字列です。 形式は <サーバホスト>:<ポート> です。

9.1.3.7 SAS 接続

次の表は、Data Federator コネクタを介した SAS データベースへの接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	値は Data Federator Server XI R4 です。
NETWORK_LAYER	値は JDBC Drivers です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。

9.1.3.8 テキストファイル接続

次の表は、Connection Server を介したテキストファイルへの接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は open です。
DBMS	値は Text Files です。
NETWORK_LAYER	値は BO OC です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
DATASOURCE	タイプは文字列です。 値は UNIX 形式または MS Windows 形式のローカルファイルパスか、HTTP、FTP、および SMB の各データソースの場合は URI です。

9.1.4 SAP 接続

次の表は、以下の SAP データソースへの接続を記述するパラメータの一覧です。

- Connection Server を介した SAP ERP システム

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	型は文字列です。 値は以下のいずれかです。 • mySAP ERP 2004 • SAP ERP 6 • SAP R/3 Release 4
NETWORK_LAYER	値は SAP Java Connector (SAP JCo) です。
AUTHENTICATION_MODE	型は列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO
USER_NAME	型は文字列です。
PASSWORD	型は文字列です。
LANGUAGE	ロケール (例: en_US)。
SAP_SSO_USE_SNC	値は True または False です。
SAP_SAVE_LANGUAGE	値は True または False です。
SAP_CLIENT_NUMBER	3 つの整数で構成されます。 型は文字列です。
SAP_SYSTEM_ID	3 つの文字で構成されます。 型は文字列です。
SAP_SERVER_TYPE	型はは列挙です。 次の値を指定できます。 • APPLICATION_SERVER • MESSAGE_SERVER
SAP_SYSTEM_NUMBER	2 つの文字で構成されます。 型は文字列です。
SAP_APPLICATION_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • アプリケーションサーバ名 (サーバタイプが APPLICATION_SERVER の場合) • 空の文字列 (それ以外の場合)

パラメータ名	説明
SAP_MESSAGE_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • メッセージサーバ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)
SAP_GROUP_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • グループ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)

- SAP HANA データソース

パラメータ名	説明
CONNECTIVITY_TYPE	値は `relational` または `Olap` です。
DBMS	型は文字列です。 値は `SAP HANA database 1.0` です。
NETWORK_LAYER	使用可能な値は、以下のとおりです。 • `JDBC Drivers` • `ODBC Drivers` • `SAP Hana Client` (OLAP の場合)
AUTHENTICATION_MODE	型は列挙です。 使用可能な値は、以下のとおりです。 • `FIXED` • `MAPPING` • `SSO`
USER_NAME	型は文字列です。
PASSWORD	型は文字列です。
DATASOURCE	型は文字列です。 以下の接続に使用されます。 • ODBC 接続 • 複数のサーバを含む JDBC 接続
SAP_HANA_SERVER_TYPE	型は列挙です。 JDBC 接続に使用されます。 使用可能な値は、以下のとおりです。 • `SINGLE_SERVER` • `MULTI_SERVER`
SAP_HANA_HOST_NAME	型は文字列です。 JDBC 接続および OLAP 接続に使用されます。
SAP_HANA_INSTANCE_NUMBER	2 つの数字で構成されます。 型は文字列です。 JDBC 接続および OLAP 接続に使用されます。
USE_SSL	型は論理値です。 JDBC 接続および OLAP 接続に使用されます。

パラメータ名	説明
LANGUAGE	ロケール (例: en_US)。 OLAP 接続に使用されます。
AUTO_RECONNECT	型は論理値です。 OLAP 接続に使用されます。
FETCH_SIZE	型は整数です。 OLAP 接続に使用されます。

- Data Federator コネクタを介した SAP BW

パラメータ名	説明
CONNECTIVITY_TYPE	値は relational です。
DBMS	値は Data Federator Server XI R4 です。
NETWORK_LAYER	値は SAP Java Connector (SAP JCo) です。
AUTHENTICATION_MODE	型は列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO
USER_NAME	型は文字列です。
PASSWORD	型は文字列です。
LANGUAGE	言語 (例: EN)。
SAP_SSO_USE_SNC	値は True または False です。
SAP_SAVE_LANGUAGE	値は True または False です。
SAP_CLIENT_NUMBER	3 つの整数で構成されます。 型は文字列です。 3 つの整数で構成されます。
SAP_SYSTEM_ID	3 つの文字で構成されます。 型は文字列です。
SAP_SERVER_TYPE	型は列挙です。 使用可能な値は APPLICATION_SERVER または MESSAGE_SERVER です。
SAP_SYSTEM_NUMBER	2 つの整数で構成されます。 型は文字列です。
SAP_APPLICATION_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • アプリケーションサーバ名 (サーバタイプが APPLICATION_SERVER の場合) • 空の文字列 (それ以外の場合)

パラメータ名	説明
SAP_MESSAGE_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • メッセージサーバ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)
SAP_GROUP_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • グループ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)
SAP_USE_GATEWAY	型は論理値です。
SAP_GATEWAY_HOST_NAME	SAP BW ゲートウェイをホストしているマシンの名前。 型は文字列です。
SAP_GATEWAY_SERVICE_NAME	SAP BW ゲートウェイサービスの名称またはポート番号。 型は文字列です。
SAP_USE_PROGRAM_MAPPING	型は論理値です。
SAP_PROGRAM_MAPPING	SAP BW が Data Federator との接続に使用するコールバックのプログラム ID。 型は文字列です。
OLAP_CUBE	インフォプロバイダ名。

- SAP BICS クライアントミドルウェアドライバを介した SAP BW 7.x 接続

パラメータ名	説明
CONNECTIVITY_TYPE	値は Olap です。
DBMS	値は SAP Netweaver BW 7.x です。
NETWORK_LAYER	値は SAP BICS Client です。
AUTHENTICATION_MODE	型は列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO • PROMPTED
USER_NAME	型は文字列です。
PASSWORD	型は文字列です。
SAP_CLIENT_NUMBER	3 つの整数で構成されます。 型は文字列です。
LANGUAGE	ロケール (例: en_US)。
SAP_SAVE_LANGUAGE	値は True または False です。

パラメータ名	説明
SAP_SYSTEM_ID	3 つの文字で構成されます。 型は文字列です。
SAP_SERVER_TYPE	型は列挙です。 使用可能な値は、以下のとおりです。 • APPLICATION_SERVER • MESSAGE_SERVER
SAP_SYSTEM_NUMBER	2 つの文字で構成されます。 型は文字列です。
SAP_APPLICATION_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • アプリケーションサーバ名 (サーバタイプが APPLICATION_SERVER の場合) • 空の文字列 (それ以外の場合)
SAP_MESSAGE_SERVER_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • メッセージサーバ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)
SAP_GROUP_NAME	型は文字列です。 使用可能な値は、以下のとおりです。 • グループ名 (サーバタイプが MESSAGE_SERVER の場合) • 空の文字列 (それ以外の場合)
OLAP_PATH	値はデータソースの種類に依存します。 • データソースがサーバの場合は空の文字列 • データソースがインフォプロバイダの場合はインフォエリアの技術名称 • データソースが BEx クエリの場合はインフォプロバイダの技術名称
OLAP_CUBE	値はデータソースの種類に依存します。 • データソースがサーバの場合は空の文字列 • データソースがインフォプロバイダの場合はインフォプロバイダの技術名称 • データソースが BEx クエリの場合は Info Query の技術名称

9.1.5 OLAP 接続

次の表は、XMLA ドライバを介した MSAS 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は Olap です。

パラメータ名	説明
DBMS	値は以下のいずれかです。 • Microsoft Analysis Services 2008 • Microsoft Analysis Services 2012
NETWORK_LAYER	値は XMLA です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO • PROMPTED
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
OLAP_SERVER_NAME	値は URL です。

次の表は、Essbase ミドルウェアを介した Essbase 接続を記述するパラメータの一覧です。

パラメータ名	説明
CONNECTIVITY_TYPE	値は Olap です。
DBMS	値は Hyperion Essbase 11.x です。
NETWORK_LAYER	値は Essbase API for Analysis です。
AUTHENTICATION_MODE	タイプは列挙です。 使用可能な値は、以下のとおりです。 • FIXED • MAPPING • SSO • PROMPTED
USER_NAME	タイプは文字列です。
PASSWORD	タイプは文字列です。
OLAP_SERVER_NAME	タイプは文字列です。 値は URL です。
OLAP_PATH	タイプは文字列です。
OLAP_CUBE	タイプは文字列です。
LANGUAGE	ロケール (例: en_US)。

9.2 事前定義済みクエリスクリプトプロパティのリファレンス

以下の表に、データファンデーションまたはビジネスレイヤに適用することができる事前定義済みクエリスクリプトプロパティを示します。デフォルト値はインフォメーションデザインツールに表示される値です。表では、どのプロパティがデータファンデーションまたはビジネスレイヤで必須であるかも示しています。包括的な定義については、インフォメーションデザインツールユーザガイドを参照してください。

名前	型	デフォルト値	データファンデーション	データファンデーションで必須	ビジネスレイヤ	ビジネスレイヤで必須
ANSI92	論理値	No	✔	○	✔	×
AUTO_UPDATE_QUERY	論理値	No	✔	○	✔	○
BEGIN_SQL	文字列	値なし	✔	○	✔	×
BLOB_COMPARISON	論理値	No	✔	○	✔	×
BOUNDARY_WEIGHT_TABLE	整数	-1	✔	○	✔	×
COMPARE_CONTEXTS_WITH_JOINS	論理値	Yes	✔	○	✔	×
CUMULATIVE_OBJECT_WHERE	論理値	No			✔	○
DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION	論理値	No			✔	○
DISTINCT_VALUES	文字列	DISTINCT			✔	○
END_SQL	文字列	値なし	✔	○	✔	×
EVAL_WITHOUT_PARENTHESIS	論理値	No			✔	○
FILTER_IN_FROM	論理値	No			✔	×
FORCE_SORTED_LOV	論理値	No	✔	○	✔	○
GROUPBY_PRIMARY_KEY	論理値	Yes			✔	×
INNERJOIN_IN_WHERE	論理値	No	✔	×	✔	×
JOIN_BY_SQL	論理値	No	✔	○	✔	×
MAX_INLIST_VALUES	整数	-1	✔	○	✔	×

名前	型	デフォルト値	データファンデーション	データファンデーションで必須	ビジネスレイヤ	ビジネスレイヤで必須
NO_CUSTOM_SQL_CHECK	論理値	No			✔	×
REPLACE_COMMA_BY_CONCAT	論理値	No	✔	○	✔	×
SELFJOINS_IN_WHERE	論理値	No	✔	×	✔	×
SHORTCUT_BEHAVIOR	文字列	ShortestPath	✔	○	✔	×
SMART_AGGREGATE	論理値	No			✔	×
THOROUGH_AGGREGATE_AWARE	論理値	Yes			✔	×
THOROUGH_PARSE	論理値	No	✔	○	✔	×
TRUST_CARDINALITIES	論理値	No	✔	○	✔	○
UNICODE_STRINGS	論理値	No	✔	○	✔	×
USE_ENHANCED_QUERY_STRIPPING	論理値	No	✔	○	✔	○

9.3 識別可能なオブジェクト参照

この節では、ID を使用して識別できるインタフェースとそれらのサブインタフェースが一覧表示されます。ここで示されるすべてのオブジェクトはインタフェース Identifiable から継承されます。

インタフェース	サブインタフェース
Column	• CalculatedColumn • DatabaseColumn • InputColumn
Context	該当なし
Join	SQLJoin
Lov	• BusinessHierarchicalLov • BusinessQueryLov • SQLQueryLov • StaticLov

インタフェース	サブインタフェース
Parameter	該当なし
Table	• AliasTable • DatabaseTable • DerivedTable • FederatedTable
BlItem	• Attribute • BlContainer • BusinessFilter • BusinessObject • Dimension • Filter • Folder • Measure • NativeRelationalFilter • RootFolder
BusinessLayerView	該当なし
NavigationPath	該当なし

9.4 ValueFormat 文字列パターン参照

ValueFormat オブジェクトを使用してカスタムデータ書式を定義します。BI Semantic Layer Java SDK には、ValueFormat の文字列部分の作成に役立つ一連の数値および日時パターンが用意されています。数値および日時書式の定義については、インフォメーションデザインツールユーザガイドを参照してください。

数値パターン [100 ページ]
以下は、カスタムデータ書式で数字を表すために使用される数値パターンの一覧です。

日時パターン [101 ページ]
以下は、カスタムデータ書式で日時を表すために使用される日時パターンの一覧です。

9.4.1 数値パターン

以下は、カスタムデータ書式で数字を表すために使用される数値パターンの一覧です。

カテゴリ	パターン	説明
符号	-	値が負の場合は、負の符号です。値が正またはゼロの場合は、何も付きません。

カテゴリ	パターン	説明
	<positive sign>	値が負の場合は、負の符号です。値が正またはゼロの場合は、正の符号です。
	(	値が負の場合は、開きかっこです。値が正またはゼロの場合は、何も付きません。
	)	値が負の場合は、閉じかっこです。値が正またはゼロの場合は、何も付きません。
桁	#	オプションの桁です。有効な場合にのみ、その数字を表示します。
	0	必須の桁です。有効な場合は、その数字を表示します。それ以外の場合は、0 を表示します。
区切り	.	数字の正数部と小数部を区切るために使用する記号です。使用される記号は、ロケールによって決まります。小数区切り文字は、表現式内で 1 度だけ使用することができます。
	,	デフォルトでは、数字はロケールによって定義されるルールおよび区切り記号によってグループ化されます。グループ化記号は、表現式内で 1 度だけ使用することができます。小数点よりも左で使用する必要があります。
指数	E+	指数符号は、常に符号付きの大文字です。1 つの式で使用できるのは、1 回のみです。
	E-	指数符号は大文字です。値が負の場合のみ、符号が付きます。1 つの式で使用できるのは、1 回のみです。
	e+	指数符号は、常に符号付きの小文字です。1 つの式で使用できるのは、1 回のみです。
	e-	指数符号は小文字です。値が負の場合のみ、符号が付きます。1 つの式で使用できるのは、1 回のみです。
パーセント	[%]	100 を乗じた値です。
	<times 100 and sign>	100 を乗じた値の後ろにパーセント記号 (%) が付いた値です。1 つの式で使用できるのは、1 回のみです。
ブール	<bool>	数値がゼロではない場合は Ture のローカライズされた値、数値がゼロの場合は False のローカライズされた値です。
	<true>	常にローカライズされた True の値を表示します。
	<false>	常にローカライズされた False の値を表示します。

9.4.2 日時パターン

以下は、カスタムデータ書式で日時を表すために使用される日時パターンの一覧です。

カテゴリ	パターン	説明
日	dd	01 から 31 の 2 桁で表される日付。

カテゴリ	パターン	説明
	d	1 から 31 の 1 桁または 2 桁で表される日付。
	dddd	ロケールに応じた曜日名。たとえば、Monday です。
	ddd	ロケールに応じた、先頭が大文字の曜日の略称。たとえば、Mon です。
	eee	001 から 366 の 3 桁で表される年間の日。
	ee	01 から 366 の 2 桁または 3 桁で表される年間の日。
	e	1 から 366 の 1 桁、2 桁または 3 桁で表される年間の日。
	F	ロケールに応じた月の曜日の番号。たとえば、3rd Monday of June の場合は 3 です。
	DDDD	大文字の曜日。たとえば、MONDAY です。
	<lower day name>	小文字の曜日。たとえば、monday です。
	Dddd	先頭が大文字の曜日。たとえば、Monday です。
	DDD	曜日の大文字の略称。たとえば、MON です。
	<lower abb day name>	曜日の小文字の略称。たとえば、mon です。
	Ddd	先頭が大文字の曜日の略称。たとえば、Mon です。
月	MM	01 から 12 の 2 桁で表される月。
	M	1 から 12 の 1 桁または 2 桁で表される月。
	mmmm	ロケールに応じた、先頭が大文字の月の名前。たとえば、June です。
	mmm	ロケールに応じた、先頭が大文字の月の名前の略称です。たとえば、Jun です。
	MMMM	大文字の月の名前。たとえば、JUNE です。
	<lower month name>	小文字の月の名前。たとえば、june です。
	Mmmm	先頭が大文字の月の名前。たとえば、June です。
	MMM	大文字の月の略称。たとえば、JUN です。
	<lower abb month name>	小文字の月の略称。たとえば、jun です。
	Mmm	先頭が大文字の月の略称。たとえば、Jun です。
年および年号	yy	00 から 99 の 2 桁で表される年。
	yyyy	0000 から 9999 の 4 桁で表される年。
	rr	元号と年
	g	元号 (英語略称) と年を表す数値。たとえば、H20 です。
	jyy	2 桁で表される元号の年。
	jy	1 桁または 2 桁で表される元号の年。

カテゴリ	パターン	説明
	gg	元号
	r	現在は使用しません。jyy パターンと同じ結果を返します。
	G	紀元前、紀元後の省略記号。たとえば、AD または BC です。
週	W	1 から 6 の 1 桁で表される月間の週番号。
	ww	01 から 53 の 2 桁で表される年間の週番号 (ISO 週番号)。
	w	1 から 53 の 1 桁または 2 桁で表される年間の週番号 (ISO 週番号)。
	iyyy	0000 から 9999 の 4 桁で表される ISO 年番号 (ISO 週番号と同じ)。
	iy	00 から 99 の 2 桁で表される ISO 年番号 (ISO 週番号と同じ)。
四半期および半期	q	1 から 4 の 1 桁で表される四半期の番号。
	qq	Q1 から Q4 で表される四半期の略称。
	qqq	第 1 四半期から第 4 四半期で表される四半期の名前。
	p	1 または 2 で表される半期の番号。
時間	H	00 から 23 の 2 桁で表される 24 時間形式の時間。
	HH	0 から 23 の 1 桁または 2 桁で表される 24 時間形式の時間。
	h	01 から 12 の 2 桁で表される 12 時間形式の時間。
	hh	1 から 12 の 1 桁または 2 桁で表される 12 時間形式の時間。
	k	01 から 24 の 2 桁で表される 24 時間形式の時間。
	kk	1 から 24 の 1 桁または 2 桁で表される 24 時間形式の時間。
	K	00 から 11 の 2 桁で表される 12 時間形式の時間。
	KK	0 から 11 の 1 桁または 2 桁で表される 12 時間形式の時間。
分	mm	00 から 59 の 2 桁で表される分。
	m	0 から 59 の 1 桁または 2 桁で表される分。
秒、ミリ秒など	ss	00 から 59 の 2 桁で表される秒。
	s	0 から 59 の 1 桁または 2 桁で表される秒。
	SSS	000 から 999 の 3 桁で表されるミリ秒。
	SS	00 から 99 の 2 桁で表される 100 分の 1 秒。
	S	1 から 9 の 1 桁で表される 10 分の 1 秒。
タイムゾーン	z	協定世界時からの補整値。たとえば、GMT+00:00 です。
AM/PM	a	ロケールに応じた、先頭が大文字の午前/午後の略語。たとえば、AM または PM です。推奨。
	A	大文字の午前/午後の略語。たとえば、AM または PM です。

カテゴリ	パターン	説明
	`<lower am/pm sign>`	小文字の午前/午後の略語。たとえば、am または pm です。
	`<cap am/pm sign>`	先頭が大文字の午前/午後の略語。たとえば、Am または Pm です。非推奨。
区切り文字	`<date separator>`	現在は使用しません。このパターンは、Desktop Intelligence で日付区切り文字として使用されていましたが、推奨されません。日付区切り文字として使用する文字を形式記述に直接入力するか、デフォルトの書式を使用します。
	`<time separator>`	現在は使用しません。このパターンは、Desktop Intelligence で時刻区切り文字として使用されていましたが、推奨されません。時刻区切り文字として使用する文字を形式記述に直接入力するか、デフォルトの書式を使用します。

重要免責事項および法的情報

ハイパーリンク

リンクの一部は、アイコンやマウスオーバーテキストで分類されています。これらのリンクから、追加の情報を得ることができます。
アイコンについて。

- このアイコンが付いたリンク: SAP がホストしているものではない Web サイトに移動します。これらのリンクを使用することで、お客様は (お客様と SAP との契約書に別段の明示的な記載がない限り) 以下のことに同意することになります。
  - リンク先のサイトのコンテンツが SAP のドキュメンテーションではないこと。お客様は、この情報に基づいて SAP に対する製品クレームを推断することはできません。
  - SAP が、リンク先のサイトのコンテンツについて同意することも反対することもなく、また SAP がその利用可能性や正確性について保証しないこと。SAP は、かかるコンテンツの使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。
- このアイコンが付いたリンク: 当該の特定の SAP 製品又はサービスのドキュメンテーションから離れ、SAP がホストしている Web サイトに移動します。これらのリンクを使用することで、お客様は (お客様と SAP との契約書に別段の明示的な記載がない限り)、この情報に基づいて SAP に対する製品クレームを推断することはできないことに同意します。

外部プラットフォームでホストされているビデオ

一部のビデオは、サードパーティのビデオホスティングプラットフォームに置かれている場合があります。SAP では、これらのプラットフォームに保存されているビデオが将来にわたって利用できると保証することはできません。また、これらのプラットフォームにホストされている、いかなる広告またはその他のコンテンツ (関連ビデオまたは同じサイトでホストされている別のビデオに移動する場合など) については、SAP の管理外であり責任を負いません。

ベータおよびその他の試験的機能

試験的機能は、SAP が将来のリリースを保証する正式に提供される機能の範囲外です。これは、試験的機能は、SAP により通知なく理由の如何を問わず随時変更される場合があることを意味します。試験的機能は、本稼働使用のためのものではありません。お客様は、試験的機能を実際の運用環境で、又は十分なバックアップがとられていないデータとともに、デモンストレーション、テスト、試験、評価その他の方法で使用してはなりません。
試験的機能の目的は、早期にフィードバックを得ることで、それに応じて顧客の皆様やパートナーが将来の製品に影響を与えることを可能にすることです。SAP コミュニティなどにおいてフィードバックを提供することで、お客様は、投稿物や二次的著作物の知的財産権が SAP の独占的所有物であり続けることを承認することになります。

コード例

ソフトウェアのコーディングやコードスニペットはすべて、例です。それらは、本稼働使用のためのものではありません。コード例は、構文や表現規則を分かりやすく説明し視覚化することのみを目的としています。SAP は、コード例の正確性や完全性について保証しません。SAP は、コード例の使用により発生した過誤や損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、損害に対して一切責任を負いません。

偏見のない表現

SAP は、ダイバーシティ & インクルージョンの文化を支持しています。SAP の文書では、可能な限り、文化、民族性、ジェンダー、および障がいの有無を問わず、すべての人々に対する偏見を伴わない表現を採用します。

www.sap.com/contactsap

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。

SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE （又は SAP の関連会社）の商標若しくは登録商標です。本書に記載されたその他すべての製品およびサービス名は、それぞれの企業の商標です。

商標に関する詳細の情報や通知については、https://www.sap.com/japan/about/legal/trademark.html をご覧ください。

THE BEST RUN SAP